

# Hewitt

---

## **Cyborg Scripting Language Advanced Customization – Participant’s Guide**

**Document Issue: 5.1**

Document Issue Status:	Sixth Release
Document Issue Level:	5.1
Document Issue Date:	April 2004
Software Version:	5.1

### **Copyright Notice**

Copyright© 2004 by Hewitt Associates LLC. All rights reserved.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series® are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear, but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

# Contents

<b>Section 1: Course Overview</b>	<b>1</b>
Course introduction .....	3
Course logistics .....	5
Course materials .....	7
<b>Section 2: Direct File Processing</b>	<b>9</b>
Introduction .....	11
The Solution Series files.....	13
Reading random files.....	17
Reading files sequentially .....	31
Section 2 exercise 1 .....	35
Unlocking files .....	37
Writing and rewriting records .....	39
Deleting records .....	45
Section summary .....	49
Section 2 exercise 2 .....	53
<b>Section 3: Option List Programming</b>	<b>55</b>
Introduction .....	57
Option lists overview.....	59
Creating option list logic .....	61
Compiling option list logic.....	65
Calling option list logic .....	67
Section summary .....	69
Section 3 exercise .....	71
<b>Section 4: User-Defined Table Records</b>	<b>73</b>
Introduction .....	75
User-defined table overview.....	77
Table analysis .....	79
Table record layout.....	81
Table field definitions .....	87
Section 4 exercise 1 .....	97
User-defined table entry form .....	99
Verify table definitions.....	105
Table record read method.....	109
Section summary .....	113
Section 4 exercise 2 .....	117

<b>Section 5: Employee Database Updating</b>	<b>119</b>
Overview .....	121
Inserting a segment.....	123
Deleting a segment .....	131
Batch transaction updating .....	135
Section summary .....	139
Section 5 exercise.....	141
<b>Section 6: Special Report Options</b>	<b>143</b>
Introduction .....	145
Special print option overview.....	147
Percentage of total report .....	149
Multiple employee format—3 across labels.....	153
Report schedule parameters.....	157
System Control Repository reporting.....	163
Section 6 exercise.....	165
<b>Appendix A: Exercise Answers</b>	<b>167</b>
Section 2 exercise 1 .....	168
Section 2 exercise 2 .....	169
Section 3 exercise.....	171
Section 4 exercise 1 .....	174
Section 4 exercise 2.....	178
Section 5 exercise.....	187
Section 6 exercise.....	189
<b>Appendix B: Extra for Experts</b>	<b>193</b>
Activating user-defined files .....	195
FILE23 delivered code for UNIX.....	198
FILE23 UNIX Overrides.....	199
FILE24 delivered code for UNIX.....	200
FILE24 UNIX Overrides.....	200
FILE25 delivered code for UNIX.....	202
FILE25 UNIX Overrides.....	202
PULL Control Record with Overrides: .....	204
System Control Repository Key Structure(s) .....	206
Employee Database Key Structure(s).....	216
<b>Appendix C: The TRACE Utility</b>	<b>221</b>
The TRACE utility .....	222
Tracing an entire program online .....	224
Tracing a portion of the program—batch reports.....	225
Tracing paragraphs within a program—TRACE verbs.....	226
Additional trace tips .....	229
Reading the TRACE output .....	231
TRACE examples.....	232
<b>Glossary</b>	<b>239</b>
<b>Index</b>	<b>257</b>

# Section 1: Course Overview

---

## Table of Contents

Course introduction.....3  
Course logistics .....5  
Course materials.....7

# Course Introduction

- **Purpose**
- **Benefits**
- **Audience**
- **Prerequisites**
- **Goals**
- **Expectations**

---

**NOTES**

---

## Course introduction

### Purpose

The purpose of this course is to teach you Cyborg Scripting Language (CSL) skills for special file manipulations and programming techniques.

### Benefits

The benefit of learning this information is to increase your CSL programming skills. In the programming role, you will be able to perform more complex programming tasks. System efficiency is also addressed.

### Audience

This course has been designed for technical project members and data processing personnel who are experienced CSL users.

### Prerequisites

Before taking this course you should have completed the following courses:

- Using the Solution Series: Administrative Solutions
- Introduction to Cyborg Scripting Language
- Cyborg Scripting Language Customization
- Cyborg Scripting Language Report Customization

### Goals

At the conclusion of this course you should be able to:

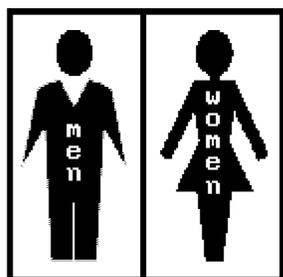
- Identify the rules and programming techniques of reading, writing and deleting records from The Solution Series files.
- Recognize and use techniques for special functions of query, form, and report programs.

### Expectations

To ensure that you accomplish the above goals please do the following:

- Ask questions.
- Share examples of your own Solution Series-related experiences.
- Ask where to obtain additional information if you have an interest in a point that is introduced.

# Logistics



---

**NOTES**

---

# Course logistics

Use the space below and in the right column to take notes about the course logistics.

## Meals

---

---

---

---

## Breaks

---

---

---

---

## Telephones

---

---

---

---

## Restrooms

---

---

---

---

## Security

---

---

---

---

## Questions

---

---

---

---

# Course Materials



---

**NOTES**

---

## Course materials

### Participant guide

The participant guide contains the following sections:

### Table of contents

- Each section contains a table of contents listing the topics within the section.

### Text layout

- Upper pages typically contain copies of overhead transparencies or forms.
- Lower left pages contain information about the overhead transparency or form.
- Lower right pages are blank for your note taking.

### Section exercise

- You will have an opportunity to practice what you have learned in each section by completing section exercises. All sections except Section 1 have section exercises.

### Appendices

- Exercise Answers  
Answers to section exercises.
- Extra for Experts  
Additional CSL topics, System Control Repository and Employee Database key layouts.
- The TRACE Utility  
Additional CSL topics, which explain and provide examples for tracing/debugging programs.

### Glossary

Glossary and syntax for the CSL verbs in this course.

### Index

An alphabetical listing of topics cross-referenced to page numbers.

---

**NOTES**

---

## Section 2: Direct File Processing

---

### Table of Contents

Introduction.....	11
The Solution Series files .....	13
Reading random files .....	17
Reading files sequentially .....	31
Section 2 exercise 1.....	35
Unlocking files.....	37
Writing and rewriting records .....	39
Deleting records .....	45
Section summary .....	49
Section 2 exercise 2.....	53

## Objectives

- **Recall the available Solution Series files**
- **Identify random file key structure**
- **Identify direct file processing verbs**
- **Create a program using direct file processing**

---

### NOTES

---

## Introduction

### **Purpose**

In this section we relate Solution Series file types and key structures to input/output functions of programming.

### **Objectives**

Upon completion of this section, you should be able to:

- Recall the available Solution Series files
- Identify random file key structure
- Identify direct file processing verbs
- Create a program using direct file processing

## The Solution Series Files

File Name	Organization	Assignment	Input/Output	Record Size	Purpose
FILE01	Random	Disk	Input/Output	80	System Control Repository
FILE02	Random	Disk	Input/Output	Variable (3060 max)	Employee Database
FILE03	Sequential	Printer	Output	132	Audit/Report/Message Print File
FILE04	Sequential	Reader	Input	80	Control Records File
FILE05	Sequential	Disk	Input	80	Data Input File
FILE06	Random	Disk	Output	80	Installation System Control Repository
FILE07	Random	Disk	Output	Variable (3060 max)	Installation Employee Database
FILE10	Sequential	Disk/Tape	Output	80	Data Output File
FILE11	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE12	Sequential	Disk/Tape	Output	256	Payroll Process Batch Master File
FILE13	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE14	Sequential	Disk	Input	150	Report Extract Input File
FILE15	Sequential	Disk	Output	150	Report Extract Output File
FILE17	Sequential	Printer	Output	132	Alternate Print File
FILE18	Sequential	Printer	Output	132	Alternate Print File
FILE19	Sequential	Printer	Output	132	Alternate Print File
FILE23	Random	Disk	Input/Output	Variable (3060 max)	User Defined File
FILE24	Sequential	Disk/Tape	Input	Variable (3060 max)	User Defined File
FILE25	Sequential	Disk/Tape	Output	Variable (3060 max)	User Defined File
FILE30	Sequential	Sequential	Output	320	Savings Bonds
FILE31	Sequential	Disk/Tape	Output	132	Check Print

---

### NOTES

---

## The Solution Series files

### Input and output functions

Input and output operations are performed on Solution Series files by identifying the file's record key.

Typical file maintenance procedures include reading, writing, rewriting and deleting records. Certain of these functions are limited to input files and others to output files. The categories of Input/Output, Input Only and Output Only include:

### Input/output files

Recall that these are The Solution Series input/output files:

- FILECL
- FILE01
- FILE02
- FILE23

### Input only files

Recall that these are The Solution Series input-only files:

- FILE04
- FILE05
- FILE11
- FILE13
- FILE14
- FILE24

### Output only files

Recall that these are Solution Series output-only files:

- FILE03
- FILE06
- FILE07
- FILE10
- FILE12
- FILE15
- FILE25
- FILE30
- FILE31

# The Solution Series Files

## System Control Repository Records

Record Type	Description
A	Machine Parameter Records
B	Working Storage Expansion Records
C	Option Lists
D	HRMS Default Information
F	Field Name Table (Data Dictionary)
P	Program Records (sub-divided by type)
PC	Org. Number Report Validation Records
PD	Org. Number Report Scheduling Records
PE	Report Scheduling Records
Q	Alternate Keys (used with QUERY)
R	Report Format Record
RQM	Query Maintenance Facility Records
RRM	Report Maintenance Facility Records
RT	Report Print Position Records
T	Tables
Y	Security Data
ZL	Lock Record(s) (temporary record)

## Employee Database Records

Record Type	Description
Binary	Report Generators
xxxxxxD	Company Records
xxxxxxF	Company Other Records
xxxxxxG	Company Other Records
xxxxxxH	Tax Body Records
xxxxxxM	Employee Records
xxxxxxW	Employee Other Records
xxxxxxX	Employee Other Records
ZI	Is/Was Audit Records
ZQ	Online Pay Calculation Records
ZR	Report Viewing Records
ZX	Executable Code
ZY	Session Records
ZZ	Audit (Log) Records
ZZA	Time Cards and Adjustment Records
ZN	Audit Log Recs

System Control Repository (FILE01)	
24 Character Key	56 Characters of Data

Employee Database (FILE02)		
Record Length	32 Character Key,	Maximum of 3028 Characters of Data

User Defined (FILE23)	
24 Character Key	Maximum of 3036 Character of Data

---

## NOTES

---

---

## The Solution Series files, continued

### System Control Repository

The System Control Repository (FILE01) contains programs, delivered option lists, fields, and any other information needed to make the system run. Any time you sign on the system, you are using the System Control Repository. The System Control Repository (FILE01) contains fixed, 80 byte records with a 24-character key.

☞ *Refer to Appendix B: Extra for Experts for a complete list of key structures.*

### Employee Database

The Employee Database (FILE02) contains your data. Both company level and employee level information are defined and reside on the Employee Database. The Employee Database contains variable length records with a maximum length of 3060. The Record Key is 32-characters long. The record key begins in the byte following Cyborg's record length descriptor.

☞ *Refer to Appendix B: Extra for Experts for a complete list of key structures.*

### User-defined

The random user-defined file (FILE23) can contain records with a maximum length of 3060 and a Record Key of up to 24 characters. Specific definition of the record size and key length are your responsibility.

☞ *Appendix B: Extra for Experts provides detail on creating COBOL overrides to define the record and key lengths for user-defined files.*

## Reading Random Files

- **Build the ‘search argument’ or key to the Record**
- **Read the random file using the key**
- **Check the status of the file read**

---

**NOTES**

---

## Reading random files

### Record retrieval random files

To read a record from a random file, a search argument must be used. In other words the desired record to be retrieved must be described by its key attributes for record look up. These steps are taken:

- The 'search argument' or key attributes are formatted in Pointer 7.
- An attempt is made to read the matching record using a CSL verb and the record read is moved to Pointer 8.
- The status of the operation is verified by checking the contents of a system maintained field.

All direct file operations including writing, rewriting and deleting require file status checking.

# Reading Random Files using READ-UNIQUE

	<b>FILE01 Example</b>
<b>Key Length</b>	MOVE '02' TO W7-02-008.
<b>Key Contents</b>	MOVE 'TA' TO W7-02-010.
<b>Read Operation</b>	READ-UNIQUE FILE01.
<b>Check Status</b>	IF STAT-KEY GREATER THAN '00' PRINT 'No Table Record Found' ELSE PRINT W8-79-000.
	<b>FILE02 Example:</b>
<b>Key Length</b>	MOVE '06' TO W7-02-044.
<b>Key Contents</b>	MOVE 'ZI' TO W7-02-046. MOVE SESSION-ID TO W7-04-048.
<b>Record Read</b>	READ-UNIQUE FILE02.
<b>Status Check</b>	IF STAT-KEY NOT EQUAL '00' PRINT 'No Audit for this session' ELSE PRINT 'Operator Id: ' W8-04-026.
	<b>FILE23 Example</b>
<b>Key Length</b>	MOVE '04' TO W7-02-008.
<b>Key Contents</b>	MOVE 'XXXX' TO W7-04-010.
<b>Read Operation</b>	READ-UNIQUE FILE23.
<b>Check Status</b>	IF STAT-KEY GREATER THAN '00' PRINT 'No User File Record Found' ELSE PRINT W8-50-000.

---

## NOTES

---

## Reading random files, continued

### Reading FILE01

To read a record from FILE01, build a search argument of up to 24 characters, then specify the keys length in Pointer 7. The specific fields for FILE01 are KEY01–SIZE a.k.a. W7–02–008, and KEY01–AREA a.k.a. W7–24–010.

### Reading FILE02

To read a record from FILE02, build a search argument of up to 32 characters, then specify the keys length in Pointer 7. The specific fields for FILE02 are KEY02–SIZE a.k.a. W7–02–044, and KEY02–AREA a.k.a. W7–32–046.

### Reading FILE23

To read a record from FILE23, build a search argument of up to 24 characters, then specify the keys length in Pointer 7. The specific fields for FILE23 are KEY01–SIZE a.k.a. W7–02–008, and KEY01–AREA a.k.a. W7–24–010.

### READ–UNIQUE

When the file is read using READ–UNIQUE a record is read into the first positions of Pointer 8, at displacement 000. The data can be processed from this location or may be moved to other pointers where field names are defined.

*Note: RDBMS users using direct reads to Company, Tax, or Employee records on FILE02 will not build the logical record in memory. These FILE02 records only contain a reference of which relational tables must be read, not the actual data.*

## Reading Random Files using READ-UNIQUE, continued

STAT-KEY	Description
00	Good I/O
01	Record read has a key greater than the search argument key
10	End of file
22	Write failed due to duplicate record
23	Record not found (Delete or Rewrite)
24	Space exhausted
90	Invalid request
91 or 93	Invalid file number
95	Open failed
99	Invalid key

---

### NOTES

---

## Reading random files, continued

### STAT-KEY

Each time a record is read, the field STAT-KEY contains the result of the read. The chart on the previous page shows the general list of STAT-KEY return codes.

# Alternate Method of Reading FILE02

## Company Record:

**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Read:** READ-COMPANY.  
**Display Data** PRINT COMPANY-NAME COMPANY-ADDRESS.

## Tax Record:

**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Key Field:** MOVE 'TX 2IL ' TO KEY-FIELD.  
**Read:** READ-TAXES.  
**Display Data** PRINT TAX-BODY TAX-FILING-NUMBER.

## Employee Record:

**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Key Field:** MOVE '1234567890' TO KEY-FIELD.  
**Read:** READ-EMPLOYEE.  
**Display Data** PRINT SOCIAL-SECURITY-NBR BIRTH-DATE.

---

## NOTES

---

---

## Reading random files, continued

### **FILE02 alternate**

As an alternate to reading FILE02 using READ-UNIQUE, you may wish to use the various READ- verbs. The benefits include:

- Using the Command Line fields to build the key.
- Accessing data by field name instead of work name.
- More than one 3060-byte record may be read.

*Note: This is a requirement for RDBMS databases since these FILE02 records only contain a reference of which relational tables must be read to build the logical record in memory.*

### **READ-COMPANY**

Reads the company record data and moves it into Pointers 21-24. This verb requires the COMPANY-NUMBER (W7-06-240) as the key.

### **READ-TAXES**

Reads the tax record and moves it into pointers 25-27. This verb requires the COMPANY-NUMBER (W7-06-240) and KEY-FIELD (W7-10-254) fields as the key.

### **READ-EMPLOYEE**

Reads the employee record data and moves it into pointers 28-37. This verb requires the COMPANY-NUMBER (W7-06-240) and KEY-FIELD (W7-10-254) fields as the key.

*Note: Programs running in batch that issue a READ-verb will end abnormally if the record is not found. A READ-UNIQUE and STAT-KEY check prior to the READ-verb will avoid a program stop.*

# Alternate Method of Reading FILE02

**Employee History Record:**  
**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Key Field:** MOVE '1234567890' TO KEY-FIELD.  
**Addl Key Field:** MOVE SPACES TO W7-04-264.  
**Read:** READ-HISTORY.  
**Display Data** PRINT SOCIAL-SECURITY-NBR BIRTH-DATE.

**Employee Labor Record:**  
**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Key Field:** MOVE '1234567890' TO KEY-FIELD.  
**Addl Key Field:** MOVE SPACES TO W7-04-264.  
**Read:** READ-LABOR.  
**Display Data** PRINT SOCIAL-SECURITY-NBR BIRTH-DATE.

**Current Employee History or Labor Record:**  
**Org. Number Field:** MOVE '999999' TO COMPANY-NUMBER.  
**Key Field:** MOVE '1234567890' TO KEY-FIELD.  
**Addl Key Field:** MOVE SPACES TO W7-04-264.  
**Read:** READ-HL.  
**Display Data** PRINT SOCIAL-SECURITY-NBR BIRTH-DATE.

---

## NOTES

---

## Reading random files, continued

### **FILE02 alternate**

#### **READ-HISTORY**

Reads the employee history record data into pointers 28-37. This verb requires the COMPANY-NUMBER (W7-06-240), KEY-FIELD (W7-10-254) and ADDITIONAL-KEYs first four positions (W7-04-264) fields as the key. If the ADDITIONAL-KEY field is left blank, the READ-HISTORY verb uses the employees EA segment MASTER-NUMBER to read the most current history record.

Subsequent reads use the ADDITIONAL-KEY field to find the next most current History Record. The READ-HISTORY verb places the next previous MASTER-NUMBER in the ADDITIONAL-KEY field for this purpose.

#### **READ-LABOR**

Reads the employee labor record data into pointers 28-37. This verb requires the COMPANY-NUMBER (W7-06-240) and KEY-FIELD (W7-10-254) fields as the key. The same functionality rules apply to this verb; however, this verb is trying to read labor records instead of history records.

#### **READ-HL**

Reads either the employee labor or history record stored immediately before the permanent master record into pointers 28-37. This verb requires the COMPANY-NUMBER (W7-06-240) and KEY-FIELD (W7-10-254) fields as the key. The same functionality rules apply to this verb; however, subsequent reads return the next previous record.

#### **Screen error**

Each time these instructions are used, check the field SCREEN-ERROR field for a value of 'Y'. If this condition is true, there are no more history or labor records for this employee.

# Custom READ- Verbs

CREATE A READ VERB

VERB NAME: XREAD-FILE23-RECORD FILE NUMBER: 23 MODULE CODE: HR

1ST FIELD: COMPANY-NUMBER

2ND FIELD: EMPLOYEE-NUMBER

3RD FIELD: '000'

4TH FIELD:

5TH FIELD:

6TH FIELD:

7TH FIELD:

8TH FIELD:

---

## NOTES

---

## Reading random files, continued

### Custom READ– verbs

The Create Read Verb (RDVERB) program enables you to create a customized READ– verb. This READ– verb locates records in FILE01, FILE02, or FILE23, based on literal values or fields.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Fields and Verbs
<b>Task:</b>		Create a Read Verb

**Result:** The Create A Read Verb form (RDVERB) displays.

### Form fields

- **Verb Name**  
The defined name of the read verb. The name may be from 1–20 characters.
- **File Number**  
The File Number field is used to identify the Random file to read. Valid file numbers are 01, 02, and 23.
- **Module Code**  
The Module Code field defines the specific module the verb/record is associated with.
- **1st–8th Field**  
The 1st–8th field are used to specify up to eight literal values or valid field names that specify the READ– verb’s key to be used in the read process.

# Custom READ- Verbs

## Read FILE01

### Example:

```
XREAD-USER-TABLE.  
IF STAT-KEY IS NOT EQUAL TO '00'  
    PRINT-REJECT 'SC900'  
    RETURN.  
PRINT XMY-FIELD1 ' ' XMY-FIELD2 ...
```

## Read FILE02

### Example:

```
XREAD-FILE02-RECORD.  
IF STAT-KEY IS NOT EQUAL TO '00'  
    PRINT-REJECT 'SC900'  
    RETURN.  
PRINT W8-10-020 ' ' W8-30-030 ...
```

## Read FILE23

### Example:

```
XREAD-FILE23-RECORD.  
IF STAT-KEY IS NOT EQUAL TO '00'  
    PRINT-REJECT 'SC900'  
    RETURN.  
PRINT W8-05-007 ' ' W8-10-012 ...
```

---

## NOTES

---

## Reading random files, continued

### Processing

The customized READ- verb performs the following when executed:

#### FILE01 READ- verb

- Builds the FILE01 key from the 1ST-8TH FIELD literal and field names.
- Reads FILE01.
- If the record is found the data is placed into pointer 40 starting at displacement 000, otherwise, pointer 40 is initialized to spaces in the first 80 displacements.

*Note:* The pointer number for the 'X' fields must be 40 as that is where the verb moves the data.

#### FILE02 READ- verb

- Builds the FILE02 key from the 1ST-8TH FIELD literal and field names.
- Reads FILE02.
- If the record is found the data is placed into pointer 8 starting at displacement 000, otherwise, the next highest record is placed into pointer 8 starting at displacement 000.

#### FILE23 READ- verb

- Builds the FILE23 key from the 1ST-8TH FIELD literal and field names.
- Reads FILE23.
- If the record is found the data is placed into pointer 8 starting at displacement 000, otherwise, the next highest record is placed into pointer 8 starting at displacement 000.

*Note:* It is important to check the STAT-KEY field which contains the status of the READ- verb.

# Reading Random Files Sequentially

## Random File Example

<b>Key Length:</b>	MOVE '06' TO KEY01-SIZE.
<b>Key Contents:</b>	MOVE 'C HR00' TO W7-06-010.
<b>Read Operation:</b>	READ-UNIQUE FILE01.
<b>Check Status:</b>	IF STAT-KEY GREATER THAN '00' RETURN.
<b>Read Loop:</b>	P100-LOOP. PRINT 'Value: ' W8-14-007. SPACE-OVER :05. PRINT 'Description: ' W8-20-023. NEXT-LINE.
<b>Read Next:</b>	READ FILE01.
<b>Check Status:</b>	IF STAT-KEY GREATER THAN '00' OR W8-06-000 NOT EQUAL TO W7-06-010 RETURN ELSE GO TO P100-LOOP.

---

## NOTES

---

## Reading files sequentially

### Random files

A Random File may be read sequentially only after it has been read randomly. Additionally, Random Files are automatically opened and closed by the CBSV COBOL programs, therefore you will not need to OPEN or CLOSE Random Files.

Once a random file record has been accessed with the READ-UNIQUE verb, and the status key is checked for the first record, the next record on file can be read using READ:

- Execute the READ verb
- Check the status of the read operation
- If the read is good, then process the data in Pointer 8 or move the data to the appropriate pointers

It is important to note that these conditions must be avoided to insure a successful sequential read:

- No random I/O operations can be performed between the execution of the READ-UNIQUE and READ verbs
- The KEY01-SIZE and KEY01-AREA should not be changed

### READ

The READ verb is used to sequentially process a file.

# Reading Sequential Files

## Sequential File Example

<b>Open Operation:</b>	OPEN FILE14. IF STAT-KEY NOT EQUAL '00' PRINT-REJECT 'SC14090' RETURN.
<b>Read Loop:</b>	P100-LOOP.
<b>Read Operation:</b>	READ FILE14.
<b>Check Status:</b>	IF STAT-KEY NOT EQUAL '00' PRINT-MESSAGE 'SC002' CLOSE FILE14 @Optional RETURN. PRINT W8-20-000. NEXT-LINE.
<b>Read Next:</b>	GO TO P100-LOOP.

---

## NOTES

---

## Reading files sequentially, continued

### Sequential files

Sequential Files may be read from start to finish using CSL. To process a sequential file it may be necessary to OPEN and CLOSE the file.

### OPEN

The OPEN verb is used to begin the processing of a file. The CBSV COBOL program opens all standard files automatically.

### CLOSE

The CLOSE verb is used to end the processing of a file. The CBSV COBOL program closes all standard files automatically. DO NOT use the CLOSE verb for a file unless the file has been opened by the same program and is being used by that program only.

### STAT-KEY

The STAT-KEY for sequential processing can be checked for a value of '00' for a good record read, or '10' for an end of file condition.

### Example

Here are the steps in the sequential file read process:

- The OPEN is issued and the status is checked
- A record is read and the status is checked
- The record is processed
- Another read attempt is made, the file is closed when there are no more records to read

# Section 2 Exercise 1

---

**NOTES**

---

## Section 2 exercise 1

### Purpose

The purpose of this exercise is to give you practice retrieving and displaying a record using the direct file method.

### Directions

Take 20 minutes to write and execute an online program that performs the following:

1. Read the System Control Repository (FILE01) Option List Records with a key of 'C SC04'.
2. Display the Description and Value of each code in the Option List.  
Option List Description: Displacement = 23  
Length = 20  
Option List Value: Displacement = 7  
Length = 14

# Unlocking a Random Record

## UNLOCK Example

<b>Key</b>	MOVE 'PM' TO W7-02-010. MOVE OPERATOR-ID TO W7-04-012. MOVE '06' TO KEY01-SIZE.
<b>FILE01 Read</b>	READ-UNIQUE FILE01.
<b>Read Status</b>	IF STAT-KEY NOT EQUAL '00'
<b>Unlock</b>	PRINT-MESSAGE 'SC130' UNLOCK FILE01 RETURN.
<b>Process Record</b>	PRINT W8-10-000.

---

## NOTES

---

## Unlocking files

### File access

When accessing a random file with a READ-UNIQUE instruction, the file is temporarily locked for access by other programs. This condition could produce a 'wait state' that is unfavorable. For system efficiency, we recommend the use of the UNLOCK verb.

### UNLOCK

UNLOCK is a verb used to unlock a file (FILE01, FILE02, FILE23) if the file access method has lock/unlock capabilities. The file is automatically unlocked by other I/O instruction such as CALL, DELETE, WRITE, REWRITE or RETURN (CYB900).

Control is passed back to CYB90 when RETURN is encountered or automatically at the end of the program. However, when the next I/O instruction to follow your program's execution is conditional, use UNLOCK to improve the system's response time.

### Example

This example attempts to read an electronic message from the System Control Repository. When no message exists, the file is unlocked for other I/O processing.

- A FILE01 Key is primed
- The record read is attempted
- The STAT-KEY is checked for an exact match
- If there is no record for this key the file is unlocked and the 'You have no message' message displays, otherwise the record is printed

# Writing Records to a File

## Example 1—Write FILE10

<b>Initialization</b>	READ-EMPLOYEE. INITIAL-80.
<b>Format Output in Pointer 8</b>	MOVE COMPANY-NUMBER TO W8-06-000. MOVE EMPLOYEE-NUMBER TO W8-10-006. MOVE CONTROL 1-2 TO W8-06-016. MOVE UNION-CODE TO W8-05-022. MOVE UNION TO W8-15-027.
<b>Write Record</b>	WRITE FILE10.

## Example 2—Read FILE05 Write FILE01

<b>Open Operation:</b>	OPEN FILE05. IF STAT-KEY NOT EQUAL '00' PRINT-REJECT 'SC05090' RETURN.
<b>Read Loop:</b>	P100-LOOP.
<b>Read Operation:</b>	READ FILE05.
<b>Check Status:</b>	IF STAT-KEY NOT EQUAL '00' PRINT 'SC002' CLOSE FILE05 @Optional RETURN.
<b>Write Record</b>	WRITE FILE01. IF STAT-KEY EQUAL '22' PRINT-REJECT 'SC01W22'.
<b>Read Next:</b>	GO TO P100-LOOP.

---

## NOTES

---

## Writing and rewriting records

### WRITE

The WRITE verb is used to write a record to a file. Since all I/O is from Pointer 8, data must first be moved to this area prior to the write statement.

### Record length

The length of data written to the output file is dependant upon the file specified in the WRITE statement. For example, WRITE FILE10 outputs 80 positions of data starting at displacement 000 of pointer 8, whereas WRITE FILE15 outputs 150 positions of data starting at displacement 000 of pointer 8.

### Random files

When writing data to a Random File, be sure to check for a duplicate record after the WRITE. A STAT-KEY value of '22' indicates that a duplicate record key is present and the write was unsuccessful.

### Example 1

This example reads the employees record and extracts specific data to FILE10.

Note that:

- Pointer 8 is initialized to spaces using the INITIAL-80 verb.
- The fields are moved to specific displacement of Pointer 8.
- This program extracts only one employees data. If you wish to extract more than one employee's data, you should run the program under QUERY as a batch process.

### Example 2

This example reads records from FILE05 and writes them to FILE01. Note that:

- The FILE05 data is assumed to be in FILE01 format.
- The STAT-KEY is checked for each file-processing verb.

# Writing Records to a File

## Example 1—WRITE-FILE10

<b>Initialization</b>	READ-EMPLOYEE. SPACE-EXTRACT-RECORD.
<b>Format Output in Pointer 11</b>	PRINT COMPANY-NUMBER EMPLOYEE-NUMBER BIRTH-DATE UNION-CODE UNION.
<b>Write Record</b>	WRITE-FILE10.

## Example 2—WRITE-EXTRACT

<b>Initialization</b>	READ-EMPLOYEE. SPACE-EXTRACT-RECORD.
<b>Format Output in Pointer 11</b>	PRINT COMPANY-NUMBER EMPLOYEE-NUMBER. FIND-SALARY. IF FOUND OUTPUT ANNUAL-SALARY ELSE OUTPUT '000000000'.
<b>Write Record</b>	WRITE-EXTRACT

---

## NOTES

---

## Writing and rewriting records, continued

### **WRITE–**

The WRITE– verbs are used to build the data to be extracted in the SCREEN area (Pointer 11), then move the data to Pointer 8 where it is finally written to a file.

### **WRITE–FILE10**

The WRITE–FILE10 verb is used to write an 80-character record to FILE10. The record is initially formatted in the SCREEN area using either PRINT or OUTPUT verbs.

### **WRITE–EXTRACT**

The WRITE–EXTRACT verb is used to write a 150-character record to FILE15. The record is initially formatted in the SCREEN area using either PRINT or OUTPUT verbs.

### **Examples 1 and 2**

These examples read the employees record and extract specific data to either FILE10 or FILE15. Note that:

- The SCREEN area (Pointer 11) initialized to spaces starting at displacement 1601 using the SPACE–EXTRACT–RECORD verb.
- The fields are moved to the SCREEN area using PRINT and OUTPUT verbs.
- The data is moved from the SCREEN area to Pointer 8, and written to their respective file using the WRITE– verb.
- If you wish to extract more than one employee’s data, you should run the program under QUERY as a batch process.



## Writing and rewriting records, continued

### Rewrite a record

To rewrite a record to a random file you must first read the record. For example, you may wish to update the Trade and Professional Association Option List with information provided from an outside source. To accomplish this you must consider the following:

- If the record is new (Association Code), the record must be added to the System Repository File, therefore it is written.
- If the record exists, the record must be updated, therefore it is rewritten.

### REWRITE

The REWRITE verb is used to update the record currently in the I/O area. This record must have been previously retrieved and no other I/O can occur between the retrieval and update. Also, the key cannot be modified.

### Example

The example illustrates the Trade and Professional; Association example discussed above. Note that:

- The FILE01 record has to be first read using READ-UNIQUE to determine which technique to use to update FILE01.
- If the STAT-KEY does not indicate an exact match, the record is added using the WRITE verb otherwise the record is updated using the REWRITE verb.
- STAT-KEY is checked for rewrite and an error is produced if the value returned is not '00'.

# Deleting Records from a Random File

**FILE01 Key:** MOVE '08' TO KEY01-SIZE.  
MOVE 'PEWEEKLY' TO W7-08-010. @Report Schedule Record.

**FILE01 Read**  
**Read Status** READ-UNIQUE FILE01.  
IF STAT-KEY NOT EQUAL '00'  
PRINT-MESSAGE 'SC056' @Purge not performed

**File Unlocked** UNLOCK FILE01.  
RETURN.

**Record Delete** DELETE FILE01.  
IF STAT-KEY EQUAL '00'  
PRINT-MESSAGE 'SC058' @Record has been Deleted  
ELSE  
PRINT-MESSAGE 'SC056'. @Purge not performed  
RETURN.

---

## NOTES

---

## Deleting records

### DELETE

The DELETE verb is used to remove the most recently accessed record from a file.

### Example

This example deletes the first WEEKLY report schedule record from the System Control Repository. The record to be deleted must be accessed and moved to the Pointer 8 I/O area. This is done by the READ-UNIQUE.

- After the Key is primed, the record is read.
- The STAT-KEY is checked.
- If the record does not exist it cannot be deleted, and the 'Purge not performed' message displays.
- If the record does exist, the DELETE verb attempts to delete the record.
- If the status of the DELETE is good (00), the 'Record has been deleted' message displays, otherwise the 'Purge not performed' message displays.
- The file is unlocked for efficiency to avoid a 'wait state' for other program's access (record could have been read and not deleted).

# Deleting Records from a Random File

## Example 1—FILE01 Delete:

<b>Key Length</b>	MOVE '03' TO KEY01-SIZE.
<b>Key Area</b>	MOVE 'QTK' TO W7-03-010.
<b>Read FILE01</b>	READ-UNIQUE FILE01.
<b>Status Check</b>	IF STAT-KEY NOT EQUAL '00' PRINT-MESSAGE 'SC056' @No Purge RETURN.
<b>Group Delete</b>	DELETE-GLOBAL FILE01. PRINT-MESSAGE 'SC001' @Delete complete

## Example 2—Employee Database Delete:

<b>Key Length</b>	MOVE '02' TO KEY02-SIZE.
<b>Key Area</b>	MOVE 'ZR' TO W7-02-046.
<b>Read FILE01</b>	READ-UNIQUE FILE02.
<b>Status Check</b>	IF STAT-KEY NOT EQUAL '00' PRINT-MESSAGE 'SC056' @No Purge RETURN.
<b>Group Delete</b>	DELETE-GLOBAL FILE02. PRINT-MESSAGE 'SC001' @Delete complete

---

## NOTES

---

## Deleting records, continued

### Group delete

Deleting a group of records from a random file can be accomplished more efficiently with a special macro verb, DELETE–GLOBAL.

### DELETE–GLOBAL

The DELETE–GLOBAL verb is used to delete a large number of records that have the same partial key and reside together on the random file.

This verb performs a sequential read/delete based on the key built for the READ–UNIQUE verb. Following the READ–UNIQUE syntax:

- The Key Length and Area are primed
- The READ–UNIQUE is executed for placement in the file
- The status is checked to insure correct record retrieval
- The DELETE–GLOBAL is executed

### Example 1

This example will delete all alternate key records with an Alternate Key ID of TK.

### Example 2

This example will delete all reports that have been routed for online review.

## Section Summary

- **The Solution Series files and record keys**
- **Reading records**
- **Unlocking files**

---

NOTES

---

## Section summary

In the space below, complete the Section Summary:

### Files and key structure

The maximum key length for records in FILE01 is: \_\_\_\_\_, FILE02 is: \_\_\_\_\_, and FILE23 is \_\_\_\_\_.

### Reading records

The \_\_\_\_\_ verb is used to read a record from a random file.

Building a search argument is required to read a random file. The fields used to build the search argument are:

FILE01: \_\_\_\_\_ and \_\_\_\_\_

FILE02: \_\_\_\_\_ and \_\_\_\_\_

FILE23: \_\_\_\_\_ and \_\_\_\_\_

The \_\_\_\_\_ field is the system maintained I/O status indicator.

When a direct I/O verb is executed, the record is read from the file to Pointer \_\_\_\_\_.

The READ verb is used to \_\_\_\_\_ a file.

## **Section Summary, continued**

- **Writing and rewriting records**
- **Deleting records**

---

### **NOTES**

---

## Section summary, continued

### Writing and rewriting a record

Writing a record requires moving data to Pointer \_\_\_\_\_ starting at displacement \_\_\_\_\_.

The length of data moved to the output file is determined by \_\_\_\_\_.

\_\_\_\_\_ is a verb used to initialize the 1st 80 positions of Pointer 8 to spaces.

Use the \_\_\_\_\_ verb to update a \_\_\_\_\_ file record.

### Deleting a record

\_\_\_\_\_ is used to remove a single record from a random file. To delete a record, it must first be accessed with the \_\_\_\_\_ or \_\_\_\_\_ verb.

\_\_\_\_\_ is used to delete a group of records from a random file. The records deleted are determined by the contents of the \_\_\_\_\_ and \_\_\_\_\_.

## Section 2 Exercise 2

---

**NOTES**

---

## Section 2 exercise 2

### Purpose

The purpose of this exercise is to give you practice retrieving, writing and deleting a record using the direct file method.

### Directions

Take 20 minutes to write and execute an online program that does the following:

1. Create a program to add the following reports to new a report group named 'CLASS'. Remember that all report groups must have a title record.

☞ *Refer to Appendix B: Extra for Experts for FILE01 key structures.*

1A-RPT

1G-RPT

1R-RPT

For each write operation, check the status key and print a message (one per line) that contains 'STAT KEY: ' and the STAT-KEY value.

2. Verify that your report group 'CLASS' has been established.
3. Create another program to delete the report group named 'CLASS' from FILE01.
4. Verify the deletion of your report group 'CLASS.'

---

**NOTES**

---

# Section 3: Option List Programming

---

## Table of Contents

Introduction..... 57  
Option lists overview ..... 59  
Creating option list logic..... 61  
Compiling option list logic..... 65  
Calling option list logic ..... 67  
Section summary ..... 69  
Section 3 exercise..... 71

## Objectives

- **Recognize the purpose of option list logic**
- **Identify the steps in developing option list logic**
- **Create a calculation option list**

---

### NOTES

---

## Introduction

### **Purpose**

This section focuses on adding CSL logic to option lists.

### **Objectives**

Upon completion of this section you will be able to:

- Recognize the purpose of option list logic
- Identify the steps in developing option list logic
- Create a calculation option list

## Option List Logic

- **Calculation option list**
- **Relational edit option list**

---

### NOTES

---

## Option lists overview

### Function of option lists

As you will recall, option list records are used to validate a field against a list of choices. In addition to this, an option list may have calculation or relational editing logic attached to each code. The terminology to identify the two types of logic that can be associated with an option list is:

- Calculation option list
- Relational edit option list

### Calculation option list

A calculation enables different logic to be performed depending on which code is selected for a field. The calculation resides in the body of the option list and is in the form of CSL.

### Relational edit option list

A relational edit enables different relational edits to be performed, depending on which code is selected for a field. The edit resides in the body of the option list and is in the form of CSL.



## Creating option list logic

### Create option list logic

The option list records are built using the Edit Utility (EDIT) using an object of Option List EL Calc (C/M) or Option List EL Edit (C/R).

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Programming Utilities
<b>Task:</b>		Edit Control Repository Objects

**Result:** The EDIT prompt displays.

1. Select either Codeset EL Calc (C/M) or Codeset EL Edit (C/R).
2. Type the name of the Option List to edit and press enter.

**Result:** The Option List Edit form displays.

### EDIT columns

The Edit form for option list logic column definitions are:

- Line Command  
Specifies whether you are adding, changing or deleting the line.
- Code  
14 positions code value.
- Seq  
3-position sequence contains an M or R followed by a two-digit (nn) sequence number.
- CodeSet Calc–EL Source  
CSL Logic for each code value.

# Creating Option List Logic

Code	Seq	Codeset Calculation - English Language Source
		Appraisal Ratings
	M	RATING-VALUE
	M01	P100-CALC-CODE-SET
	M02	MOVE '00000' TO WORK-RANGE-MAX
#		(None)
1		1-Outstanding
1	M01	MOVE :100.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
2		2-Above Standard
2	M01	MOVE :080.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
3		3-Standard
3	M01	MOVE :050.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
4		4-Needs Improvement
4	M01	MOVE :040.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
5		5-Unsatisfactory
5	M01	MOVE :040.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
7		7-Not Evaluated
7	M01	MOVE :070.00 TO WORK-RANGE-MAX. CODE-SET-DONE.
8		8-Too New To Rate
8	M01	MOVE :010.00 TO WORK-RANGE-MAX. CODE-SET-DONE.

**Example:** MOVE :100.00 TO WORK-RANGE-MAX.  
CODE-SET-DONE.

---

## NOTES

---

## Creating option list logic, continued

### Option list field

The field that is related to the option list must be defined with the option list logic. The field name is defined in the 1st 20 positions of the CSL source, with a code of spaces and a sequence number of 'M' for calculation option list, and 'R' for a relational edit option list.

### Option list logic

The option list logic is related to each code in the option list by repeating the code value and using the sequence field for each unique line of code.

- The sequence number for Calculation Option List logic is 'Mnn' and for Relational Edit logic is 'Rnn', where nn is a sequence number from 01–99.
- Lines of code independent of an option list value are considered common initialization logic and can be added to the blank code value.
- The CODE–SET–DONE verb is used to mark the end of a calculation code routine. It must be coded at the end of the calculation or edit logic for each option list code.

# Compiling Option List Logic

## Calculation Option List Compile

```
COMPILE CALCULATION CODE SET CODE                                RECALC

The RECALC program compiles English Language code found
in a Calculation Code Set file.

Complete the text boxes below, then hit enter: X
Enter Calculation CODE SET NAME: HR16
```

## Relational Edit Option List Compile

```
COMPILE RELATIONAL CODE SET CODE                                REEDIT

The REEDIT program is used to compile relational edit
coding found in calculation Code Sets. All code must
be entered in the Code Set and the RECALC program must
have been run prior to executing the REEDIT program.

Complete the text boxes below, then hit enter: X
Enter the CODE SET NAME: HR16
```

---

## NOTES

---

## Compiling option list logic

### Compiling option list logic

To compile Option List logic, use either the Compile Calculation Option List (RECALC) or Compile Relational Option List (REEDIT) programs.

### Compile calc option list

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** Option Lists  
**Task:**  Compile Calculation Option List

**Result:** The Compile Calculation Code Set form (RECALC) displays.

**Result:** The Option List source is compiled.

### Compile edit option list

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** Option Lists  
**Task:**  Compile Relational Option List

**Result:** The Compile Edit Code Set form (REEDIT) displays.

 *Recompiled option lists must also be RELOADED.*

# Invoking the Option List Logic

**Example:**    CALC-CODE-SET 'HR16 ' .  
                  IF AVERAGE-RATING GREATER THAN WORK-RANGE-MAX  
                  PRINT-REJECT 'HR980' .

---

## NOTES

---

## Calling option list logic

### Calling option list logic

To call the option list logic associated with the option list include either the `CALC-CODE-SET` or `EDIT-CODE-SET` verbs in the relational edit portion of the main program.

#### ■ `CALC-CODE-SET`

The `CALC-CODE-SET` verb is used in source code to call in calculation option list logic (sequence 'Mnn') as an integrated subroutine. `CALC-CODE-SET` must be followed by the name of the calculation option list as a literal, in single quotes.

#### ■ `EDIT-CODE-SET`

The `EDIT-CODE-SET` verb is used in source code to call in edit option list logic (sequence 'Rnn') as an integrated subroutine. `EDIT-CODE-SET` must be followed by the name the calculation option list as a literal, in single quotes.

## Section Summary

- **Option list programming**
- **Components of option list logic**
- **Compiling the option list logic**
- **Invoking the option list logic**

---

NOTES

---

## Section summary

In this section, you learned about calculation and edit option lists. Complete the following question to summarize the section.

### Option list programming

The \_\_\_\_\_ utility is used to maintain option list logic on the System Control Repository. An Object of \_\_\_\_\_ indicates Calc Option List logic, while an Object of \_\_\_\_\_ indicates Relational Edit Option List logic.

### Components of option list logic

For calculation logic, the SEQ field must contain an entry of \_\_\_\_\_ for Calc Option List logic, and \_\_\_\_\_ for Relational Edit Option List logic.

A \_\_\_\_\_ is required as the first entry in the Option List logic.

Each code's logic is terminated with the \_\_\_\_\_ verb.

### Recompiling the option list logic

The \_\_\_\_\_ utility program compiles calculation (M) logic. This process produces 'P' Control File program records identified as \_\_\_\_\_.

The \_\_\_\_\_ utility re-compiles edit (R) logic and produces \_\_\_\_\_ 'P' Control File program records.

### Invoking the option list logic

The \_\_\_\_\_ verb invokes the calculation (M) logic from a main program.

The \_\_\_\_\_ verb invokes the edit (R) logic from a main program.

# Section 3 Exercise

---

**NOTES**

---

## Section 3 exercise

### Purpose

The purpose of this exercise is to create a program that calls option list logic.

1. Modify Option List PP29 by adding logic to calculate an annualized salary for the employees regular pay (HED 001). The annualization calculation is as follows:

$$\text{SALARY} * \text{Annualization Factor (PERM-02-V2)} = \text{Annual Salary (PERM-01-V2)}$$

Frequency	Annualization Factor
1 weekly	52.00
2 bi-weekly	26.00
3 semi-monthly	24.00
4 monthly	12.00

2. Modify the H1-SCR program to display the result of the calculation.

---

**NOTES**

---

## Section 4: User-Defined Table Records

---

### Table of Contents

Introduction.....	75
User-defined table overview .....	77
Table analysis.....	79
Table record layout .....	81
Table field definitions .....	87
Section 4 exercise 1.....	97
User-defined table entry form .....	99
Verify table definitions .....	105
Table record read method.....	109
Section summary .....	113
Section 4 exercise 2.....	117

## Objectives

- **Identify the steps to create user-defined table fields/records**
- **Analyze user-defined data requirements**
- **Create a user-defined table record layout**
- **Create new field definitions for a user-defined table**
- **Create a data entry form for the user-defined table**
- **Verify the user-defined definitions**

---

### NOTES

---

## Introduction

### **Purpose**

In this section, you will learn the steps to create user-defined table fields/records and a program for table record maintenance.

### **Objectives**

Upon completion of this section you will be able to:

- Identify the steps to create user-defined table fields/forms
- Analyze user-defined data requirements
- Create a user-defined table record layout
- Create new field definitions for a user-defined table
- Create a data entry form for the user-defined table
- Verify the user-defined definitions

## **Creating User-Defined Tables**

- 1. Analyze the data requirements**
- 2. Create the record layout**
- 3. Create the field definitions**
- 4. Create the table maintenance form**
- 5. Verify the table record field layout**

---

### **NOTES**

---

## User-defined table overview

### User-defined tables

The Solution Series provides the capability to create user-defined table records. To do so, complete the following procedure:

1. Analyze the data requirements
2. Create the record layout
3. Create the field definitions
4. Create the table maintenance form
5. Verify the table record field layout

# User-Defined Request and Analysis

## MEMO

**To:** Jane Hanson, Programmer

**From:** John Grossman, Human Resource Director

**Subject:** Employee Supervisor Information

We need to have access to each employee's supervisor. I am not familiar with a Solution Series program that readily shows this information. I'd like to be able to have supervisor information available for emergency and personnel purposes. The Payroll department may use it to ensure that an employee received their check or deposit advice.

Since we're tracking an employee's HRMS location, it would be useful to track all of the first-line supervisors in the company based on that. We'd like to be able to revise records, retaining the old records for reference. Perhaps we can develop a coding scheme for all of the supervisors in the company? I'd be glad to help you with that.

Here's the information needed for each supervisor: name, work phone and extension, home phone and the supervisor's mail distribute data. The mail distribute data is essential to check distribution.

Please call me at x4541 if you have any questions.

---

## NOTES

---

## Table analysis

### Data requirements

The 1st step in creating user-defined table fields is to analyze the data requirements. This determines how the data is to be stored.

- Are there existing, delivered techniques to accommodate the requirements?
- Is the table record to be stored historically or is a single version of the table entry sufficient?
- Does the Table data require more than one physical table record?
- What fields uniquely identify one table entry from another?
- Do any fields require validation?
- Are any fields required?
- Is the table's use common or Organization specific?

### Example analysis

Using the memorandum and the analysis questions provides these data requirements for the user-defined Table record:

- There are no existing table records to satisfy this request.
- The table records are to be stored historically, that is, multiple versions of the record are required, and therefore a date is necessary in the key.
- The table data fields do not exceed 80 characters. Only one physical record is required.
- The Location and Supervisor code fields are validated against option lists. The key fields for the table record are required.
- The table data usage is common to all organizations.



## Table record layout

### Table record layout

The 2nd step in creating a user-defined table is to construct a record layout for the user-defined table.

The Record Layout consists of:

- Table Record Key—Record Identifier, Table Identifier, Table Key Fields
- Table Body—Non-key fields

### Record identifier

The Record ID for a user-defined Table record is an 'X'. This recommended convention is to differentiate 'T', 'U', or 'V' table records from user-defined table records.

### Table identifier

The Table ID is any one character that will uniquely identify one user-defined table record from another.

### Table key fields

The table key fields are the final component of the table record key. The fields define the remaining 22 positions of the record key. Key fields can include alphanumeric, numeric fields, and date fields.

### Table data

The table data contains the detail of the table record.

### Multi-part table layout

When more than 79 characters of data are needed, multiple table records can be used. Each table record would have the same key structure, except a one character key separator is used in the last position of the key to uniquely identify each record.

*Note: Data placed in the 80th position will be lost at the next upgrade of The Solution Series, since the 80th position is used for an A, C, or D in MAINTI processing.*



## Table record layout, continued

### Key field design

The order of the key field layout is significant when the table record is read from the System Control Repository. These decisions affect the layout:

- A determination about common or organization number specific usage (control number).
- A determination about whether the table record is to maintain historical entries (table date).

### Key field order

When a control number is included in the table record layout, it should be the first field following the user table identifier and table record identifier. Note that the control number follows the 'X1' literal in the layout above.

### Other key fields

Key fields are laid out in significance order; therefore, the most significant field following the (optional) control number is first in the layout. The least significant is the last.

When a table record entry is historical the key date field is typically the last field in the key field layout. This design allows for 'As of' logic. When the table record is retrieved for use, this layout allows for matching on the more significant key fields.



## Table record layout, continued

### Table layout example

The example above shows the layout for the supervisor table record. Recall that any given System Control Repository record is limited to 80 characters.

The example shows the result of creating the table layout with the information derived from the analysis and memo requirements.

- The record layout shows the physical structure of the record.
- This step is critical in our process for creating user-defined table records, since it will be used to create the field definitions.

*Note: The table data may not be placed in the first 24 positions of the layout; this is reserved for the key fields.*

# Supervisor Control Number

The screenshot shows a 'Field Maintenance And Edit' dialog box for the field 'X1-SV-CTRL-NBR'. The dialog is divided into several sections: 'Field Location' with Pointer: 40, Storage Length: 004, and Displacement: 002; 'Field Options' with Propagate, Rounding, and Header Switch (305132) dropdowns, and a checked 'RDBMS Field' checkbox; and 'Field Properties' with Data Type: Numeric 0 Decimals, Field Type, Template, Lengths (Display and Entry checkboxes), Module, Structure, Seg/Table ID: X1, Table Separator, Codeset, and Edit Routine.

# Supervisor Job Location

The screenshot shows a 'Field Maintenance And Edit' dialog box for the field 'X1-SV-LOCATION-CD'. The dialog is divided into several sections: 'Field Location' with Pointer: 40, Storage Length: 004, and Displacement: 006; 'Field Options' with Propagate, Rounding, and Header Switch (405153) dropdowns, and a checked 'RDBMS Field' checkbox; and 'Field Properties' with Data Type: Alphanumeric, Field Type, Template, Lengths (Display and Entry checkboxes), Module, Structure, Seg/Table ID: X1, Table Separator, Codeset: HR439, and Edit Routine.

---

## NOTES

---

## Table field definitions

### Define fields

The third step is to define the fields in the Field Name Table. This step will be accomplished using Field Name/Maintenance utility (F-NAME).

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** Fields and Verbs  
**Task:**  Define a Field

**Result:** The Field Maintenance and Edit form displays.

- Table fields must be defined in Pointer 40.
- Table key fields do not require a field type of K or P since tables are accessed using direct file processing.

### Field Examples

The table Layout is used to assist in determining the field definitions. The following describes each field's definition:

#### **X1-SV-CTRL-NBR**

This numeric field contains four digits. Its edit length is four positions with zero decimals. It is edited as: 9999.

#### **X1-SV-LOCATION-CD**

This alphanumeric field contains four characters. It is edited against Code Set HR439.

# Supervisor Job Description

The screenshot shows the 'Field Maintenance And Edit' dialog box for the field 'X1-SV-LOCATION'. The dialog is divided into several sections:

- Action:** A small square icon.
- Field Name:** X1-SV-LOCATION
- Field Location:** Pointer: 40, Storage Length: 004, Displacement: 006
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 305132,  RDBMS Field
- Field Properties:** Data Type: Alphanumeric, Field Type: Codeset Description, Template: (dropdown), Lengths: Display: 20, Entry: (checkbox), Module: (dropdown), Structure: (dropdown), Seg/Table ID: X1, Table Separator: (checkbox), Codeset: HR439, Edit Routine: (text field)

# Supervisor Effective Date

The screenshot shows the 'Field Maintenance And Edit' dialog box for the field 'X1-SV-EFFECTIVE'. The dialog is divided into several sections:

- Action:** A small square icon.
- Field Name:** X1-SV-EFFECTIVE
- Field Location:** Pointer: 40, Storage Length: 006, Displacement: 010
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 405210,  RDBMS Field
- Field Properties:** Data Type: Century/Complement D, Field Type: (dropdown), Template: Date, Lengths: Display: 10, Entry: (checkbox), Module: (dropdown), Structure: (dropdown), Seg/Table ID: X1, Table Separator: (checkbox), Codeset: (text field), Edit Routine: (text field)

## NOTES

## Table field definitions, continued

### Field examples

#### **X1-SV-LOCATION**

This alphanumeric field contains four characters. It is edited against option list PPSV.

#### **X1-SV-EFFECTIVE**

This field contains six digits. Its edit length is ten positions, it is edited as: MM-DD-YYYY (US and Canada) or DD-MM-YYYY (elsewhere). This is a century/key date retained on the file in complement form.

# Supervisor Name Code

The screenshot shows the 'Field Maintenance And Edit' dialog box. The 'Field Name' is 'X1-SV-NAME-CODE'. The 'Field Location' section shows Pointer: 40, Storage Length: 004, and Displacement: 024. The 'Field Options' section includes Propagate, Rounding, and Header Switch: 110432, with a checked 'RDBMS Field' option. The 'Field Properties' section shows Data Type: Alphanumeric, Field Type, Template, Lengths (Display: , Entry: ), Module, Structure, Seg/Table ID: X1, Table Separator: , Codeset: PPSV, and Edit Routine.

# Supervisor Name

The screenshot shows the 'Field Maintenance And Edit' dialog box. The 'Field Name' is 'X1-SV-NAME'. The 'Field Location' section shows Pointer: 40, Storage Length: 004, and Displacement: 024. The 'Field Options' section includes Propagate, Rounding, and Header Switch: 100610, with an unchecked 'RDBMS Field' option. The 'Field Properties' section shows Data Type: Alphanumeric, Field Type: Codeset Description, Template, Lengths (Display: 20, Entry: ), Module, Structure, Seg/Table ID: X1, Table Separator: , Codeset: PPSV, and Edit Routine.

## NOTES

## Table field definitions, continued

### Field examples

#### **X1-SV-NAME-CODE**

This alphanumeric field contains four characters. It is edited against option list PPSV.

#### **X1-SV-NAME**

This alphanumeric field contains 20 characters. It is the description of the value found in option list PPSV.

# Supervisor Work Phone

The screenshot shows a 'Field Maintenance And Edit' dialog box for the field 'X1-SV-WORK-PHONE'. The dialog is divided into several sections:

- Action:** A dropdown menu.
- Field Name:** A text box containing 'X1-SV-WORK-PHONE'.
- Field Location:** A section containing:
  - Pointer:** 40
  - Storage Length:** 010
  - Displacement:** 028
- Field Options:** A section containing:
  - Propagate:** A dropdown menu.
  - Rounding:** A dropdown menu.
  - Header Switch:** 110410
  - RDBMS Field
- Field Properties:** A section containing:
  - Data Type:** Numeric 0 Decimals
  - Field Type:** A dropdown menu.
  - Template:** A dropdown menu.
  - Lengths:** Display:  Entry:
  - Module:** A dropdown menu.
  - Structure:** A dropdown menu.
  - Seg/Table ID:** X1
  - Table Separator:** A dropdown menu.
  - Codeset:** A dropdown menu.
  - Edit Routine:** A text box.

# Supervisor Work Extension

The screenshot shows a 'Field Maintenance And Edit' dialog box for the field 'X1-SV-WORK-EXT'. The dialog is divided into several sections:

- Action:** A dropdown menu.
- Field Name:** A text box containing 'X1-SV-WORK-EXT'.
- Field Location:** A section containing:
  - Pointer:** 40
  - Storage Length:** 004
  - Displacement:** 038
- Field Options:** A section containing:
  - Propagate:** A dropdown menu.
  - Rounding:** A dropdown menu.
  - Header Switch:** 305132
  - RDBMS Field
- Field Properties:** A section containing:
  - Data Type:** Numeric 0 Decimals
  - Field Type:** A dropdown menu.
  - Template:** A dropdown menu.
  - Lengths:** Display:  Entry:
  - Module:** A dropdown menu.
  - Structure:** A dropdown menu.
  - Seg/Table ID:** X1
  - Table Separator:** A dropdown menu.
  - Codeset:** A dropdown menu.
  - Edit Routine:** A text box.

---

## NOTES

---

## Table field definitions, continued

### Field examples

#### **X1-SV-WORK-PHONE**

This numeric field contains ten digits. Its edit length is ten positions with zero decimals. It is edited as: 9999999999.

#### **X1-SV-WORK-EXT**

This numeric field contains four digits. Its edit length is four positions with zero decimals. It is edited as: 9999.

# Supervisor Home Phone

The screenshot shows the 'Field Maintenance And Edit' dialog box for the field 'X1-SV-HOME-PHONE'. The dialog is divided into several sections:

- Action:** A dropdown menu.
- Field Name:** A text box containing 'X1-SV-HOME-PHONE'.
- Field Location:** A section containing:
  - Pointer:** 40
  - Storage Length:** 010
  - Displacement:** 042
- Field Options:** A section containing:
  - Propagate:** A dropdown menu.
  - Rounding:** A dropdown menu.
  - Header Switch:** 110443
  - RDBMS Field
- Field Properties:** A section containing:
  - Data Type:** Numeric 0 Decimals
  - Field Type:** A dropdown menu.
  - Template:** A dropdown menu.
  - Lengths:** Display:  Entry:
  - Module:** A dropdown menu.
  - Structure:** A dropdown menu.
  - Seg/Table ID:** X1
  - Table Separator:** A dropdown menu.
  - Codeset:** A dropdown menu.
  - Edit Routine:** A text box.

# Supervisor Mail Distribution

The screenshot shows the 'Field Maintenance And Edit' dialog box for the field 'X1-SV-MAIL-DIST'. The dialog is divided into several sections:

- Action:** A dropdown menu.
- Field Name:** A text box containing 'X1-SV-MAIL-DIST'.
- Field Location:** A section containing:
  - Pointer:** 40
  - Storage Length:** 010
  - Displacement:** 052
- Field Options:** A section containing:
  - Propagate:** A dropdown menu.
  - Rounding:** A dropdown menu.
  - Header Switch:** 110464
  - RDBMS Field
- Field Properties:** A section containing:
  - Data Type:** Alphanumeric
  - Field Type:** A dropdown menu.
  - Template:** A dropdown menu.
  - Lengths:** Display:  Entry:
  - Module:** A dropdown menu.
  - Structure:** A dropdown menu.
  - Seg/Table ID:** X1
  - Table Separator:** A dropdown menu.
  - Codeset:** A dropdown menu.
  - Edit Routine:** A text box.

---

## NOTES

---

## Table field definitions, continued

### Field examples

#### **X1-SV-HOME-PHONE**

This alphanumeric field contains ten digits. Its edit length is ten positions with zero decimals. It is edited as: 9999999999.

#### **X1-SV-MAIL-DIST**

This alphanumeric field contains ten characters; it is edited as: 9999999999.

### Multi part table fields

Multi part table records require the setup of field definitions for the key fields in the second and subsequent records. These values are pre filled by the table form program. The table data fields are defined in starting displacement 104 of pointer 40 for the second record.



## Section 4 exercise 1

### Directions

Your organization has a need to store the number of days in each pay period for each frequency code. More than one frequency identifier has been established for use (ex: Weekly vs. Semi monthly). Each company within the organization may have a different scheme for frequency code meaning. Design a table record layout and field definitions for this purpose.

### Layout

Use the form on the opposite page to complete the table layout.

Field Use	Field Type/Length
Record Identifier*	X
Table Identifier*	Two
Control Number*	Numeric, four positions
Table Pay Frequency*	Alphanumeric, one position, Option List=PP29
Table Period End Date*	Century Date, six positions
Table Workdays by Period	Numeric, three positions
Table Holidays by Period	Numeric, three positions

\*Key fields

## **Create the Table Maintenance Form**

- **Form design using the form design application**
- **CSL form program**

---

### **NOTES**

---

## User-defined table entry form

### Data entry form

The 4th step in creating a user-defined table is to create the data entry form for the table. This requires two development tasks:

### The form design application

The form design application is used to design the form appearance. There is little difference between the way you use Form Builder to design company or employee forms and table forms. These minor differences are discussed on the following pages.

### CSL

Table forms are coded very differently than company or employee forms. This is primarily because the table data is stored in the System Control Repository, and there are a limited number of macro verbs available for programming table forms.

To simplify the development of the CSL program used in a table form, a template program will be used to explain and develop the form. A copy of the template can be found in Appendix B: Extra for Experts.

# Form Designs

## Entry Mode Design

Form Builder - XSVSCR.SAT  
File Form Mode Add Control

Supervisor Table Screen Control Number> XXXX

Location Supervised> XXXXXXXXXXXXXXXXXXXX  
Effective Date> XXXXXXXXXX  
Name: XXXXXXXXXXXXXXXXXXXX  
Work Phone: XXXXXXXXXX Ext. : XXXX  
Home Phone: XXXXXXXXXX  
Mail Distribution: XXXXXXXXXX

## Inquiry/Select Mode Design

Form Builder - XSVSCR.SAT  
File Form Mode Add Control

Supervisor Table Screen Control Number> XXXX

Location Supervised	Effective Date	Name	Work Phone	Home Phone	Mail Distribution
XXXX	XXXXXXXXXX	XXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX

## NOTES

## User-defined table entry form, continued

### Form design

The three primary components of a form are:

- **Form Header**  
Used to provide a form title as well as other informative information on the form.
- **Form Body**  
Used to provide either update or inquiry access to a field. Fields can be represented as Edit Boxes, Inquiry Fields, Radio Buttons, Check Boxes, or List Boxes.
- **Context-sensitive menus**  
Contains prompts which are used to provide a link to the next logical form.

### Entry mode design

Regardless of your form design, layout key fields must adhere to the following rules:

- All key fields must be grouped in table layout order.
- The last key field **MUST** indicate paragraph 850 in the exit routine After field of the Entry Dialog.
- All key field labels will end with a '>' (greater than) symbol denoting the fact that the field is a key.
- Non-key data follows the key fields.
- An empty row should exist between the last key field and the first non-key data field on the form, when possible.
- The CONTROL-NUMBER must be in Screen Section '5' at the top right corner of the form, in inquiry.
- If needed, define the key separator field at displacement 23 in each record but do not include it on the form layout.

### Select/Inquiry mode design

The format of the Select/Inquiry form section must contain the following format:

- Headings appearing on two lines above the entry field with the data displayed inside the unprotected areas.
- All key fields must appear in order from left to right as entry fields.
- One non-key field must appear immediately following the key fields as an entry field.
- All remaining non-key fields that will fit onto one line will appear as inquiry fields.
- It is a requirement that labels are painted on lines 5/6 with the data fields falling on line 7.

# Table forms

## Example of single part table form - TE-SCR

Occupation Group Table Control Number> 9999

Occupation Group> Legislators

Effective Date> 01-01-1925

Job Family: Exec/Admn/Mangr

EEO Job Categories

EEO-1: Officials/Managers	EEO-4: Officials/Admins
EEO-5: No Code Entered	IPEDS: No Code Entered

## Example of two part table form - TA-SCR

Job Code Table Control Number> 9999

Job Code> 10500

Job Code Extent> 0001

Effective Date> 01-01-2002

Job Title: CALL CENTER OPERATOR

Salary Grade: A11

Workers Comp Code: SUP50

Survey Indicators:

Occupation Group: Telephone Operators

Assigned Shift: 1st Shift

Job Type: Hourly Union

Union Job:

Trainee Job: Not A Trainee Job

FLSA Classification

Exempt

Non-exempt

Previous Job

Job Code:

Extent:

## NOTES

## User-defined table entry form, continued

### Creating a table form

To simplify the development of the CSL program used in a table form, you could find an existing table form that is similar to your requirements and copy the code. You can then make changes to that code to get the results you require.

A suggestion would be to use the existing code from the Occupation Group Table (TE-SCR) for single part tables and the code from the Job Code Table (TA-SCR) for two part tables.

Make and test small corrections at a time to the new table program until it is correct. If it is still not working correctly, use the TRACE utility to see where the problems lie. Details on the TRACE utility can be found in Appendix C: The TRACE Utility.

# Table Data Entry Form and Verification

## Entry Mode

Supervisor Table Screen Control Number>9999

Location Supervised> Region 3333

Effective Date> 01-01-1990

Name: William Haze

Work Phone: 3124541865 Ext.: 0045

Home Phone: 3125557898

Mail Distribution: 02A-33

## Inquiry/Select Mode

	Location supervised	Effective Date	Name	Work Phone	Home Phone	Mail Distribution
<input type="checkbox"/>	3030	03-26-1998	JOHN	3124541865	3125551579	13H-72
<input type="checkbox"/>	3333	01-01-1990	BILL	3124541865	3125557898	02A-33
<input type="checkbox"/>	3388	01-01-1999	JACK	3124541865	3125551313	13D-56

---

## NOTES

---

## Verify table definitions

### Verify form and fields

After creating the table form it is necessary to verify that the program functions properly. Verification includes:

- Key fields are the first Entry field on the form, and are in table layout order.
- Key fields are required.
- Valid values can be entered into each field:
  - Date fields require data in the format YYMMDD or MM-DD-YY (US and Canada) or YYDDMM or DD-MM-YY (elsewhere).
  - Numeric fields accept the proper number of integers and decimals. Results are displayed to the form properly.
  - Name fields require data in 'Last, First' format.
  - Required fields must be entered, otherwise an error occurs.
  - Option list fields are edited against an option list.
  - Option list description fields display the description properly.
- Enter at least three table records and test the selection of the top, bottom, and middle records.

# Verify the Table Layout

```
DISPLAY CONTROL FILE                                DSP01

The DSP01 screen may be used to view any of the records on
FILE01 except security and object code records. For object
code records a count is provided.

Complete the text boxes below, then hit enter:  X
Enter STARTING KEY: X1
or enter START to view the beginning of the file.
```

```
0...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
X199993030201J06      JOHN31245418650013312555157913H-72
X199993333201L31      BILL31245418650045312555789802A-33
X199993388200L31      JACK31245418650010312555131313D-56
----End of file----
```

---

## NOTES

---

## Verify table definitions, continued

### Verify table layout

The final verification step is to match your original hard copy Table record layout to the physical layout of the record.

The Display System Control Repository utility (DSP01) allows you to view System Control Repository records.

To access the Display System Control Repository utility (DSP01):

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** System Control Repository Utilities  
**Task:**  List System Control Repository

**Result:** The Display System Control Repository HELP displays.

1. Type a System Control Repository record key, for example X1 (a partial key is valid).
2. Press Enter.

**Result:** The Display Control File utility (DSP01) displays the requested System Control Repository records.

## Table READ– Verbs

CREATE A READ VERB		
VERB NAME:	<input type="text" value="X1-SV-READ-TABLE"/>	FILE NUMBER: <input type="text" value="01"/> MODULE CODE: <input type="text" value="PP"/>
1ST FIELD:	<input type="text" value="'X1'"/>	
2ND FIELD:	<input type="text" value="JOB-CODES"/>	
3RD FIELD:	<input type="text" value="CTRL-THREE"/>	
4TH FIELD:	<input type="text" value="LOCATION-EFFECTIVE"/>	
5TH FIELD:	<input type="text"/>	
6TH FIELD:	<input type="text"/>	
7TH FIELD:	<input type="text"/>	
8TH FIELD:	<input type="text"/>	

---

### NOTES

---

## Table record read method

### READ– verb utility

Recall that the Create a READ– Verb (RDVERB) utility is used to create a read macro verb. In our example, we will create a READ– verb to access a specific supervisor table record.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Fields and Verbs
<b>Task:</b>		Create A READ Verb

**Result:** The Create A READ Verb form (RDVERB) displays.

### Supervisor read verb

To create the Supervisor read verb, which will be used to access the supervisor information for a specific employee, the table key fields must be built using the employee’s location information.

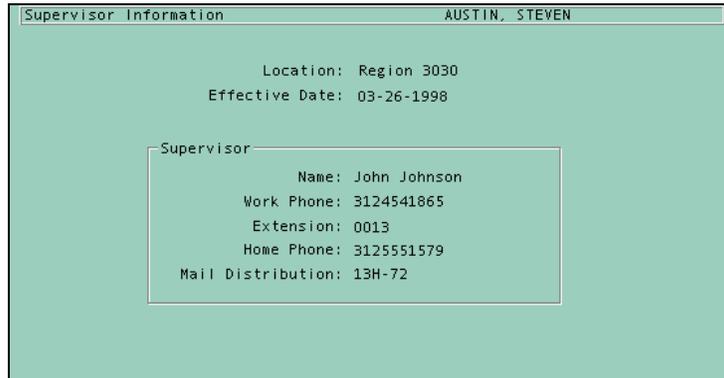
- **VERB NAME—X1–SV–READ–TABLE** is the name of the verb that is created by the utility for your use. Recall that an ‘X’ in the 1st position is used to distinguish user-defined verbs from delivered verbs.
- **FILE NUMBER—**The FILE NUMBER field contains 01 for the System Control Repository.
- **MODULE CODE—**The MODULE CODE field specifies the PP Payroll/Personnel application that the verb/records reside within.
- **1ST FIELD—**The literal ‘X1’ is the Record ID and Table ID.
- **2ND FIELD—**The JOB–CODES cross-reference field.
- **3RD FIELD—**The HRMS Location CTRL–THREE for the employee.
- **4TH FIELD—**The HRMS Location LOCATION–EFFECTIVE for the employee.
- **5TH-8TH FIELD—**Unused.

### FILE01 READ– verb

Recall the functions that the READ- verb performs:

- Builds the FILE01 key from the 1ST–8TH FIELD literal and field names.
- Reads FILE01.
- If the record is found (STAT-KEY=00), the data is placed into pointer 40 starting at displacement 000. Otherwise, pointer 40 is initialized to spaces in the first 80 displacements.

# Using X1-SV-READ-TABLE



```
P100-START.
  READ-EMPLOYEE.
  SCREEN-SECTION '0'.
P150-FIND-LOCATION.
  FIND-LOCATION.
  IF NOT FOUND
  PRINT 'No Location Found' RETURN.
P200-READ-CROSS-REFERENCE-TABLE.
  MOVE CURRENT-DATE TO CROSS-REFERENCE-DATE.
  READ-TZAX-TABLE.
P300-READ-X1-TABLE.
  X1-SV-READ-TABLE.
  SCREEN-SECTION '1'.
```

---

## NOTES

---

## Table record read method, continued

### Example

Before the X1-SV-READ-TABLE verb can be used, several fields used in the verb must be accessed:

- The Cross-Reference Table record is read after executing the READ-TZAX-TABLE verb. This verb requires a valid date in the field CROSS-REFERENCE-DATE before it is executed.
- The HRMS Location segment for the employee must be accessed to obtain the CTRL-THREE and LOCATION-EFFECTIVE data.

### Read status

The code generated for the FILE01 read depends on the length and contents of the key fields named in the 1ST-8TH field parameters:

- If the key data does not include a century/complement date as the last key field, the total key length is used to read the record and verify that the record was found.
- If a century/complement date is the last of the key fields, the total key length is used for the read, however only the key fields up to but not including the date field are used to determine if the record was found. This allows for an 'As of' technique for reading records.

If the record is found, the data is moved into Pointer 40, otherwise spaces are moved to Pointer 40.

### Multi-part records

To read a multi-part record, execute a READ FILE01 statement. The guidelines for the READ verb are in Section 2 of this documentation.

## Section Summary

- **User-defined segments overview**
- **Data requirements analysis**
- **User-defined table layout**

---

### NOTES

---

## Section summary

### User-defined table overview

The steps to create user-defined Table records include:

1. \_\_\_\_\_
2. \_\_\_\_\_
3. \_\_\_\_\_
4. \_\_\_\_\_
5. \_\_\_\_\_

### Data requirements analysis

Table key design must consider a common or \_\_\_\_\_ use.

Each Table record cannot exceed \_\_\_\_\_ characters.

If the Table usage requires a history of information, a \_\_\_\_\_ field is the last key field design.

### User-defined table layout

A Table's record layout consists of a \_\_\_\_\_ and \_\_\_\_\_.

The Table Record Key consists of a \_\_\_\_\_, a \_\_\_\_\_, and \_\_\_\_\_.

The Record Identifier value is \_\_\_\_\_.

The Table Key fields are placed in displacements \_\_\_\_\_ and the Table Data starts in displacement \_\_\_\_\_.

Multi-part Tables must designate a \_\_\_\_\_ field.

## **Section Summary, continued**

- **User-defined field definitions**
- **User-defined entry form**
- **Verify user-defined definitions**

---

### **NOTES**

---

## Section summary, continued

### User-defined field definitions

Table record field definitions use pointer \_\_\_\_\_.

The 1st Table Key Field starts in displacement \_\_\_\_\_.

When a Control Number is used, it is the \_\_\_\_\_ Table Key field.

When a date is used, it is typically the \_\_\_\_\_ Table Key field unless a Table Key Separator is incorporated for multi-part records.

The Table Data fields of a multi-part record start at displacement \_\_\_\_\_.

### User-defined entry form

The last key field defined in the Form Builder design must perform Paragraph \_\_\_\_\_ after the field is displayed.

### Table record read method

The \_\_\_\_\_ utility can be used to create Table record read logic.

## Section 4 Exercise 2

---

**NOTES**

---

## Section 4 exercise 2

### Directions

Complete the Pay Period Days Table by designing the form using the form design application. Write a program to display the number of paydays and workdays in a pay period. Use the following steps as your development guidelines:

1. Use the form design application to create the form's appearance using the fields defined in Section 4 Exercise 1. Do not forget to Perform paragraph 850 after the date field.
2. Test the form by entering data into the table and verifying the entry and inquiry/select modes. Additionally, match the layout defined in Section 4, Exercise 1 to the layout of each record stored in FILE01 (DSP01 utility).

---

**NOTES**

---

## Section 5: Employee Database Updating

---

### Table of Contents

Overview .....	121
Inserting a segment .....	123
Deleting a segment .....	131
Batch transaction updating .....	135
Section summary .....	139
Section 5 exercise .....	141

## Objectives

- **Recognize the syntax and use of INSERT– verbs**
- **Recognize the syntax and use of DELETE– verbs**
- **Describe the process of batch transactions updating**

---

### NOTES

---

## Overview

### **Purpose**

In this section you will learn alternative techniques of Employee Database updating.

### **Objectives**

When you complete this section you will be able to:

- Recognize the syntax and use of INSERT– verbs
- Recognize the syntax and use of DELETE– verbs
- Describe the process of batch transactions updating

# INSERT– Verbs

Verb	Segment Type	Segment data
INSERT–B–SEGMENT	B	Company Hours, Earnings & Deductions
INSERT–C–SEGMENT	C	Other Company Information
INSERT–D–SEGMENT	D	Batch Payroll Report Requests
INSERT–F–SEGMENT	F	Employee Name and Address
INSERT–G–SEGMENT or INSERT–PTR31	G	Employee Labor Splits
INSERT–H–SEGMENT or INSERT–PTR32	H	Employee Hours, Earnings & Deductions
INSERT–J–SEGMENT or INSERT–PTR34	J	Employee Taxes
INSERT–L–SEGMENT or INSERT–PTR36	L	Human Resource/User Employee Data
INSERT–P–SEGMENT	P	Employee Period End
INSERT–PTR36–BATCH	L	Human Resource/User Employee Data (Batch)

---

## NOTES

---

## Inserting a segment

### Segment updating

In addition to updating segments using a form, you can accomplish updates using the INSERT- verb. This verb is used in:

- Form programs
- Packaged report programs
- Query programs

### INSERT rules

The rules for INSERT- include:

- Initialize the Pointer 8 work area.
- Create the segment layout in field displacement order in Pointer 8, starting at displacement 000.
- Position the pointer to the segment to be inserted using a FIND or SET verb before the INSERT.

### INSERT- verbs

The INSERT- verbs are used for a particular Segment Type. All of the INSERT- verbs recalculate the pointer addresses of the segments that follow them except for INSERT-PTR36. INSERT-L-SEGMENT should be used instead of INSERT-PTR36. INSERT-PTR36-BATCH should be used in batch programs to initialize SCREEN-ERROR to 'F' prior to the insert.

### Initialization

The INITIAL-SEGMENT-AREA verb initializes the first 110 positions of pointer 8 to spaces. INITIAL-x-SEGMENT verbs initialize the segment type and that segment's fields to either spaces or zeroes, as appropriate, starting at displacement 000 of pointer 8.

# INSERT- Form Example

## Example 1: Inserting an L Segment from the Performance Appraisal Results (49-SCR)

	UPDATE-EMPLOYEE. : : P300-VERIFY. SET-FOR-MESSAGES. IF ERRORS-EXIST RETURN. IF RECORD-NOT-UPDATED GO TO P999-PROMPTS. : : IF APPRAISAL-TYPE EQUALS 'JA' MOVE '010000' TO WORK-TIME-SPAN CALCULATE RATING-DATE + WORK-TIME-SPAN = HOLD-DATE
<b>Calculate Next Appraisal Date:</b>	
<b>Position Pointer:</b>	FIND NEXT-APPRAISAL-TYPE STARTING WITH HOLD-DATE IF FOUND RETURN ELSE
<b>Initialize Ptr 8: Build Segment:</b>	INITIAL-SEGMENT-AREA MOVE 'LZS' TO W8-03-000 MOVE HOLD-DATE TO W8-06-003 MOVE 'JA' TO W8-02-009
<b>Insert Segment:</b>	INSERT-L-SEGMENT. : :

## Example 2: Update a G Segment from the Location Assignment/Changes (05CSCR)

	UPDATE-EMPLOYEE. : : P300-VERIFY. SET-FOR-MESSAGES. IF ERRORS-EXIST RETURN. IF RECORD-NOT-UPDATED GO TO P999-PROMPTS. : : READ-TG-TABLE. IF STAT-KEY GREATER THAN '01' RETURN. FIND CONTROL-3 IF NOT FOUND RETURN. IF CONTROL-THREE-SWITCH EQUAL '3' MOVE CTRL-THREE TO CONTROL-3. IF CONTROL-FOUR-SWITCH EQUAL '3' MOVE CTRL-FOUR TO CONTROL-4. IF CONTROL-FIVE-SWITCH EQUAL '3' MOVE CTRL-FIVE TO CONTROL-5. IF CONTROL-SIX-SWITCH EQUAL '3' MOVE CTRL-SIX TO CONTROL-6.
<b>Read System Options: Position Pointer:</b>	
<b>Update Segment:</b>	

---

## NOTES

---

## Inserting a segment, continued

### Form example

Recall, that updating a segment using a form is automatic when the UPDATE– and ENTRY verbs are used. However, to create other segments within a different segment or additional segments within the same segment the INSERT– logic is used. Use INSERT– only after the verification logic has executed.

### Example 1

This example shows inserting the next performance appraisal segment from the Performance Appraisal Results form (49–SCR).

- The FIND positions the pointer where the segment is to be inserted using the segment key value.
- If the segment is not found the segment is built in pointer 8 with the segment type, Segment Code, Segment Key and segment fields. Then the segment is inserted into the record, and is updated to the file by the UPDATE–EMPLOYEE logic.

### Example 2

This example shows an update to the Payroll Home Location/Pay Allocations Segment from the Location Assignment/Changes form (05CSCR):

- Once the Pointer is positioned, each non–key field may be changed using their field names.

*Warning* No error messages display as a result of a segment inserted with an invalid layout!

# INSERT– Report Example

## Example 1: Inserting an L Segment

<b>Initialize Pointer 8:</b>	INITIAL-SEGMENT-AREA.
<b>Build Segment:</b>	MOVE 'LXX' TO W8-03-000. MOVE WORK-DATE TO W8-06-003. MOVE XNEWDATA-FIELD TO W8-30-009.
<b>Position Pointer:</b>	FIND XLSEG-FIELD STARTING WITH W8-06-003. IF FOUND RETURN.
<b>Insert Segment:</b>	INSERT-PTR36-BATCH.
<b>Test the Insert:</b>	IF SCREEN-ERROR EQUALS 'F'
<b>Record Write:</b>	MOVE 'Y' TO RECORD-UPDATED. : :

## Example 2: Update an L Segment

<b>Position Pointer</b>	FIND XLSEG-FIELD STARTING WITH W6-06-036. IF FOUND
<b>Update Segment</b>	MOVE CURRENT-DATE-CYYMDD TO XMY-DATE MOVE '00' TO XMY-FIELD
<b>Record Write:</b>	MOVE 'Y' TO RECORD-UPDATED. : :

---

## NOTES

---

## Inserting a segment, continued

### Report example

Inserting a segment(s) from a report program has the same INSERT- rules as stated earlier.

### INSERT-**PTR36 BATCH**

This verb inserts a new Pointer 36 Segment occurrence using the data in the first 71 positions of pointer 8. It is inserted in the Segment stack based on the Segment pointer address. This verb also initializes the system field SCREEN-ERROR to an 'F'.

### Record **WRITE**

When using a REPORT program to update segments, two system level fields must be considered:

- The SCREEN-ERROR field is used to determine if the segment insert was successful. 'F' = successful update, 'Y' = unsuccessful update.
- The RECORD-UPDATED field must be set to 'Y' to rewrite the record to FILE02.

### Example 1

This example demonstrates inserting a new segment:

- The segment is built in pointer 8 with the Segment type, segment code, segment key and fields.
- The FIND positions the pointer where the segment is to be inserted using segment key value. If the segment is not found the segment is inserted into the record in memory, and will be updated to the file by the report process.

### Example 2

This example demonstrates updating an existing segment:

- Once the Pointer is positioned, each non-key field may be changed using their field names.

# INSERT– Query Example

## Example 1: Insert/Update an H Segment Using UPDATE–EMPLOYEE

**Update Record:** UPDATE-EMPLOYEE.  
**Selection:** FIND RESULTING-EMP-STATUS.  
IF NOT FOUND  
UNLOCK-EMPLOYEE RETURN.  
IF RESULTING EMP-STATUS NOT EQUAL '0'  
UNLOCK-EMPLOYEE RETURN.

**Position Pointer:** FIND HED-AMOUNT-YTD STARTING WITH '008'.  
IF FOUND

**Update Fields:** MOVE '17' TO FREQUENCY-CODE  
ELSE

**Build Segment:** INITIAL-H-SEGMENT  
MOVE 'H' TO W8-01-000  
MOVE '008' TO W8-03-001HED  
MOVE '17' TO W8-02-004FREQ  
@MOVE COMP VALUES TO TO-DATE FIELDS

**Insert Segment:** INSERT-H-SEGMENT.  
**Write Record:** WRITE-EMPLOYEE.

## Example 2: Insert/Update an H Segment Using READ-EMPLOYEE

**Read Record:** READ-EMPLOYEE.  
**Selection:** FIND RESULTING-EMP-STATUS.  
IF NOT FOUND  
RETURN.  
IF RESULTING-EMP-STATUS NOT EQUAL '0'  
RETURN.

**Position Pointer:** FIND HED-AMOUNT-YTD STARTING WITH '008'.  
IF FOUND

**Update Fields:** MOVE '17' TO FREQUENCY-CODE.  
ELSE

**Build Segment:** INITIAL-H-SEGMENT  
MOVE 'H' TO W8-01-000  
MOVE '008' TO W8-03-001HED  
MOVE '17' TO W8-02-004FREQ  
@MOVE COMP VALUES TO TO-DATE FIELDS

**Insert Segment:** INSERT-H-SEGMENT.  
**Write Record:** WRITE-EMPLOYEE.

---

## NOTES

---

## Inserting a segment, continued

### Query example

Inserting a segment from a Query program has the same INSERT- rules as stated earlier.

### UPDATE- verbs

When the UPDATE- verbs are used in a Query program, it assures that no other access to the Employee Database record is possible. If the record is not to be updated you must UNLOCK- the record.

### UNLOCK- verbs

The UNLOCK- verbs are used to delete a ZL record from FILE01 which allows access to a record by other programs.

### WRITE- verbs

The WRITE- verbs are used to update a Company, Employee or Tax record on FILE02. Issue the WRITE- verb only once for each master file record you are updating, not for each segment that you insert.

### Example 1

In this example, the UPDATE- verb is used to insert/update an H segment into the employee record. If the employee is not to be updated the record must be unlocked, otherwise the segment is either updated or inserted, and then written to the file using the WRITE- verb.

### Example 2

In this example, the READ- verb is used to insert/update an H segment into the employee record. If the employee is not to be updated no unlocking is necessary, the segment is either updated or inserted, and then written to the file using the WRITE- verb.

# DELETE– Verbs

Verb	Segment Type	Segment data
DELETE–B–SEGMENT	B	Company Hours, Earnings & Deductions
DELETE–C–SEGMENT	C	Other Company Information
DELETE–D–SEGMENT	D	Batch Payroll Report Requests
DELETE–F–SEGMENT	F	Employee Name and Address
DELETE–G–SEGMENT	G	Employee Labor Splits
DELETE–H–SEGMENT	H	Employee Hours, Earnings & Deductions
DELETE–J–SEGMENT	J	Employee Taxes
DELETE–L–SEGMENT	L	Human Resource/User Employee Data
DELETE–P–SEGMENT	P	Employee Period End

**Example: Delete continuation segments from the Spouse/Dependent Information (10–SCR)**

```

UPDATE–EMPLOYEE .
      :
P300–VERIFY .
      SET–FOR–MESSAGES .
      IF ERRORS–EXIST RETURN .
      IF RECORD–NOT–UPDATED GO TO
      P999–PROMPTS .
      :
P900–DELETE–SEGMENT .
      IF W8–01–395 NOT EQUAL 'Z' EXIT .
      FIND OTHER–ADDRESS–1 STARTING WITH W7–03–081 .
      IF FOUND PERFORM P910–DELETE .
      FIND OTHER–ADDRESS–2 STARTING WITH W7–03–081 .
      IF NOT–FOUND EXIT .
P910–DELETE .
      DELETE–L–SEGMENT . EXIT .
    
```

**Delete This Entry?:**

**Position Ptr:**

**Position Ptr:**

**Delete Segment:**

---

## NOTES

---

## Deleting a segment

### Segment deletion

A segment can be deleted from an Employee Database record using one of the DELETE- verbs listed in the table. These verbs may be used in:

- Form program
- Packaged report program
- Query program

### DELETE rules

The rules for a DELETE- include:

- Position the pointer to the segment to be deleted using a FIND or SET verb before the DELETE.

### Form example

This example shows the deletion of the continuation segments for the Spouse/Dependent Information form (10-SCR): P900 must be an After paragraph on the last key field of the form:

- W8-01-395 is check for the value 'Z', which signifies that the user selected Delete This Entry (ZDELETE) from the Action menu.
- The pointer is positioned to the continuation segments.
- The DELETE- verb is executed.

## DELETE– Report Example

### Example: Delete a Segment using a Report Program

```
          :           :
Position Ptr:      FIND CTRL-3-CODE STARTING WITH '03'.
                   IF NOT-FOUND RETURN.
Delete Segment    DELETE-G-SEGMENT.
Insure Write:     MOVE 'Y' TO RECORD-UPDATED.
                   :           :
```

## DELETE– Query Example

### Example 1: Delete a Segment Using Query Program and UPDATE– verb

```
Read Record:      UPDATE-EMPLOYEE.
Position Pointer:  FIND HED-AMOUNT-YTD STARTING WITH '008'.
                   IF FOUND
Delete Segment:    DELETE-H-SEGMENT
Write Record:     WRITE-EMPLOYEE
                   ELSE
Unlock Record:    UNLOCK-EMPLOYEE.
```

### Example 2: Delete a Segment Using Query Program and READ– verb

```
Read Record:      READ-EMPLOYEE.
Position Pointer:  FIND HED-AMOUNT-YTD STARTING WITH '008'.
                   IF FOUND
Delete Segment:    DELETE-H-SEGMENT
Write Record:     WRITE-EMPLOYEE.
```

---

## NOTES

---

## Deleting a segment, continued

### Report example

When using a Report program to delete segments, the RECORD-UPDATED field must be set to 'Y' to rewrite the record to FILE02.

### Query program example 1

In the 1st example, the UPDATE- verb is used to retrieve the employee record. If the employee is not to be updated, the record must be unlocked, otherwise delete the segment and rewrite the record to FILE02 using the WRITE-verb.

### Query program example 2

In the 2nd example, the READ- verb is used to retrieve the employee record. If the employee is not to be updated, the record must be unlocked, otherwise delete the segment and rewrite the record to FILE02 using the WRITE- verb.

*Note: Refer to Inserting a Segment, Query example for additional details of using UPDATE- versus READ- verbs in a Query program.*

# Batch Transaction Updating

Batch Layout Report for Scheduled Salary Review Data  
 Change mode regular display 43-SCR layout for United States

From	To	Field Name	Length	Comments	Format/Edit
=====					
1	8	Program Literal	008	P CONTRL	Constant
9	14	Task Number	006	T00010	Constant
15	15	Filler	001	Space	Constant
16	16	Comm-Cancel	001	Space	Constant
17	22	Company Number	006	999999	Alphanumeric
23	28	Program Field	006	43-SCR	Constant
29	29	Code-1	001	Space	Constant
30	30	Code-2	001	Space	Constant
31	40	Key Field	010		Alphanumeric
41	55	Additional Key	015		Alphanumeric
56	65	NEXT-REVIEW-DATE	010		MM-DD-YYYY
66	67	NEXT-REVIEW-TYPE	002		HR34
68	74	DISTRIBUTION-DATA	010	Pos 001-007	Alphanumeric
75	75	Continuation-Ind	001	*	Constant
1	8	Program Literal	008	P CONTRL	Constant
9	14	Task Number	006	T00020	Constant
15	15	Filler	001	Space	Constant
16	18	DISTRIBUTION-DATA	010	Pos 008-010	Alphanumeric
19	28	DISTRIBUTION-DATE	010		MM-DD-YYYY
29	58	REVIEWER-NAME	030		
59	68	DATE-RETURN-EXPECTED	010		MM-DD-YYYY
69	74	ACTUAL-RETURN-DATE	010	Pos 001-006	MM-DD-YYYY
75	75	Continuation-Ind	001	*	Constant
1	8	Program Literal	008	P CONTRL	Constant
9	14	Task Number	006	T00030	Constant
15	15	Filler	001	Space	Constant
16	19	ACTUAL-RETURN-DATE	010	Pos 007-010	MM-DD-YYYY

## NOTES

## Batch transaction updating

### Form image transactions

A segment can be updated using a batch transaction in the format of the form image. Form image transactions are processed by CBSVB in a batch mode. This technique is used when:

- Converting to The Solution Series
- Interfacing to The Solution Series
- An audit trail is required

### Form image layouts

The BATCHL program provides layouts for form images. This utility produces a report (FILE03) detailing the contents and layout of each form image transaction that you may want to process. The example shown is for the 43-SCR that creates a ZQ Segment when processed.

### Applying the transaction file

The technique of updating via form image transactions requires two steps:

1. Develop and execute the program that will write the form image transactions to a file.
2. Execute CBSVB to read the output from Step 1 as FILE04 input.

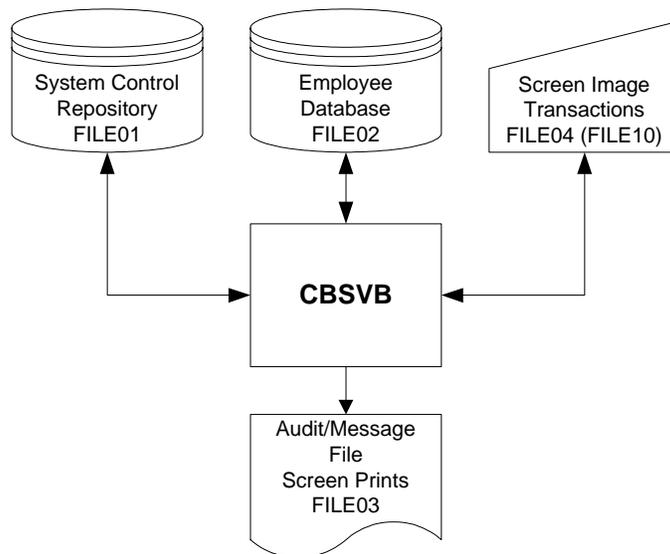
Your Step 1 program may be written using any language. If using CSL a Report or Query (Batch) program can be written to extract the form image transactions. Recall that both standard jobs include FILE10 as an output file.

# BATCHL Transaction Updating

## Step 1: Example Program - Query

<b>Read Record:</b>	READ-EMPLOYEE.
<b>Selection:</b>	P100-SELECT. @SELECTION LOGIC HERE.
<b>Date Calculation</b>	P200-CALCULATE-DATES. @DATE CALCULATIONS GO HERE.
<b>Build Form Images</b>	P300-CREATE-SCREEN-IMAGES. SPACE-EXTRACT-RECORD. PRINT 'P 43-SCRT00010 ' CONTROL-1-2 '43-SCR ' EMPLOYEE-NUMBER SPACE-OVER :15 WORK-DATE 'SA' SPACE-OVER :10 '*' ' '.
<b>Write File:</b>	WRITE-FILE10.
<b>Build Form Image:</b>	PRINT 'P 43-SCRT00011 ' SPACE-OVER :10 SAVE-DATE SPACE-OVER :30 HOLD-DATE SPACE-OVER :08 SPACE-OVER:10.
<b>Write File:</b>	WRITE-FILE10.

## Step 2: Applying the Form Image Transactions




---

## NOTES

---

---

## Batch transaction updating, continued

### Example program

The contents of the FILE10 records follow the BATCHL guidelines for the 43-SCR layout and conform to the FILE04 control record fields, for example:

- Program Literal = P 43-SCR (comments)
- Task Number = T00010 (comments)
- Key Field = EMPLOYEE-NUMBER
- NEXT-REVIEW-DATE = WORK-DATE (calculated)

### SCREEN work area

The SCREEN work area is used to build the form image transaction. Recall these verbs that affect the SCREEN work area:

- SPACE-EXTRACT-RECORD
- PRINT, SPACE-OVER :99, OUTPUT

### WRITE-FILE10

Recall that this verb writes an 80-character record to FILE10. The record is written from the SCREEN work area starting at :1601. After the write, the SCREEN address is set back to :1601.

### Recommended update technique

This method of updating the file has several advantages:

- The form program edits the transactions ‘as if’ you had entered the form manually.
- The audit/message file (FILE03) displays all errors.
- An audit record is created only when The Solution Series is running in a Production mode (PRODUCTION-VERSION field = Y).

## Section Summary

- **Inserting a segment**
- **Deleting a segment**

---

### NOTES

---

## Section summary

In this section, you learned several techniques for updating the Employee Database. Complete the following questions to summarize the section:

### Inserting a segment

- The verb used to initialize the Pointer 8 work area is \_\_\_\_\_.
- Segment data to be inserted is laid out in \_\_\_\_\_ order.
- Before executing an INSERT- verb, position the pointer with a \_\_\_\_\_ or \_\_\_\_\_ verb.
- In a report program, the system field \_\_\_\_\_ is used to verify that INSERT- was successful.
- During a query update the use of the \_\_\_\_\_ insures that the record is locked.
- An employee record is unlocked by the \_\_\_\_\_ verb.

### Deleting a segment

- The verb \_\_\_\_\_ is used to delete employee level tax segments.
- Like the INSERT- verbs, when a DELETE- verb is used in a form entry program, it must follow the \_\_\_\_\_ logic and the \_\_\_\_\_ logic.

### Updating with BATCHL transactions

- The \_\_\_\_\_ program produces a report detailing a form image transaction layout.
- When a form image requires more than one 80-character record, a \_\_\_\_\_ must be present in position 75 indicating continuation.
- Form image transaction updates are read in as input to \_\_\_\_\_

## Section 5 Exercise

### Segment Layout for the ZQ Segment:

SEGMENT LAYOUT REPORT L SEGMENT					
POSITIONS		FIELD NAME	LENGTH	PIC	COM
1	1	SEGMENT-TYPE	001	X(001)	L
2	3	SEGMENT-CODE	002	X(002)	ZQ
4	9	NEXT-REVIEW-DATE	006	9(006)	K
10	11	NEXT-REVIEW-TYPE	002	X(002)	HR34
12	41	REVIEWER-NAME	030	X(030)	
37	41	FILLER	005	X(005)	
42	51	DISTRIBUTION-DATA	010	X(010)	
52	57	DISTRIBUTION-DATE	006	9(006)	
58	63	DATE-RETURN-EXPECTED	006	9(006)	
64	69	ACTUAL-RETURN-DATE	006	9(006)	

---

## NOTES

---

## Section 5 exercise

### Purpose

The purpose of this exercise is to practice inserting segments. You will update the Employee Database with a Next Review Segment.

### Directions

Write and execute a query program to insert a Scheduled Salary Review (43–SCR, ZQ Segment) based on the current Performance Appraisal Results (49–SCR, ZS segment) and the following data.

### Selection

Segments should not be inserted for employees who already have a current review segment or whose status (RESULTING–EMP–STATUS) is: Retired (9), Deceased (5), or Terminated (1).

Field	Data
NEXT–REVIEW–DATE (CYYMDD)	One year anniversary of last review (RATING–DATE + 1 year = NEXT–REVIEW–DATE)
NEXT–REVIEW–TYPE	SA– Salary Review Annual
DISTRIBUTION–DATE (YYMMDD)	One month prior to the NEXT–REVIEW–DATE
DATE–RETURN–EXPECTED (CYYMDD)	One week (7 days) from the NEXT–REVIEW–DATE

Execute the QUERY program using the Query Key of '00'.

---

**NOTES**

---

# Section 6: Special Report Options

---

## Table of Contents

Introduction..... 145  
Special print option overview ..... 147  
Percentage of total report ..... 149  
Multiple employee format—3 across labels ..... 153  
Report schedule parameters ..... 157  
System Control Repository reporting ..... 163  
Section 6 exercise..... 165

## Objectives

- **Recall the special print program features**
- **Develop techniques to handle special print procedures**
- **Use report schedule parameters for advanced techniques**

---

NOTES

---

## Introduction

### **Purpose**

In this section advanced reporting techniques for special processing and printing are covered.

### **Objectives**

Upon completion of this section you will be able to:

- Recall the special print program features
- Develop techniques to handle special print procedures
- Use report schedule parameters for advanced techniques

## Special Print Option Paragraphs

Paragraph	RTPRNT Process Timing
100	The report is starting
120	New Organization Number is starting
140	The header records are ready to print
160	The header records have been printed
180	Detail data with matching sort keys is ready
200	Total data is ready
220	A detail line is ready to print
240	A total line is ready to print
260	The report is done

---

### NOTES

---

## Special print option overview

### Special print option paragraphs

As you will recall, the special print option paragraphs are performed at specific times during the report print phase. Each paragraph is designed to allow access to specific parts of the report output before and after the print line has been formatted.

General rules associated with the special print options program include:

- All paragraph labels **MUST** be present, even if your special print option program does not use them.
- Each paragraph must contain an EXIT that directs control back to RTPRNT.
- Additional paragraphs can be added to the program in the range of 100–999 (with the exception of the ones already used). Each new paragraph must contain an EXIT, so control can be properly returned to the performing paragraph.

### 5M–RP program

The 5M–RP program is a delivered model that you may use as a prototype whenever you need to create a special print options program. The 5M–RP program contains all the required paragraphs, as well as comments providing direction for each paragraph's use.

# Percentage of Total Report

## Report Extract:

```
DEFINE-REPORT TOTALS-ONLY.  
@HEADING Statements  
P100-START-SELECT.  
  SET-EMP-PTRS-TO-1ST. SET-CO-PTRS-TO-1ST.  
  FIND-ACTIVITY.  
  IF NOT FOUND RETURN.  
  IF RESULTING-EMP-STATUS NOT EQUAL '0' RETURN.  
P200-INITIALIZATION.  
  MOVE :0 TO PERM-01-V0.  
  MOVE :0 TO PERM-02-V0.  
  IF SEX-CODE EQUALS 'M'  
    MOVE :1 TO PERM-01-V0      @Male Count  
  ELSE  
    MOVE :1 TO PERM-02-V0.    @Female Count  
  MOVE :1 TO PERM-03-V0.    @Male/Female Count  
  MOVE :0 TO PERM-04-V2.    @Init Male Ratio  
  MOVE :0 TO PERM-05-V2.    @Init Female Ratio  
P300-SORT.  
SPACE-EXTRACT-RECORD.  
OUTPUT '1XAVG' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
DOUBLE-SPACE-BEFORE CONTROL-1-2 DOUBLE-SPACE-BEFORE.  
FIND CONTROL-3.  
IF NOT FOUND SPACE-OVER :04  
ELSE OUTPUT CONTROL-3-CODE BREAK-DEFAULT  
SORT-LENGTH-25 EMPLOYEE-NUMBER.  
P400-REPORT.  
OUTPUT '0' PERM-01-V0 PERM-02-V0 PERM-03-V0  
PERM-04-V2 PERM-05-V2.  
WRITE-EXTRACT.  
RETURN.
```

## Report Print Positions:

SEQ NBR	FIELD NAME	-----PRINT----- POS.	LINE	TOTAL
05	PERM-01-V0	030	01	Y
10	PERM-02-V0	050	01	Y
15	PERM-03-V0	070	01	Y
20	PERM-04-V2	090	01	Y
25	PERM-05-V2	110	01	Y

---

## NOTES

---

## Percentage of total report

### Percent of total

Any report that requires a percentage of total must first have the total calculated before it can calculate a percentage of total. In The Solution Series the percentage of total calculation is performed in the special print options program, but also requires some special set-up in the report extract program and report print position records.

### Example

The following example is for a report that gives a percentage of males and females in each Organization Level 3. A grand total count and percentage is also given.

### Extract Program

The extract program is used to provide a count for each male, female and a total of all employees. Paragraph P200 updates the PERM- fields with each employee's count and initialize two additional PERM- field used in the calculation of the percentage of total. It is necessary to define all fields used in the calculation here before coding the special print options program.

### Print Position Record (RTEDIT)

The Print Position Record defines the fields in the extract. Remember it is necessary to define all fields used in the calculation before coding the special print options program.

### Special Print Options Program

This program is used to calculate the percentage of total. The paragraphs used to produce the labels are detailed on the following pages.

# Percentage of Total Report

## Special Print Options Program:

```
P200-P. @ TOTAL DATA IS READY
  CALCULATE 0:PERM-01-V0 / 0:PERM-03-V0 = 0:PERM-04-V2.
  CALCULATE 0:PERM-02-V0 / 0:PERM-03-V0 = 0:PERM-05-V2.
EXIT.
```

## Percentage of Total Report:

CORPORATION 99		ACME MANUFACTURING		Ratio of Men to Women		REPT PERIOD		FILE VERSION 00		PAGE 1	
DIVISION 9999		PRODUCTION CTL 1-2				XAVG PERIOD		TIME 15:01:10		DATE 11-07-2002	
Control		Nbr of Males	Nbr of Females	Total Active		Male Ratio		Female Ratio			
3 Code				Employee							
*CONTROL-3-CODE		33	17	50		.66		.34			
*CONTROL-3-CODE	01	13	5	18		.72		.28			
*CONTROL-3-CODE	02	5	2	7		.71		.29			
*CONTROL-3-CODE	03		2	2		.00		1.00			
*CONTROL-1-2	999999	51	26	77		.66		.34			
*REPORT CODE	XAVG	51	26	77		.66		.34			

---

## NOTES

---

## Percentage of total report, continued

### Example

#### Special Print Program paragraphs

Paragraph P200 is used to calculate the percentage of total.

#### P200

At the time P200 is performed, the PERM values contain the total counts for Male, Female, and all employees. Now we can calculate the percentage of total by dividing the individual counts by the total count. The PERM counter is referenced using the RECORD-TYPE:FIELD-NAME technique.

- Male count (PERM-01-V0) / employee count (PERM-03-V0) = % of Males (PERM-04-V2).
- Female count (PERM-02-V0) / employee count (PERM-03-V0) = % of Females (PERM-05-V2).

# Multiple Employee Format—3 Across Labels

## Report Extract:

```
DEFINE-REPORT NO-HEADINGS.  
P100-INITIALIZE. @INITIALIZE COUNTERS HERE IF NECESSARY.  
SPACE-EXTRACT-RECORD.  
SET-EMP-PTRS-TO-1ST.  
P200-SELECT. @RECORD SELECTION LOGIC GOES HERE.  
P300-SORT.  
PRINT '1XLAB' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL  
SORT-LENGTH-51  
CONTROL-1-2 EMPLOYEE-NAME EMPLOYEE-NUMBER.  
P400-OUTPUT.  
MOVE EMPLOYEE-NAME TO LAST-FIRST. CALL 'FMLED'T'.  
MOVE PRINT-FIELD TO FIRST-LAST.  
PRINT '0' FIRST-LAST ADDRESS CITY/STATE  
ZIP-CODE.  
WRITE-EXTRACT.
```

## Report Print Positions:

SEQ	FIELD	----PRINT----		
NBR	NAME	POS.	LINE	TOTAL
05	EMPLOYEE-NAME	080	01	N
10	ADDRESS	080	02	N
15	CITY/STATE	080	03	N
20	ZIP-CODE	112	03	N

---

## NOTES

---

## Multiple employee format—3 across labels

### Multiple employee format

Any report requiring multiple employee records to print on one line may be accomplished by holding each employees data until all employees have been accessed.

### Example

The following example prints three across mailing labels.

### Extract Program

Data for labels is extracted in the REPORT step. As usual, extract records are written separately for each employee. Notice that the employee name will be converted to First, Last order prior to being extracted.

### Print Position Record (RTEDIT)

The print position record is used to provide a general layout for one label. Each field is assigned to a different print line, this will be used in the special print options program to determine the timing of when to hold and print the labels.

### Special Print Options Program

This program is used to override detail processing. The paragraphs used to produce the labels are detailed on the following pages.

# Multiple Employee Format—3 Across Labels

## Initialization

```
P100-P. @ REPORT IS STARTING
MOVE :1 TO PERM-01-V0
MOVE SPACES TO W8-60-700. MOVE SPACES TO W8-38-760.
MOVE SPACES TO W8-60-830. MOVE SPACES TO W8-38-890.
EXIT.
```

```
P220-P. @ A DETAIL LINE IS READY TO PRINT - POINTER 8
```

## Hold the 1st Employees Address Information

```
IF PERM-01-V0 EQUALS :1
  IF W8-02-158 EQUALS '01'
    MOVE W8-30-079 TO W8-30-700 @ NAME
    MOVE '99' TO LINE-ADVANCE EXIT
  ELSE IF W8-02-158 EQUALS '02'
    MOVE W8-30-079 TO W8-30-730 @ ADDRESS
    MOVE '99' TO LINE-ADVANCE EXIT
  ELSE IF W8-02-158 EQUALS '03'
    MOVE W8-38-079 TO W8-38-760 @ CITY/STATE ZIP
    CALCULATE PERM-01-V0 + :1 = PERM-01-V0
    MOVE '99' TO LINE-ADVANCE EXIT.
```

## Hold the 2nd Employees Address Information

```
IF PERM-01-V0 EQUALS :2
  IF W8-02-158 EQUALS '01'
    MOVE W8-30-079 TO W8-30-830 @ NAME
    MOVE '99' TO LINE-ADVANCE EXIT
  IF W8-02-158 EQUALS '02'
    MOVE W8-30-079 TO W8-30-860 @ ADDRESS
    MOVE '99' TO LINE-ADVANCE EXIT
  IF W8-02-158 EQUALS '03'
    MOVE W8-38-079 TO W8-38-890 @ CITY/STATE ZIP
    CALCULATE PERM-01-V0 + :1 = PERM-01-V0
    MOVE '99' TO LINE-ADVANCE EXIT.
```

## Process the Employee Address Information

```
IF PERM-01-V0 EQUALS :3
  IF W8-02-158 EQUALS '01'
    MOVE W8-30-700 TO W8-30-000 @ NAME #1
    MOVE W8-30-830 TO W8-30-039 @ NAME #2
    MOVE '02' TO LINE-ADVANCE @ DOUBLE SPACE
  IF W8-02-158 EQUALS '02'
    MOVE W8-30-730 TO W8-30-000 @ ADDRESS #1
    MOVE W8-30-860 TO W8-30-039 @ ADDRESS #2
    MOVE '01' TO LINE-ADVANCE @ SINGLE SPACE
  IF W8-02-158 EQUALS '03'
    MOVE W8-38-760 TO W8-38-000 @ CITY/STATE ZIP #1
    MOVE W8-38-890 TO W8-38-039 @ CITY/STATE ZIP #2
    MOVE '01' TO LINE-ADVANCE @ SINGLE SPACE
    MOVE :1 TO PERM-01-V0
    MOVE SPACES TO W8-60-700 MOVE SPACES TO W8-38-760
    MOVE SPACES TO W8-60-830 MOVE SPACES TO W8-38-890.
EXIT.
```

## Process any Remaining Employees

```
P260-P. @ THE REPORT IS DONE.
IF SPACES EQUAL W8-60-700 EXIT.
MOVE W8-30-700 TO W8-30-000. @ NAME #1
MOVE W8-30-830 TO W8-30-039. @ NAME #2
MOVE '02' TO LINE-ADVANCE. WRITE FILE03.
MOVE W8-30-730 TO W8-30-000. @ ADDRESS #1
MOVE W8-30-860 TO W8-30-039. @ ADDRESS #2
MOVE '01' TO LINE-ADVANCE. WRITE FILE03.
MOVE W8-38-760 TO W8-38-000. @ CITY/STATE ZIP #1
MOVE W8-38-890 TO W8-38-039. @ CITY/STATE ZIP #2
MOVE '01' TO LINE-ADVANCE. WRITE FILE03.
EXIT.
```

## 3-Across Label Examples:

RICHARD ADAMS 4272 NORTH AVE CHICAGO, IL	60635	GEOFFERY ALSON 840 MARGRET STREET DES PLAINES, IL	60016	KARI ANDERSEN 692 S. 9TH AVENUE LA GRANGE, IL	60240
--	-------	---	-------	---	-------

## NOTES

---

## Multiple employee format—3 across labels, continued

### Example

#### Special Print Program paragraphs

Paragraphs P100, P220, and P260 are used to process the three across labels.

#### P100

The PERM counter and hold area in pointer 8 used to process each group of three employees is initialized.

#### P200

Print control is based on groups of three employees:

- W8-02-158 contains the Line Number value.
- If the count equals one, each print line for the first employee is moved to a hold area and the print line is suppressed.
- If the count equals two, each print line for the second employee is moved to a hold area and the print line is suppressed.
- If the count equals three,
  - The first employees hold area is restored to positions 1-39
  - The second employees hold area is restored to positions 30-79
  - The third employee prints in positions 80-110 as defined in the print position record
  - The PERM counter and hold area are initialized for the next group

#### P260

If less than three employees were processed in a group, one or two employee names have not been printed.

- If hold area W8-30-700 is spaces, no detail exists.
- The hold areas are moved to the print areas and printed.

# Report Parameter Validation/Error Message

**Parameter:** IF FIRST-TIME-IND EQUALS 'F'  
**Validation:** IF SPACES EQUALS W6-10-036  
**Error Message:** INITIAL-PRINT-LINE  
MOVE '\*\*\*REJECT: X10RPT - Missing Report Schedule Parameters  
TO W8-53-000  
WRITE FILE03  
**Bypass Employees:** SET WORK TO :7  
MOVE 'X' TO WORK RETURN.

## FILE03 output from REPORT:

```
CSSS 0005 ( (999999(REPORT( (CLASS ( ) 12:10:01 10-15 USER  
***REJECT: X10RPT - Missing Report Schedule Parameters  
***REJECT: X10RPT - Missing Report Schedule Parameters  
***REJECT: X10RPT - Missing Report Schedule Parameters  
----Complete----
```

---

## NOTES

---

## Report schedule parameters

When using report parameters, it is important to either provide defaults when no parameters are entered or to stop the execution of the report and issue an error message.

### Example

The example above validates the report parameter, prints an error message, and stops the execution of the report.

### Parameter Validation

The logic for the parameter validation should be performed the first time the report program is executed. The FIRST-TIME-INDICATOR field will contain an 'F' when the report is being executed the first time. W6-36-036 contains the report parameters.

### Error Message

Error messages are written to FILE03 using the direct file processing technique explained in Section 2.

### Bypass Employees

To avoid the processing of any additional employees through the report for the current Control 1-2, move an 'X' to position 7 of the WORK pointer.

*Note: The Bypass Employees Report logic will only skip Employee Records, the Company Record is still read for each Organization Number in the Organization Number Schedule. Therefore an error message will be written for each Organization Number in the Organization Number Schedule since the FIRST-TIME-IND is set to 'F' for each Organization Number. This report should be run alone in the Report Schedule since the Bypass Employees logic impacts the reading of records regardless of the reports in the Schedule.*

# Changing the Sort Sequence using Report Parameters

```

Standard Sort:   PRINT '1X01R' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL
                   SORT-LENGTH-51 CONTROL-1-2.

Variable Sort:  IF W6-02-036PARMS EQUALS 'NM'
                   PRINT EMPLOYEE-NAME EMPLOYEE-NUMBER
                   ELSE IF W6-02-036 EQUALS 'SS'
                   PRINT SOCIAL-SECURITY-NBR EMPLOYEE-NUMBER SPACE-OVER :18
                   ELSE IF W6-02-036PARMS EQUALS 'C4'
                   PRINT CTRL-FOUR EMPLOYEE-NUMBER SPACE-OVER :26
                   ELSE PRINT EMPLOYEE-NUMBER SPACE-OVER: 30.
    
```

Sort Sequence	Report Parameter	Sort Key Length
EMPLOYEE-NAME and EMPLOYEE-NUMBER	NM	Standard Sort = 11 Name = 30 Number = <u>10</u> 51
SOCIAL-SECURITY-NBR and EMPLOYEE-NUMBER	SS	Standard Sort = 11 Social Security Number = 12 Number = 10 Spaces = <u>18</u> 51
CTRL-FOUR and EMPLOYEE-NUMBER	C4	Standard Sort = 11 Control = 4 Number = 10 Spaces = <u>26</u> 51
EMPLOYEE-NUMBER	EN	Standard Sort = 11 Number = 10 Spaces = <u>30</u> 51

---

## NOTES

---

## Report schedule parameters, continued

### Variable Sorts

A report may be requested which has several different sorts. However, a sort structure is internal to the report's logic and cannot be changed without re-compiling the program. An efficient way to solve this dilemma is to build the sort variability into the report.

### Example

The example above uses a report parameter to determine the sort sequence for a given run.

### Standard Sort

The standard sort defines the maximum sort length using the SORT-LENGTH-nn verb.

### Variable Sort

Each sort sequence is assigned a two character input code- NM, SS, C4, EN respectively. When a sort combination falls short of 51 characters, the sort key for the extract record is padded with blanks. For instance, the C4 sort option has only 25 characters, so 26 spaces are needed. The sort length must remain static.

*Note: This technique works only when the Variable Sort fields do not have Sort Options such as DOUBLE-SPACE-BEFORE, NO-PRINT-SUBTOTAL, and so forth. Recall that these options cause the report to reserve accumulators and line control parameters that are specific to the field they immediately follow.*

# Printing Report Schedule Parameters

## Extract Program:

```
DEFINE-REPORT ALLOCATE-12.  
:  
:  
Period IF W6-01-035 EQUALS 'F'  
Record PRINT '1X1-R' CONTROL-1-2 SPACE-OVER:10 PRINT '.2063'  
SCREEN-KEY '..' WRITE-EXTRACT.
```

## Special Print Options Program:

```
Initialization: P100-P. @REPORT IS STARTING  
MOVE 'Y' TO W6-01-100.  
EXIT.  
:  
:  
P140-P. @HEADER RECORDS READY TO PRINT  
IF W6-01-100 NOT EQUAL 'Y' EXIT.  
Read Report MOVE '14' TO KEY01-SIZE.  
Parameter Record MOVE 'PE' TO W7-02-010.  
SET SCREEN TO :1435.@ HDR 2 POS 63  
MOVE SCREEN TO W7-06-012.  
MOVE 'X1-RPT' TO W7-06-018.  
READ-UNIQUE FILE01.  
Print Parameters IF STAT-KEY EQUAL '00'  
MOVE W8-36-014 TO W8-36-800  
INITIAL-PRINT-LINE  
MOVE 'Report Schedule Parameters: ' TO W8-28-033  
MOVE W8-36-800 TO W8-36-063  
WRITE FILE03  
INITIAL-PRINT-LINE  
MOVE '00' TO LINE-ADVANCE  
WRITE FILE03.  
SET SCREEN TO :1435.  
MOVE ' ' TO SCREEN.  
MOVE SPACE TO W6-01-100.  
EXIT.
```

## Report Parameters Example:

Report Schedule Parameters: 200L23200A01
--

---

## NOTES

---

## Report schedule parameters, continued

### Print report parameters

Report schedule parameters can be printed on the report by passing the Report Schedule Name to the Special Print program via the extract record.

### Example

#### Extract Program

A period record is used to pass the Report Schedule Name (SCREEN-KEY) to the extract record. A dummy Organization Level 2 value is used on this extract record.

#### Special Print Program

Paragraphs P100 and P140 are used to process the report parameters.

#### P100

The Pointer 6 field is initialized in P100. This field is used to indicate the first pass through the Special Print Options logic so the report parameters can be printed.

#### P140

The report header that contains the Report Schedule Name is available in this paragraph.

- When the Pointer 6 field is 'Y' the paragraph is executed.
- The Report Schedule (PE) record is read from FILE01 using the Schedule Name in Header 2, print position 063, and report name.
- The 36 character Report Schedule Parameters are printed on the report.
- The SCREEN area is then reset to the area where the Report Schedule name appears, and is initialized to blanks.

# System Control Repository Reports

## Extract Program:

```
DEFINE-REPORT NO-PE-DATES ALLOCATE-20.
@HEADER Statements
:      :
```

## Bypass Employees:

```
P100-START.
IF W6-06-047 EQUAL '      '
  MOVE CONTROL-1-2 TO W6-06-047
  MOVE 'Y' TO W6-01-035
  SET WORK TO :7 MOVE 'X' TO WORK
ELSE
  IF CONTROL-1-2 NOT EQUAL W6-06-047
    SET WORK TO :7 MOVE 'X' TO WORK RETURN.
```

## Read the 1st Record:

```
INITIAL-TABLE-AREA.
MOVE '02' TO KEY01-SIZE.
MOVE 'X1' TO W7-02-010. @KEY01-AREA
READ-UNIQUE FILE01.
IF STAT-KEY NOT EQUAL '00' RETURN.
```

## Build the Extract Record:

```
P200-SORT.
MOVE-TABLE-RECORD.
SPACE-EXTRACT-RECORD.
OUTPUT '1XSVR' FORMS/REPORT-CODE
NO-PRINT-GRAND-TOTAL
  SORT-LENGTH-25 CONTROL-1-2 XSUPV-CTRL-NBR
  XSUPV-LOCATION XSUPV-EFFECTIVE.
P300-REPORT.
PRINT '0' XSUPV-CTRL-NBR XSUPV-LOCATION
  XSUPV-EFFECTIVE XSUPV-NAME-CODE XSUPV-NAME
  XSUPV-WORK-PHONE XSUPV-WORK-EXT SUPV-HOME-PHONE
  XSUPV-MAIL-DIST.
WRITE-EXTRACT.
```

## Read Subsequent Records:

```
P400-READ-NEXT.
UNLOCK FILE01. MOVE TABLE-C01-40 TO W7-24-010.
MOVE '24' TO W7-00-008.
READ-UNIQUE FILE01. READ FILE01.
IF STAT-KEY EQUAL TO '00' AND W8-02-000 EQUALS 'X1'
  GO TO P200-SORT.
RETURN.
```

## Supervisor Table Report Example:

Ctrl Nbr	Location Code	Name	Effective Date	Supervisor Code	Supervisor Name	Work Phone	Work Ext.	Home Phone	Mail Stop
9999	01	Midwest	01-01-1980	JOHN	John Johnson	3124541865	0045	3125557898	02A-331122
9999	3030	Region 3030	01-01-1990	BILL	William Haze	3124541865	0045	3125557898	02A-33
9999	3333	Region 3333	01-01-1992	JACK	Jack Smith	3124541865	0045	3125557898	02A 3311

## NOTES

## System Control Repository reporting

### FILE01 reports

It is possible to use the report process to produce reports for System Control Repository records. This technique requires the report extract program to:

- Stop the automatic read of FILE02 records performed by the REPORT process.
- Include direct file processing to read the System Control Repository records.

### Example

The following example prints a report for the supervisor table example created previously in Section 4.

### Bypass employees

To stop the processing of employee records for the current Organization Number, an 'X' is moved to position 7 of WORK. Since this logic skips to the next Company in the Report Schedule, the Company Record is still read for each Organization Number in the Organization Number Schedule. Therefore, the Allocate area is being used to store the 1st Organization Number, and then tested to see if the report logic should be executed. The report extract logic is only executed for the 1st Organization Number.

### Read the 1st Record

FILE01 is read using the direct file processing techniques discussed in Section 2.

### Build the Extract Record

Once the table record is move to pointer 40 (MOVE-TABLE-RECORD), the extract record is created.

### Read Subsequent Records

Subsequent reads to FILE01 must be randomly read (READ-UNIQUE) for the current table record, and then sequentially read (READ) for the next table record. TABLE-C01-40 contains the current table key.

# Section 6 Exercise

---

**NOTES**

---

## Section 6 exercise

### Directions

Take 20 minutes to complete the following program:

1. Write a report to display the Pay Period Days Table. Refer to Section 4, exercise 1 for the specific field names you used to defining the table layout.

Record Identifier*	X
Table Identifier*	Two
Table Pay Frequency*	Alphanumeric, one position, Option List=PP29
Table Period End Date*	Century Date, six positions
Table Workdays by Period	Numeric, three positions
Table Holidays by Period	Numeric, three positions

---

**NOTES**

---

## Appendix A: Exercise Answers

---

### Table of Contents

Section 2 exercise 1.....	168
Section 2 exercise 2.....	169
Section 3 exercise.....	171
Section 4 exercise 1.....	174
Section 4 exercise 2.....	178
Section 5 exercise.....	187
Section 6 exercise.....	189

## Section 2 exercise 1

1. Read the System Control Repository (FILE01) Option List Records with a key of 'C SC04'.
2. Display the Description and Value of each option in the Option List.

```
SECURITY ' ' . @ Section 2 Exercise 1 XX
@LAST MODIFIED ON: 11-15-94 BY: XXXX AUTHOR: XXXX
P100-START.
MOVE 'C SC04' TO W7-06-010. @KEY01-AREA
MOVE '06' TO KEY01-SIZE.
READ-UNIQUE FILE01.
IF STAT-KEY NOT EQUAL '00'
PRINT-MESSAGE 'SC002' RETURN.
P200-LOOP.
MOVE W8-80-000 TO W8-80-800.
INITIAL-PRINT-LINE.
MOVE W8-14-807 TO W8-14-000. @Code Set Value
MOVE W8-20-823 TO W8-20-015. @Code Set Description
PRINT W8-79-000. NEXT-LINE.
READ FILE01.
IF STAT-KEY NOT EQUAL '00' OR
W8-06-000 NOT EQUAL TO 'C SC04'
PRINT-MESSAGE 'SC002' RETURN.
GO TO P200-LOOP.
```

## Section 2 exercise 2

1. Create a program to add the following report to a Report Schedule named 'CLASS': 1A-RPT, 1G-RPT, and 1R-RPT. For every write operation, check the status key and print a message (one per line) that contains 'STAT KEY: ' and the STAT-KEY value.

```
00000 SECURITY ' '. @ Section 2 Exercise 2
00001 @LAST MODIFIED ON:          BY:          AUTHOR:  USER
00100 P100-CREATE-HEADER.
00200   INITIAL-80.
00300   MOVE 'PECLASS ' TO W8-08-000.
00400   MOVE 'HCLASS EXERCISE REPORT GROUP' TO W8-00-023.
00500   WRITE FILE01.              @ HEADER RECORD
00600   PRINT 'STAT KEY: ' STAT-KEY. NEXT-LINE.
00700 P200-ADD-REPORTS.
00800   MOVE '1A-RPT' TO W8-06-800.
00900   PERFORM P300-WRITE.
01000   MOVE '1G-RPT' TO W8-06-800.
01100   PERFORM P300-WRITE.
01200   MOVE '1R-RPT' TO W8-06-800.
01300   PERFORM P300-WRITE.
01400   RETURN.
01500 P300-WRITE.
01600   INITIAL-80.
01700   SET PRINT-FIELD TO FIRST.
01800   MOVE 'PECLASS ' TO PRINT-FIELD.
01900   MOVE W8-06-800 TO PRINT-FIELD.
02000   WRITE FILE01.
02100   PRINT 'STAT KEY: ' STAT-KEY.
02200   NEXT-LINE.
02300   EXIT.
```

## Section 2 exercise 2, continued

2. Create another program to delete the Report Schedule named 'CLASS' from the System Control Repository.

```
SECURITY ' '. @ Section 2 Exercise 1 XX
@LAST MODIFIED ON: 11-15-94 BY: XXXX AUTHOR: XXXX
P100-START.
MOVE '08' TO KEY01-SIZE.
MOVE 'PECLASS ' TO W7-08-010. @Report Schedule Record.
READ-UNIQUE FILE01.
IF STAT-KEY NOT EQUAL '00'
PRINT-MESSAGE 'SC056' @Purge not performed
UNLOCK FILE01 RETURN
ELSE
DELETE-GLOBAL FILE01.
RETURN.
```

### Section 3 exercise

1. Modify Option List PP29 by adding logic to calculate an annualized salary for the employees regular pay (HED 001). The annualization calculation is as follows:

$$\text{SALARY} * \text{Annualization Factor (PERM-02-V2)} = \text{Annual Salary (PERM-01-V2)}$$

<u>Frequency</u>	<u>Annualization Factor</u>
1 weekly	52.00
2 bi-weekly	26.00
3 semi-monthly	24.00
4 monthly	12.00

```
Code      Seq Codeset Calculation - English Language Source
          Pay Frequency
          M   PAY-FREQUENCY-CODE
          M01 MOVE :0 TO PERM-01-V2. MOVE :0 TO PERM-02-V2.
          M02 GO TO P299-EXIT.
          M03 P200-CALCULATE.
          M04 IF SALARY NOT EQUAL :0
          M05 CALCULATE SALARY * PERM-02-V2 = PERM-01-V2. EXIT.
          M06 P299-EXIT.
1         Weekly
1         M01 MOVE :52.00 TO PERM-02-V2.
1         M03 PERFORM P200-CALCULATE.
1         M05 CODE-SET-DONE.
2         Bi Weekly
2         M01 MOVE :26.00 TO PERM-02-V2.
2         M03 PERFORM P200-CALCULATE.
2         M05 CODE-SET-DONE.
3         Semi Monthly
3         M01 MOVE :24.00 TO PERM-02-V2.
3         M03 PERFORM P200-CALCULATE.
3         M05 CODE-SET-DONE.
4         Monthly
4         M01 MOVE :12.00 TO PERM-02-V2.
4         M03 PERFORM P200-CALCULATE.
4         M05 CODE-SET-DONE.
```



## Section 3 exercise, continued

Display Box

Field Name: PERM-01-U2

Label: Annualized Salary:

Exit Routines

Before: 400

After:

When Shown

Always

Character Mode

Graphical Mode

Label Location

No Label

Left, Left Justified

Left, Right Justified

Above, Left Justified

Above, Centered on 2 Lines

Section: 1

OK Delete Cancel

```

P100-START-SCREEN.
KEY-REQUIRED. NO-ZDELETE-ALLOWED.
UPDATE-EMPLOYEE. NEW-SCREEN-STYLE.
SCREEN-SECTION '0'.
IF SCREEN-CODE1 EQUALS 'E'
  MOVE ' ' TO SCREEN-CODE1
ELSE IF SCREEN-CODE1 EQUALS 'I'
  MOVE 'E' TO W7-01-252
  MOVE 'E' TO W7-01-298
  MOVE 'S' TO W7-01-301
RETURN.
P200-ENTRY-SCREEN.
SCREEN-SECTION '1'.

P300-VERIFY.
SET-FOR-MESSAGES.
IF ERRORS-EXIST RETURN.
IF RECORD-NOT-UPDATED GO TO P999-PROMPT.
IF HED-NUMBER NOT EQUAL '001' PRINT-REJECT 'PR007'
ELSE MOVE 'H' TO KEY-FIELD-4.
RETURN.
P400-CALC-CODE-SET.
  CALC-CODE-SET 'PP29 '.
EXIT.
P990-INQUIRY-SCREEN.
RETURN.
P999-PROMPT.
SCREEN-SECTION '9'.
MOVE 'Y' TO AUTO-KEY-SWITCH.

```

## Section 4 exercise 1

```

      1   1   2   2   3   3   4   4   5   5   6   6   7   7   8
0....5....0....5....0....5....0....5....0....5....0....5....0
X29999XCYYMDD          999999
    
```

	Positions	Definition
<b>Table Record Key</b>		
X	0	Record Identifier
2	1	Table Identifier
9999	2–5	Control Number
X	6	Pay Frequency Code
CYYMDD	7–13	Period End Date
(blank)	14–23	Unused key area
<b>Table Data</b>		
999	24–26	Period Workdays
999	27–29	Period Holidays

## Section 4 exercise 1, continued

Field Maintenance And Edit

Action: **A**  
Field Name: **X2-ND-CONTROL-NUM**

Field Location  
Pointer: **40**  
Storage Length: **004**  
Displacement: **002**

Field Options  
Propagate:   
Rounding:   
Header Switch: **405153**  
 RDBMS Field

Field Properties  
Data Type: **Numeric 0 Decimals**  
Field Type:   
Template:   
Lengths: Display:  Entry:   
Module:   
Structure:   
Seg/Table ID: **X2** Table Separator:   
Codeset:   
Edit Routine:

Field Maintenance And Edit

Action: **A**  
Field Name: **X2-ND-PAY-FREQ-CD**

Field Location  
Pointer: **40**  
Storage Length: **001**  
Displacement: **006**

Field Options  
Propagate:   
Rounding:   
Header Switch: **109254**  
 RDBMS Field

Field Properties  
Data Type: **Alphanumeric**  
Field Type:   
Template:   
Lengths: Display:  Entry:   
Module: **Payroll/HRMS**  
Structure:   
Seg/Table ID: **X2** Table Separator:   
Codeset: **PP29**  
Edit Routine:

## Section 4 exercise 1, continued

Field Maintenance And Edit

Action:   
Field Name: X2-ND-PAY-FREQ

Field Location

Pointer: 40  
Storage Length: 001  
Displacement: 006

Field Options

Propagate:   
Rounding:   
Header Switch: 100410  
 RDBMS Field

Field Properties

Data Type: Alphanumeric  
Field Type: Codeset Description  
Template:   
Lengths: Display: 20 Entry:   
Module: Payroll/HRMS  
Structure:   
Seg/Table ID: X2 Table Separator:   
Codeset: PP29  
Edit Routine:

Field Maintenance And Edit

Action:   
Field Name: X2-ND-PERIOD-ED-DT

Field Location

Pointer: 40  
Storage Length: 006  
Displacement: 007

Field Options

Propagate:   
Rounding:   
Header Switch: 112521  
 RDBMS Field

Field Properties

Data Type: Century/Complement D  
Field Type:   
Template: Date  
Lengths: Display: 10 Entry:   
Module: Payroll/HRMS  
Structure:   
Seg/Table ID: X2 Table Separator:   
Codeset:   
Edit Routine:

## Section 4 exercise 1, continued

Field Maintenance And Edit

Action:

Field Name:

Field Location

Pointer:

Storage Length:

Displacement:

Field Options

Propagate:

Rounding:

Header Switch:

RDBMS Field

Field Properties

Data Type:

Field Type:

Template:

Lengths: Display:  Entry:

Module:

Structure:

Seg/Table ID:  Table Separator:

Codeset:

Edit Routine:

Field Maintenance And Edit

Action:

Field Name:

Field Location

Pointer:

Storage Length:

Displacement:

Field Options

Propagate:

Rounding:

Header Switch:

RDBMS Field

Field Properties

Data Type:

Field Type:

Template:

Lengths: Display:  Entry:

Module:

Structure:

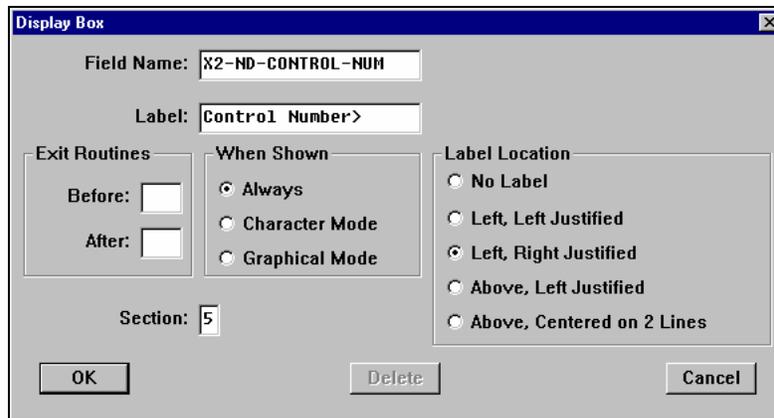
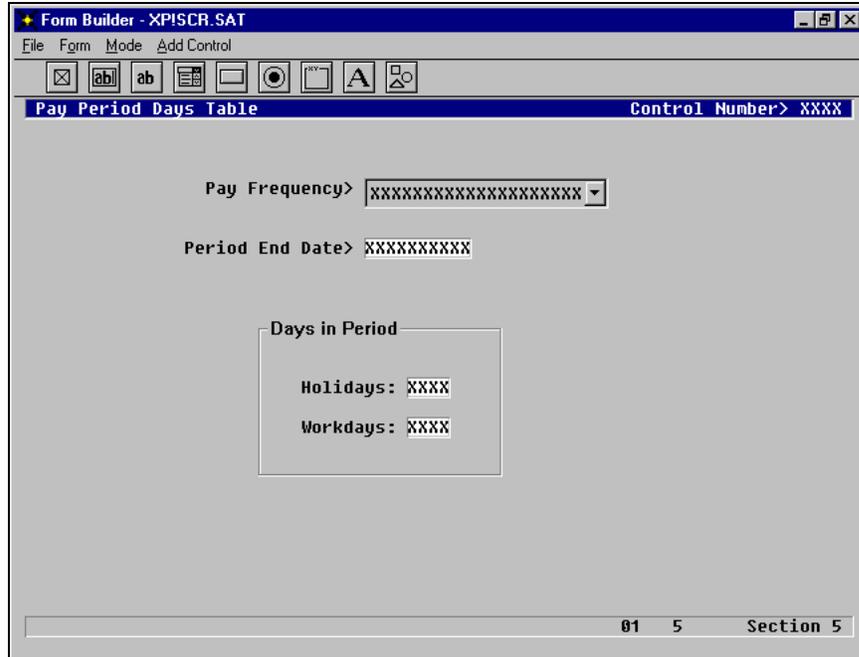
Seg/Table ID:  Table Separator:

Codeset:

Edit Routine:

## Section 4 exercise 2

1. Use Form Builder to create the form's appearance using the fields defined in Section 4 Exercise 1. Do not forget to Perform paragraph 850 after the date field.



## Section 4 exercise 2, continued

**List Box**

Field Name: X2-ND-PAY-FREQ-CD

Label: Pay Frequency

Section: 1

Exit Routines

Before:

After:

When Shown

Always

Character Mode

Graphical Mode

Big Option List

Label Location

No Label

Left, Left Justified

Left, Right Justified

Above, Left Justified

Above, Centered on 2 Lines

OK Delete Cancel

**Text Box**

Field Name: X2-ND-PERIOD-ED-DT

Label: Period End Date>

Section: 1

Exit Routines

Before:

After: 850

When Shown

Always

Character Mode

Graphical Mode

Spin Button?

Label Location

No Label

Left, Left Justified

Left, Right Justified

Above, Left Justified

Above, Centered on 2 Lines

Current Value

Don't Show

Show Inside Box

Show Below Box

OK Delete Cancel

## Section 4 exercise 2, continued

The screenshot shows a 'Text Box' dialog box with the following settings:

- Field Name: X2-ND-PERIOD-HL-DY
- Label: Holidays:
- Section: 1
- Spin Button?:
- When Shown:  Always,  Character Mode,  Graphical Mode
- Current Value:  Don't Show,  Show Inside Box,  Show Below Box
- Label Location:  No Label,  Left, Left Justified,  Left, Right Justified,  Above, Left Justified,  Above, Centered on 2 Lines
- Exit Routines: Before: , After:

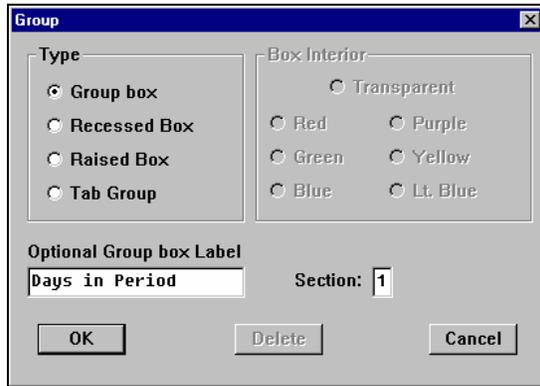
Buttons: OK, Delete, Cancel

The screenshot shows a 'Text Box' dialog box with the following settings:

- Field Name: X2-ND-PERIOD-WK-DY
- Label: Workdays:
- Section: 1
- Spin Button?:
- When Shown:  Always,  Character Mode,  Graphical Mode
- Current Value:  Don't Show,  Show Inside Box,  Show Below Box
- Label Location:  No Label,  Left, Left Justified,  Left, Right Justified,  Above, Left Justified,  Above, Centered on 2 Lines
- Exit Routines: Before: , After:

Buttons: OK, Delete, Cancel

## Section 4 exercise 2, continued



```

SECURITY ' '. @ Pay Period Days Table                                EXX
@LAST MODIFIED ON:                BY:    AUTHOR:
P100-START-SCREEN.
  ZDELETE-SECURITY.
  PERFORM P840-INITIAL-TABLE. RESET-TABLE-HELP.
  NEW-SCREEN-STYLE. MOVE 'N ' TO W8-02-750.
  SCREEN-SECTION '0'.
  IF TABLE-SELECTION SET-SELECTION-MODE.
  IF NOT INQUIRY-MODE AND NOT SELECTION-MODE
    GO TO P200-ENTRY-SCREEN.
P110-INQUIRY-SCREEN.
  IF TABLE-INQUIRY-UP
    OR TABLE-INQUIRY-DOWN
  IF TABLE-INQUIRY-PASS
    AND NOT SELECTION-MODE
    SET-TABLE-INQ-UPDN
    GO TO P200-ENTRY-SCREEN.
  SET P-E-PLACE TO :14.
  SET WORK TO :45.
  IF SELECTION-MODE
    IF W7-01-252 EQUAL ' ' GO TO P170-RESET-RETURN
    ELSE GO TO P130-SEARCH-WORK.
  IF W7-01-253 EQUAL 'F' MOVE ' ' TO W7-01-253
    MOVE ' ' TO SCREEN-CODE2 MOVE SPACES TO W7-24-254
  ELSE IF W6-02-200 EQUAL 'X2'
    AND TABLE-SCREEN-ENTRY
    MOVE W6-07-206 TO W7-07-254.
  MOVE SPACES TO W6-24-200.
  PERFORM P160-READ-CONTROL-NBR.
  MOVE 'E' TO W7-01-298.
  MOVE 'Q' TO W7-01-302.
  MOVE '00' TO SCREEN-LINE.
  IF SPACES EQUAL W7-11-254
    MOVE '06X2' TO W7-04-008
  ELSE MOVE '13X2' TO W7-04-008
    MOVE W7-07-254 TO W7-07-016
    MOVE SPACES TO W7-25-254.
  READ-UNIQUE FILE01.
  IF STAT-KEY NOT GREATER THAN '01'
    AND W8-06-000 EQUAL W7-06-010
    PERFORM P150-INQUIRY-SCREEN
  ELSE GO TO P170-RESET-RETURN.
P120-READ-NEXT.
  READ FILE01.

```

```
IF STAT-KEY NOT GREATER THAN '01'  
  AND W8-06-000 EQUAL W7-06-010  
  PERFORM P150-INQUIRY-SCREEN  
  IF SCREEN-LINE NOT EQUAL :17  
    GO TO P120-READ-NEXT.  
MOVE '*' TO WORK.  
SET-TABLE-SELECT. RETURN.  
P130-SEARCH-WORK.  
MOVE WORK TO W8-14-800.  
IF W8-01-811 EQUAL '@'  
  MOVE 'K' TO W7-01-252  
  PERFORM P140-MOVE-KEY  
  SET-TABLE-SCREEN  
ELSE IF WORK NOT EQUAL '*'  
  GO TO P130-SEARCH-WORK  
  ELSE MOVE 'I' TO W7-01-252  
    IF SPACES EQUAL W7-20-254  
      PERFORM P140-MOVE-KEY.  
RESET-TABLE-SELECT. RETURN.  
P140-MOVE-KEY.  
MOVE W8-01-800 TO W7-01-254.  
MOVE W8-02-801 TO SAVE-MONTH.  
MOVE W8-02-804 TO SAVE-DAY.  
MOVE W8-04-807 TO SAVE-DATE-CENTYEAR.  
MOVE CENTURY-SAVE-DATE TO WORK-DATE.  
MOVE WORK-DATE TO W7-06-255.  
EXIT.  
P150-INQUIRY-SCREEN.  
MOVE-TABLE-RECORD.  
MOVE TABLE-KEY TO W6-24-300.  
MOVE W7-40-176 TO WORK.  
SCREEN-SECTION '8'.  
CALCULATE SCREEN-LINE + :1 = SCREEN-LINE.  
MOVE W6-24-300 TO W7-24-010. MOVE '24' TO W7-02-008.  
READ-UNIQUE FILE01. EXIT.  
P160-READ-CONTROL-NBR.  
READ-CONTROL-NUMBER.  
MOVE JOB-CODES TO W7-04-012.  
EXIT.  
P170-RESET-RETURN.  
SET-INQUIRY-RETURN. RETURN.  
P200-ENTRY-SCREEN.  
IF GUI-IN-USE SET-SCREEN-TO-ENTRY  
  SET-SELECT-ONLY.  
CHECK-TABLE-INQUIRY.  
IF W7-01-253 EQUAL 'F'  
  PERFORM P160-READ-CONTROL-NBR  
  PERFORM P810-READ-UNIQUE  
  MOVE ' ' TO W7-01-253  
  MOVE 'S' TO W7-01-301  
  GO TO P525-MOVE-AND-RETURN.  
IF W7-01-302 EQUAL 'Q'  
  AND SPACES NOT EQUAL W6-24-200  
  AND W6-02-200 EQUAL 'X2'  
  AND SPACES EQUAL W7-24-254  
  AND TABLE-SCREEN-ENTRY  
  AND NOT EMPTY-TABLE-SCREEN  
  MOVE W6-07-206 TO W7-07-254.  
CLEAR-HOLD-TABLE-KEY.  
IF NOT TABLE-SCREEN-ENTRY  
  MOVE SPACES TO W7-25-254.
```

```
IF SPACES NOT EQUAL W7-07-254
  AND W7-07-264 NOT EQUAL 'ZDELETE'
  MOVE SPACES TO KEY01-AREA
  MOVE W7-07-254 TO W7-07-016
  MOVE SPACES TO W7-24-254
  PERFORM P810-READ-UNIQUE
ELSE
  IF W7-01-302 EQUAL 'Q'
    PERFORM P160-READ-CONTROL-NBR
    MOVE 'F' TO W7-01-253
    PERFORM P810-READ-UNIQUE
    MOVE ' ' TO W7-01-253
    IF HOLD-STAT-KEY EQUAL '00'
      MOVE TABLE-KEY TO W6-24-200
    ELSE SET-EMPTY-TABLE-SCR.
  SET-TABLE-SCREEN.
  SCREEN-SECTION '1'.
  SCREEN-SECTION '2'.
  PERFORM P570-SECTION-5.
  IF RECORD-NOT-UPDATED
    AND W6-01-100 EQUAL 'Q'
    MOVE 'Q' TO RECORD-UPDATED.
  IF W7-07-264 EQUAL 'ZDELETE'
    AND W8-01-750 EQUAL 'U'
    MOVE 'Y' TO RECORD-UPDATED.
P300-VERIFY.
NEXT-LINE.
PERFORM P560-SET-MESSAGES.
IF ERRORS-EXIST MOVE 'E' TO W7-01-301 RETURN.
IF W6-01-199 EQUAL 'S'
  MOVE 'S' TO W7-01-301
  GO TO P525-MOVE-AND-RETURN.
IF W8-01-751 NOT EQUAL SPACES GO TO P400-EDIT-ROUTINES.
IF RECORD-UPDATED EQUAL 'Q'
  IF W7-07-264 EQUAL 'ZDELETE'
    GO TO P400-EDIT-ROUTINES
  ELSE RETURN.
IF RECORD-NOT-UPDATED
  IF W7-01-278 NOT EQUAL SPACES CALL 'TBLMSG' RETURN
  ELSE IF W7-01-253 EQUAL 'U' OR 'B'
    AND W6-02-200 EQUAL 'X2'
    MOVE W6-24-200 TO KEY01-AREA
    PERFORM P810-READ-UNIQUE
    RETURN
  ELSE RETURN.
IF WHAT-IF-MODE
  GO TO P500-ERROR-CHECK.
P400-EDIT-ROUTINES.
IF W7-07-264 EQUAL 'ZDELETE'
  IF W8-01-750 EQUAL 'U' GO TO P500-ERROR-CHECK
  ELSE PRINT-REJECT 'SC526'.
IF W8-01-751 NOT EQUAL SPACES
  IF W8-01-751 EQUAL 'N' PRINT-REJECT 'HR007'
  GO TO P500-ERROR-CHECK
ELSE
  IF W8-01-751 EQUALS 'S' PRINT-REJECT 'HR012'
  GO TO P500-ERROR-CHECK.
P500-ERROR-CHECK.
IF ERRORS-EXIST
  OR WARNINGS-EXIST MOVE 'E' TO W7-01-301 RETURN.
IF WHAT-IF-MODE
```

```
RESET-RECORD-UPDATE
MOVE W6-24-200 TO TABLE-KEY
GO TO P525-MOVE-AND-RETURN.
IF W8-01-750 EQUAL 'U' GO TO P520-UPDATE.
P510-ADD-RECORD.
MOVE-TABLE-TO-PRINT.
WRITE FILE01.
IF STAT-KEY NOT EQUAL '00' PRINT-REJECT 'SC01W'
GO TO P500-ERROR-CHECK.
MOVE 'A' TO W7-01-278. GO TO P524-SET-RECORD-LOCKED.
P520-UPDATE-RECORD.
IF W7-07-264 EQUAL 'ZDELETE' GO TO P530-DELETE.
MOVE-TABLE-TO-PRINT. REWRITE FILE01.
IF STAT-KEY NOT EQUAL '00' PRINT-REJECT 'SC01T'
GO TO P500-ERROR-CHECK.
MOVE 'U' TO W7-01-278.
P524-SET-RECORD-LOCKED.
MOVE '40' TO RECORD-LOCKED.
P525-MOVE-AND-RETURN.
MOVE X2-ND-PAY-FREQ-CD TO W7-07-254. RETURN.
P530-DELETE.
MOVE TABLE-KEY TO W7-24-010.
MOVE '13' TO KEY01-SIZE.
DELETE-GLOBAL FILE01.
IF STAT-KEY EQUAL '50' OR '53'
PRINT-REJECT 'SC01G' GO TO P500-ERROR-CHECK.
MOVE 'Z' TO W7-01-278.
MOVE '40' TO RECORD-LOCKED.
MOVE SPACES TO W6-24-200.
MOVE SPACES TO W7-14-264.
RETURN.
P560-SET-MESSAGES.
SET-FOR-MESSAGES. EXIT.
P570-SECTION-5.
SET SCREEN TO SAVE.
SCREEN-SECTION '5'.
SET SCREEN TO SAVE.
EXIT.
P800-FIND-RECORD.
MOVE SPACES TO W7-22-012.
MOVE CONTROL-NUMBER TO W7-22-012.
P810-READ-UNIQUE.
IF W7-01-253 NOT EQUAL 'F'
PERFORM P830-FIND-AX-RECORD
MOVE JOB-CODES TO CONTROL-NUMBER
MOVE JOB-CODES TO W7-04-012
MOVE '13X2' TO W7-04-008
ELSE
MOVE '06X2' TO W7-04-008.
READ-UNIQUE FILE01.
MOVE STAT-KEY TO HOLD-STAT-KEY.
MOVE ' ' TO W6-01-199.
IF SPACES NOT EQUAL W6-24-200
IF W6-24-200 NOT EQUAL KEY01-AREA
AND NOT WHAT-IF-MODE
MOVE 'S' TO W6-01-199
ELSE NEXT SENTENCE
ELSE MOVE KEY01-AREA TO W6-24-200.
IF STAT-KEY EQUAL '00' MOVE 'U' TO W8-01-750
GO TO P820-MOVE-RECORD
ELSE IF RECORD-UPDATED EQUAL 'Q' MOVE 'A' TO W8-01-750
```

```
        MOVE 'X2' TO TABLE-C01-02
        ELSE IF W7-02-008 EQUAL '06' EXIT
            ELSE MOVE KEY01-AREA TO TABLE-KEY EXIT.
    IF W6-01-199 EQUAL 'S'
        AND W8-01-750 EQUAL 'A'
        MOVE ' ' TO W6-01-199
        IF W8-01-751 NOT EQUAL 'N' AND 'S'
            MOVE KEY01-AREA TO W6-24-200. EXIT.
P820-MOVE-RECORD.
    IF W7-01-253 EQUALS 'F'
        MOVE W8-24-000 TO W7-24-010.
    MOVE-TABLE-RECORD.
    PERFORM P860-SET-BUTTONS.
    EXIT.
P830-FIND-AX-RECORD.
    MOVE KEY01-AREA TO W8-24-110.
    MOVE W8-06-117 TO CROSS-REFERENCE-DATE.
    READ-TZAX-TABLE.
    IF STAT-KEY GREATER THAN '01' MOVE 'N' TO W8-01-751
        PERFORM P835-INITIAL-TABLE-KEY
    ELSE IF SPACES EQUAL JOB-CODES
        MOVE 'S' TO W8-01-751
        PERFORM P835-INITIAL-TABLE-KEY.
    MOVE W8-24-110 TO KEY01-AREA.
    EXIT.
P835-INITIAL-TABLE-KEY.
    MOVE SPACES TO W6-24-200.
    IF GUI-IN-USE SET-SELECT-ONLY.
P840-INITIAL-TABLE.
    INITIAL-TABLE-AREA.
    EXIT.
P850-CHECK-KEY-FLDS.
    MOVE ' ' TO W6-01-100.
    IF RECORD-UPDATED NOT EQUAL SPACE
        AND NOT ERRORS-EXIST
        MOVE 'Q' TO RECORD-UPDATED
        PERFORM P800-FIND-RECORD
        MOVE 'Q' TO W6-01-100
        MOVE ' ' TO RECORD-UPDATED. EXIT.
P860-SET-BUTTONS.
    IF W7-01-253 EQUAL 'B' GO TO P865-BACK-UP.
    IF NOT GUI-IN-USE
        AND W7-01-253 NOT EQUAL 'U' EXIT.
P861-READ-NEXT.
    READ FILE01.
    IF STAT-KEY NOT EQUAL '00'
        OR W8-06-000 NOT EQUAL W7-06-010
        GO TO P862-PREVIOUS.
    MOVE W8-06-007 TO WORK-DATE.
    IF WORK-DATE LESS THAN CROSS-REFERENCE-DATE
        GO TO P861-READ-NEXT.
    IF W7-01-253 EQUAL 'U' MOVE W8-11-006 TO W7-11-254
        GO TO P868-RETURN.
    SET-PGDN-NORMAL.
P862-PREVIOUS.
    IF NOT GUI-IN-USE GO TO P864-EXIT.
    MOVE '06' TO W7-02-008. READ-UNIQUE FILE01.
P863-TEST.
    IF STAT-KEY EQUAL '00'
        AND W8-13-000 NOT EQUAL W7-13-010
        MOVE W8-06-007 TO WORK-DATE
```

```
IF WORK-DATE LESS THAN CROSS-REFERENCE-DATE
  READ FILE01 GO TO P863-TEST
ELSE
  SET-PGUP-NORMAL.
P864-EXIT.
  MOVE ' ' TO W7-01-253. MOVE '24' TO W7-02-008. EXIT.
P865-BACK-UP.
  MOVE W7-07-016 TO W7-07-254.
  MOVE '06' TO W7-02-008. READ-UNIQUE FILE01.
P866-TEST.
  IF STAT-KEY EQUAL '00'
    AND W8-13-000 NOT EQUAL W7-13-010
    MOVE W8-06-007 TO WORK-DATE
    IF WORK-DATE LESS THAN CROSS-REFERENCE-DATE
      READ FILE01 GO TO P866-TEST
    ELSE
      MOVE W8-07-006 TO W7-07-254
      READ FILE01 GO TO P866-TEST.
P868-RETURN.
  MOVE 'S' TO W7-01-301. MOVE ' ' TO W7-01-253. RETURN.
```

## Section 5 exercise

```

SECURITY ' '. @ Exercise 5 - READ-EMPLOYEE Technique
@LAST MODIFIED ON: 09-29-94 BY: AUTHOR:
P100-START.
  READ-EMPLOYEE.
P110-STATUS-SELECTION.
  FIND RESULTING-EMP-STATUS STARTING WITH CURRENT-DATE-CYYMDD
  MATCH-SEGMENT-CODE.
  IF NOT FOUND GO TO P900-RETURN.
  IF RESULTING-EMP-STATUS EQUAL '1' OR '5' OR '9'
  GO TO P900-RETURN.
P120-APPRAISAL-RESULTS-SELECTION.
  FIND RATING-VALUE STARTING WITH CURRENT-DATE-CYYMDD
  MATCH-SEGMENT-CODE.
  IF NOT FOUND GO TO P900-RETURN.
P130-INSERT-NEXT-REVIEW-SEGMENT.
  MOVE '010000' TO WORK-TIME-SPAN.
  CALCULATE RATING-DATE + WORK-TIME-SPAN = HOLD-DATE.
  FIND NEXT-REVIEW-TYPE STARTING WITH HOLD-DATE.
  IF FOUND GO TO P900-RETURN.
  MOVE '000100' TO WORK-TIME-SPAN.
  CALCULATE HOLD-DATE - WORK-TIME-SPAN = SAVE-DATE-CYYMDD.
  MOVE '000007' TO WORK-TIME-SPAN.
  CALCULATE HOLD-DATE + WORK-TIME-SPAN = WORK-DATE.
  INITIAL-SEGMENT-AREA.
  MOVE 'LZQ' TO W8-03-000.          @Segment Type/Code
  MOVE HOLD-DATE TO W8-06-003.     @NEXT-REVIEW-DATE
  MOVE 'SA' TO W8-02-009.          @NEXT-REVIEW-TYPE
  MOVE SAVE-DATE-CYYMDD TO W8-06-051. @DISTRIBUTION-DATE
  MOVE WORK-DATE TO W8-06-057.     @DATE-RETURN-EXPECTED
  INSERT-L-SEGMENT.
  WRITE-EMPLOYEE.
P900-RETURN.
  RETURN.

```

```

SECURITY ' '. @ Exercise 5 - UPDATE-EMPLOYEE Technique
@LAST MODIFIED ON: 09-29-94 BY: AUTHOR:
P100-START.
  UPDATE-EMPLOYEE.
P110-STATUS-SELECTION.
  FIND RESULTING-EMP-STATUS STARTING WITH
  CURRENT-DATE-CYYMDD
  MATCH-SEGMENT-CODE.
  IF NOT FOUND UNLOCK-EMPLOYEE GO TO P900-RETURN.
  IF RESULTING-EMP-STATUS EQUAL '1' OR '5' OR '9'
  UNLOCK-EMPLOYEE GO TO P900-RETURN.
P120-APPRAISAL-RESULTS-SELECTION.
  FIND RATING-VALUE STARTING WITH CURRENT-DATE-CYYMDD
  MATCH-SEGMENT-CODE.
  IF NOT FOUND UNLOCK-EMPLOYEE GO TO P900-RETURN.
P130-INSERT-NEXT-REVIEW-SEGMENT.
  MOVE '010000' TO WORK-TIME-SPAN.
  CALCULATE RATING-DATE + WORK-TIME-SPAN = HOLD-DATE.
  FIND NEXT-REVIEW-TYPE STARTING WITH HOLD-DATE.
  IF FOUND UNLOCK-EMPLOYEE GO TO P900-RETURN.
  MOVE '000100' TO WORK-TIME-SPAN.
  CALCULATE HOLD-DATE - WORK-TIME-SPAN = SAVE-DATE-CYYMDD.
  MOVE '000007' TO WORK-TIME-SPAN.
  CALCULATE HOLD-DATE + WORK-TIME-SPAN = WORK-DATE.
  INITIAL-SEGMENT-AREA.

```

```
MOVE 'LZQ' TO W8-03-000.      @Segment Type/Code
MOVE HOLD-DATE TO W8-06-003.  @NEXT-REVIEW-DATE
MOVE 'SA' TO W8-02-009.      @NEXT-REVIEW-TYPE
MOVE SAVE-DATE-CYYMDD TO W8-06-051.  @DISTRIBUTION-DATE
MOVE WORK-DATE TO W8-06-057.  @DATE-RETURN-EXPECTED
INSERT-L-SEGMENT.
WRITE-EMPLOYEE.
P900-RETURN.
RETURN.
```

## Section 6 exercise

```

P XNDRPT 00000 SECURITY ' ' . @ Pay Period Days Table Report XHRUSER
P XNDRPT 00001 @LAST MODIFIED ON: 03-19-99 BY: USER AUTHOR: USER USER
P XNDRPT 00100 DEFINE-REPORT NO-PE-DATES ALLOCATE-20. USER
P XNDRPT 00200 HEADER-1 :60 'Pay Period Days'. USER
P XNDRPT 00300 HEADER-2 :60 ' Table Report'. USER
P XNDRPT 00400 HEADER-3 :002 'Control ' . USER
P XNDRPT 00500 HEADER-4 :002 'Nbr Pay Frequency'. USER
P XNDRPT 00600 HEADER-3 :039 ' Period Period Period'. USER
P XNDRPT 00700 HEADER-4 :039 'End Date Workdays Holidays'. USER
P XNDRPT 00800 P100-START. USER
P XNDRPT 00900 IF W6-06-047 EQUAL ' ' ' USER
P XNDRPT 01000 MOVE CONTROL-1-2 TO W6-06-047 MOVE 'Y' TO W6-01-035 USER
P XNDRPT 01100 SET WORK TO :7 MOVE 'X' TO WORK USER
P XNDRPT 01200 ELSE IF CONTROL-1-2 NOT EQUAL W6-06-047 USER
P XNDRPT 01300 SET WORK TO :7 MOVE 'X' TO WORK RETURN. USER
P XNDRPT 01400 INITIAL-TABLE-AREA. USER
P XNDRPT 01500 MOVE '02' TO KEY01-SIZE. USER
P XNDRPT 01600 MOVE 'X2' TO W7-02-010. @KEY01-AREA USER
P XNDRPT 01700 READ-UNIQUE FILE01. USER
P XNDRPT 01800 IF STAT-KEY NOT EQUAL '00' RETURN. USER
P XNDRPT 01900 P200-SORT. USER
P XNDRPT 02000 MOVE-TABLE-RECORD. USER
P XNDRPT 02100 SPACE-EXTRACT-RECORD. USER
P XNDRPT 02200 OUTPUT '1XNDR' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL USER
P XNDRPT 02300 SORT-LENGTH-22 CONTROL-1-2 X2-ND-CONTROL-NUM USER
P XNDRPT 02400 X2-ND-PAY-FREQ-CD X2-ND-PERIOD-ED-DT. USER
P XNDRPT 02500 P300-REPORT. USER
P XNDRPT 02600 PRINT '0' X2-ND-CONTROL-NUM X2-ND-PAY-FREQ-CD USER
P XNDRPT 02700 X2-ND-PAY-FREQ X2-ND-PERIOD-ED-DT X2-ND-PERIOD-WK-DY USER
P XNDRPT 02800 X2-ND-PERIOD-HL-DY. USER
P XNDRPT 02900 WRITE-EXTRACT. USER
P XNDRPT 03000 P400-READ-NEXT. USER
P XNDRPT 03100 UNLOCK FILE01. MOVE TABLE-C01-40 TO W7-24-010. USER
P XNDRPT 03200 MOVE '24' TO W7-00-008. USER
P XNDRPT 03300 READ-UNIQUE FILE01. READ FILE01. USER
P XNDRPT 03400 IF STAT-KEY EQUAL TO '00' AND W8-02-000 EQUALS 'X2' USER
P XNDRPT 03500 GO TO P200-SORT USER
P XNDRPT 03600 ELSE USER
P XNDRPT 03700 RETURN. USER
RTXNDR 00 01220139114032221001 13:01:25 03-19 USER*
RTXNDR010X2-ND-CONTROL-NUM 4000400210 X2002010 0014 4051530000
RTXNDR020X2-ND-PAY-FREQ-CD 4000100600 X2011010 0032 PP29 1092540014
RTXNDR030X2-ND-PAY-FREQ 4002000600 DX2013010 0047 PP29 1004100032
RTXNDR040X2-ND-PERIOD-ED-DT 4001000700 X2038010 0081 1125210047
RTXNDR050X2-ND-PERIOD-WK-DY 4000302410 X2052010 0105 1125640081
RTXNDR060X2-ND-PERIOD-HL-DY 4000302710 X2061010 0122 1125640105

```

## Section 6 exercise, continued

CORPORATION	99	ACNE MANUFACTURING	Pay Period Days	REPT	FILE	VERSION	00	Page	1
DIVISION	9999	PRODUCTION CTL 1-2	Table Report	XNDR	TIME	12:58	DATE	03-19-2002	
Control Nbr	Pay Frequency	Period End Date	Period Workdays	Period Holidays					
9999	1 Weekly	03-31-2002	005	000					
9999	2 Bi Weekly	03-12-2002	010	000					
9999	3 Semi Monthly	03-31-2002	010	001					
9999	4 Monthly	03-31-2002	020	001					

---

**NOTES**

---



## Appendix B: Extra for Experts

---

### Table of Contents

Activating user-defined files .....	195
FILE23 delivered code for UNIX .....	198
FILE23 UNIX Overrides.....	199
FILE24 delivered code for UNIX .....	200
FILE24 UNIX Overrides.....	200
FILE25 delivered code for UNIX .....	202
FILE25 UNIX Overrides.....	202
PULL Control Record with Overrides: .....	204
System Control Repository Key Structure(s) .....	206
Employee Database Key Structure(s).....	216

## **Activating User-Defined Files**

- 1. Extract COBOL Source**
- 2. Analyze COBOL Source**
- 3. Extract COBOL Source Applying Overrides**
- 4. Compile Program/Link COBOL**

---

### **NOTES**

---

## Activating user-defined files

### COBOL programs

To use the User-defined files in your CSL programs, the CBSV COBOL programs must be modified using overrides. The steps to override the CBSV COBOL for User-defined files Includes:

1. Extract COBOL Source—Use the PULL utility to extract a copy of the CBSV COBOL program.
2. Analyze COBOL Source—Determine what overrides are necessary for your platform.
3. Extract COBOL Source Applying Overrides—Use the PULL utility to extract a copy of the CBSV COBOL program applying the overrides.
4. Compile Program/Link COBOL.



## Activating user-defined files, continued

### Step 1: COBOL Extract

The Solution Series COBOL Source Code programs reside in the CBSV Source File. These programs are extracted from this file using the COBOL Extract (PULL) process. Be sure to include the User File Indicator value in order to extract the User-defined File definition(s).

To execute the COBOL Extract (PULL) program you must create a Control Record in FILE04 containing the following:

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Form Field</b>	23–28	Type PULL into the Program field.
<b>Code-2 Field</b>	30	Optionally, type M into the Code-2 field to indicate the presence of override COBOL code in FILE04. This topic is discussed in more detail later.
<b>Key Field</b>	31–37	Type the name of the COBOL program being ‘PULled’, ending in a period.
	38–40	Type your Operating System Code.
<b>Additional Key Field</b>	41–47	Optionally, type a COBOL PROGRAM-ID, ending in a period, to override the ID.
	49	Type a single digit identifying the user-defined file(s) definitions to be included into the source code. Each file has a file code: FILE23 = 1, FILE24 = 2, FILE25 = 4. These codes can be added together to include various file combinations.

# Analyze COBOL—FILE23

## FILE23 delivered code for UNIX

```

003895  SELECT FILE23  ASSIGN TO  EXTERNAL FILE23
003970      ORGANIZATION IS INDEXED ACCESS IS DYNAMIC
003980      RECORD KEY IS FILE23-KEY STATUS IS STAT-KEY.
      :
      :
011800  FD  FILE23
011840      BLOCK CONTAINS 24 RECORDS LABEL RECORDS ARE STANDARD.
011900  01  FILE23-RECORD.
011920      05  FILE23-KEY          PIC X(???).
011940      05  FILE23-DATA        PIC X(???).
      :
      :
060860      05  W7-36-008RE REDEFINES W7-36-008.
060870      07KEY-SIZE-AND-KEY.
060880          10  KEY23-SIZE          PIC S99.
060900          10  START-KEY23        PIC X(24).
060920          07FILLER              PIC X(10).
      :
      :
065000      25  FILE-AREA-80RE REDEFINES FILE-AREA-80.
065020          30  KEY-23              PIC X(24).
065040          30  FILLER              PIC X(56).
      :
      :
328850  22223-FILE23.  MOVE ZEROS TO STAT-KEY.
328900      MOVE 3060 TO X-LENGTH.
328950          IF RTN3 = 'O'
329000              IF FILE23-STATUS = 'O' GO TO 22290-RESULT
329050              ELSE GO TO 22223-OPEN.
329100          IF RTN3 = 'C'
329150              IF FILE23-STATUS = 'O' GO TO 22223-CLOSE
329200              ELSE GO TO 22290-RESULT.
329250          IF FILE23-STATUS NOT = 'O' GO TO 22223-OPEN.
329350          IF RTN3 = 'R'              GO TO 22223-READ.
329600          IF RTN3 = 'D' GO TO 22223-DELETE.
329650          IF RTN3 = 'T' GO TO 22223-REWRITE.
329800          MOVE 'N' TO START-23.
329850          IF RTN3 = 'B' OR 'S' GO TO 22223-START.
329900          IF RTN3 = 'W' GO TO 22223-WRITE.
329950          IF RTN3 = 'U' GO TO 22290-RESULT.
330000          MOVE '90' TO STAT-KEY.  GO TO 22290-RESULT.
330050  22223-DELETE.  MOVE 'N' TO START-23.
330350      MOVE FILE-AREA TO FILE23-RECORD.
330600      DELETE FILE23 RECORD INVALID KEY GO TO 222891-NRF.
332000      GO TO 22290-RESULT.
332050  22223-CLOSE.
332150      CLOSE FILE23
332550          MOVE 'C' TO FILE23-STATUS.
332600          GO TO 22290-RESULT.

332650  22223-OPEN.
332750      OPEN I-O FILE23
333500          MOVE 'O' TO FILE23-STATUS.
333550          IF RTN3 NOT = 'O' GO TO 22223-FILE23.
333600          GO TO 22290-RESULT.
333650  22223-READ.  IF START-23 NOT = 'Y' GO TO 222891-NRF.
333700  22223-READ-IT.
333750      READ FILE23 NEXT INTO FILE-AREA
333850          AT END GO TO 222893-EOF.
335650          IF STAT-KEY NOT = ZEROS GO TO 22290-RESULT.
336050  22223-CHECK.  IF RTN3 NOT = 'R' GO TO 22223-START-CHECK.
336350          GO TO 22290-RESULT.
336400  22223-REWRITE.
337100          MOVE 'N' TO START-23.
337150      MOVE FILE-AREA TO FILE23-RECORD.
337350      REWRITE FILE23-RECORD
337450          INVALID KEY GO TO 222891-NRF.
339800          GO TO 22290-RESULT.
339850  22223-START.
340100      MOVE KEY23-SIZE TO RTN-SUB.  MOVE 24 TO NBR-CHARS.
340200      MOVE LOW-VALUES TO VAR2.  MOVE RTN-SUB TO NBR-CHARS.
340350      MOVE START-KEY23 TO VAR2.  MOVE 24 TO NBR-CHARS.
340400      MOVE VAR2 TO KEY-23.  MOVE RTN-SUB TO NBR-CHARS.

```

```

341550 MOVE KEY-23 TO FILE23-KEY.
341600 START FILE23 KEY IS NOT LESS THAN FILE23-KEY
341700     INVALID KEY GO TO 222891-NRF.
343650 IF STAT-KEY NOT = ZEROS
343750     GO TO 22290-RESULT.
343800     MOVE 'Y' TO START-23.
344100     GO TO 22223-READ-IT.
344600 22223-START-CHECK.
344800 MOVE START-KEY23 TO VAR1. MOVE KEY-23 TO VAR2.
344850     IF VAR1 NOT = VAR2 MOVE '01' TO STAT-KEY.
345500     GO TO 22290-RESULT.
345550 22223-WRITE.
345600 WRITE FILE23-RECORD OF FILE23 FROM FILE-AREA
345700     INVALID KEY GO TO 222892-DUP.
347450     GO TO 22290-RESULT.

```

**FILE23 UNIX Overrides**

```

011920 05 FILE23-KEY     PIC X(24).
011940 05 FILE23-DATA   PIC X(476).

```

## Activating user-defined files, continued

### Step 2: Analyze COBOL

The second step in activating User-defined files is analyzing the COBOL for each User-defined file you will be using.

The functions to handle file processing have already been included in the standard COBOL paragraphs 222nn, where 'nn' is 23, 24, or 25 indicating the user file. However, you must provide the record length for each file.

#### FILE23

The example for FILE23 is for a UNIX platform, additional overrides may be needed dependant upon your platform:

- Check the SELECT statement. Is it correct for your environment?
- Check the FD statements. Where applicable, make changes to the record size and blocking factor.

*Note: The FILE23-KEY size may not exceed 24 positions, and the total FILE23-RECORD size may not exceed 3060 positions.*

- Review paragraphs 22223 and make machine dependent changes as needed. To eliminate update capabilities, change paragraph P22223-FILE23 code by commenting or deleting statements testing RTN3 (D=Delete, T=Rewrite, W=Write).
- IBM/CICS users must specify a proper FILENAME in the CICS command.
- UNIVAC users must review TIP specifications.

# Analyze COBOL—FILE24

## FILE24 delivered code for UNIX

```
003995  SELECT FILE24 ASSIGN TO  EXTERNAL FILE24.
      :                               :
011960  FD  FILE24
011980  BLOCK CONTAINS 12 RECORDS LABEL RECORDS ARE STANDARD.
      :                               :
012040 01  FILE24-RECORD          PIC X(3060).
      :                               :
347500 22224-FILE24.
347550  MOVE ZEROS TO STAT-KEY.
347600  IF RTN3 = 'R'
347650  IF FILE24-STATUS = 'O' GO TO 22224-READ
347700  ELSE GO TO 22224-OPEN.
347950  IF RTN3 = 'O'
348000  IF FILE24-STATUS = 'O' GO TO 22290-RESULT
348050  ELSE GO TO 22224-OPEN.
348300  IF RTN3 = 'C'
348350  IF FILE24-STATUS NOT = 'O' GO TO 22290-RESULT
348400  ELSE MOVE 'C' TO FILE24-STATUS
348500  CLOSE FILE24
348700  GO TO 22290-RESULT.
348750  MOVE '90' TO STAT-KEY.  GO TO 22290-RESULT.
348760 22224-OPEN.
348770  OPEN INPUT FILE24.
348775  IF STAT-KEY NOT = '00' GO TO 22290-RESULT.
348785  MOVE 'O' TO FILE24-STATUS.
348790  IF RTN3 = 'O' GO TO 22290-RESULT.
348800 22224-READ.
349000  READ FILE24 AT END
349050  MOVE HIGH-VALUES TO FILE-AREA-150 GO TO 222893-EOF.
349100  STRING FILE24-RECORD DELIMITED SIZE INTO FILE-STRING.
349200  GO TO 22290-RESULT.
```

## FILE24 UNIX Overrides

```
012040 01  FILE24-RECORD          PIC X(500).
```

## Activating user-defined files, continued

### FILE24

The example for FILE24 is for a UNIX platform, additional overrides may be needed dependant upon your platform:

- Check the SELECT statement. Is it correct for your environment?
- Check the FD statement. Where applicable, make changes to the record size and blocking factor.

*Note: The FILE24-RECORD size may not exceed 3060 positions.*

- Review paragraphs 22224 and make machine dependent changes as needed.
- IBM/CICS users must specify a proper FILENAME in the CICS command.
- UNIVAC users must review TIP specifications.

# Analyze COBOL—FILE25

## FILE25 delivered code for UNIX

```
004115  SELECT FILE25  ASSIGN TO  EXTERNAL FILE25.
      :                :                :
012060  FD  FILE25
012080  BLOCK CONTAINS 12 RECORDS LABEL RECORDS ARE STANDARD.
012140 01  FILE25-RECORD      PIC X(3060).
      :                :                :
349250 22225-FILE25.
349300  MOVE ZEROS TO STAT-KEY.
349350  IF RTN3 = 'W'
349400  IF FILE25-STATUS = 'O' GO TO 22225-WRITE
349450  ELSE GO TO 22225-OPEN.
349700  IF RTN3 = 'C'
349750  IF FILE25-STATUS NOT = 'O' GO TO 22290-RESULT
349800  ELSE MOVE 'C' TO FILE25-STATUS
349900  CLOSE FILE25
350100  GO TO 22290-RESULT.
350150  IF RTN3 = 'O'
350200  IF FILE25-STATUS = 'O' GO TO 22290-RESULT
350250  ELSE GO TO 22225-OPEN.
350500  MOVE '90' TO STAT-KEY. GO TO 22290-RESULT.
350510 22225-OPEN.
350520  OPEN OUTPUT FILE25.
350525  IF STAT-KEY NOT = '00' GO TO 22290-RESULT.
350535  MOVE 'O' TO FILE25-STATUS.
350540  IF RTN3 = 'O' GO TO 22290-RESULT.
350550 22225-WRITE.
350750  WRITE FILE25-RECORD FROM FILE-AREA.
350850  GO TO 22290-RESULT.
```

## FILE25 UNIX Overrides

```
012140 01  FILE25-RECORD      PIC X(300).
```

## Activating user-defined files, continued

### FILE25

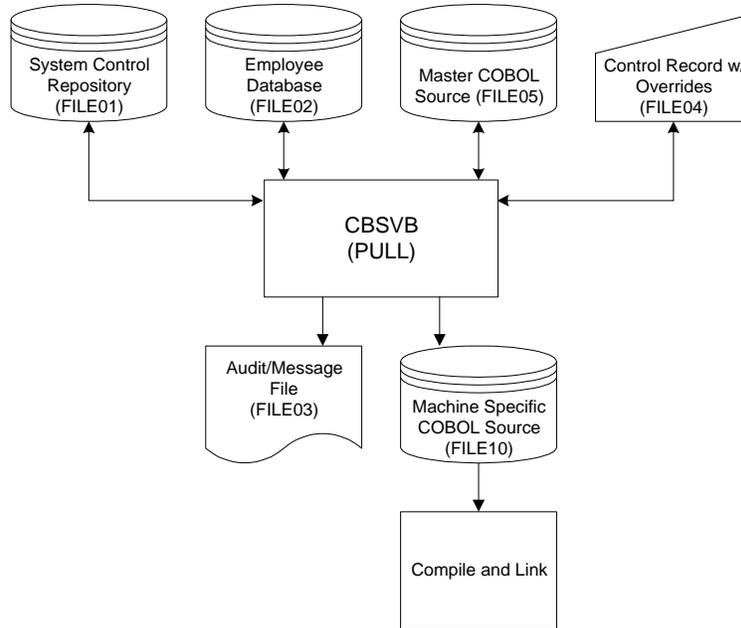
The example for FILE25 is for a UNIX platform, additional overrides may be needed dependant upon your platform:

- Check the SELECT statement. Is it correct for your environment?
- Check the FD statement. Where applicable, make changes to the record size and blocking factor.

*Note: The FILE25-RECORD size may not exceed 3060 positions.*

- Review paragraphs 22225 and make machine dependent changes as needed.
- IBM/CICS users must specify a proper FILENAME in the CICS command.
- UNIVAC users must review TIP specifications.

# Extract The Solution Series COBOL Programs



**Input Files:**

- FILE01 System Control Repository
- FILE02 Employee Database
- FILE04 Control Record File
- FILE05 Master CBSV Source File

**Execute:**

CBSVB

**Output Files:**

- FILE03 Audit/Message File
- FILE10 Extracted CBSV Source Code

**PULL Control Record with Overrides:**

```

1      1      2      2      3      3      4      4      5      5      6      6      7      8
1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...//.0
          PULL  MCBSVO. MF2          7
C011920      05  FILE23-KEY          PIC X(24) .
C011940      05  FILE23-DATA        PIC X(476) .
C012040      01  FILE24-RECORD      PIC X(500) .
C012140      01  FILE25-RECORD      PIC X(300) .
    
```

## Activating user-defined files, continued

### Step 3: Extract COBOL Apply Overrides

The third step in activating User-defined Files is to extract the CBSV COBOL programs with the appropriate overrides. The PULL program is executed with COBOL overrides.

- The Control Record in FILE04 is used to indicate whether overrides are present.
- Override lines of COBOL code are placed in FILE04 following the PULL Control Record.
- When the override indicator is present, the program reads FILE04 and applies the override COBOL code with the CBSV source file to FILE10.

#### **PULL Control Record Modification**

To indicate that override COBOL code is present, change the Control Record in FILE04 with the following:

 Refer to *PULL* for a full explanation of the other parameters.

#### **Code-2 Field**

<u>Position</u>	<u>Option/Step/Description</u>
30	Type M into the Code-2 field to indicates the presence of override COBOL code in FILE04.

### Step 4: Compile and Link

The 4th step in activating User-defined files is to compile and link the CBSV COBOL programs. This is accomplished by following your normal compile and link procedures for a The Solution Series COBOL program.

## System Control Repository Key Structure(s)

### 'A' Records

The 'A' Records contain the Machine Control Cards and Pointer Fields unique to each Operating System platform. These Machine Parameters include the Pointer and System Control Information.

Key structure is:

Column(s)	Value
1	A (literal)
2	Blank
3-12	A COMPUTER (literal) (First record only—subsequent records equal B followed by the machine tailor code and program name)
25-27	Operating system code of computer being used

Object Code(s):

Object	Object Description
A	A Records (Machine Parameters)

### 'B' Records

The 'B' records contain the Working Storage Expansion Parameters that are used when the CBSV programs are created. The Working Storage areas are defined the same way for both CBSVO and CBSVB. Over time, the areas will need to be expanded to accommodate the size of your company's Employee, and Company information. The 'B' Records contain the information the system needs to expand these areas.

Key structure is:

Column(s)	Value
1	B (literal)
2-7	Sequence number

Object Code(s):

Object	Object Description
B	B Records (Expansion Records)

### 'C' Records

The 'C' Records contain the Option List values and description (value translation) for fields using a code scheme. These values are used by The Solution Series to validate data input into specific fields.

Key structure is:

Column(s)	Value
1	C (literal)
2	Blank
3-7	Option List Name
8-21	Option value
22-24	Sequence number

Object Code(s):

Object	Object Description
C	Option List
C/D	Option List Description
C/V	Option List Values

### 'D' Records

The 'D' Records contain defaults for form models that can be set up to contain pre-defined data. These are 'compiled' into form-image records that reside on the System Control Repository (FILE01).

Key structure is:

Column(s)	Value
1	D (literal)
2	Blank
3-8	Control 1-2 value
9-24	Form Key fields and data fields

### 'ECM' Records

The ECM records create context sensitive menu records that allow the user to navigate to related forms directly without using the Menus or Navigator. The ECM records are used in online forms instead of Section 9 Prompts.

Key structure is:

<b>Column(s)</b>	<b>Value</b>
1-3	ECM (literal)
4-9	Form program
10	Blank
11-12	Sequence number
13-24	Blank
25-30	Form program to be accessed by the ECM record
31	Blank
32	'L' to indicate a line separator otherwise leave blank
33	'Y' to indicate a form program otherwise leave blank
34-64	Form name of the program to be accessed by the ECM record

### 'F' Records

The 'F' records contain Cyborg's data dictionary called the Field Name Table. All data fields are defined here with their associated characteristic code and documentation. Each must have a unique name that is part of the Key to the Field Name record. Data Dictionary items include Verbs, Reserved Words, Fields, and File definitions.

Key structure is:

<b>Column(s)</b>	<b>Value</b>
1	F (literal)
2	Blank
3-22	Name of field, reserved word, file or verb
23-24	Sequence number/code

Object Code(s):

<b>Object</b>	<b>Object Description</b>
F	Field Table Entries
F/D	Field Table Documentation
F/S	Field Security
F/V	Verbs
F1	Field Entry
FTM	Field Table Menu
FTX	Field Table Cross Reference

### 'MCL' Records

The 'MCL' records are used to store the updates needed for FILECL.

Key structure is:

<b>Column(s)</b>	<b>Value</b>
1-3	MCL (literal)
3-11	Date (ccyymmdd)
12-15	Time (hhmm)
16-19	Session number
20-23	Transaction number

### 'MMN' Records

The 'MMN' records are used to store the system menus that are placed into the menu bar and Navigator.

Key structure is:

<b>Column(s)</b>	<b>Value</b>
1-3	MMN (literal)
4	Language Indicator (P=primary, A=alternate)
5	Menu number
6-7	1st level Sub menu number
8-9	2nd level Sub menu number

### 'P' Records

The 'P' Records contain all source, object, documentation, error messages, and so forth for CSL/EL forms, reports and utilities reside on the System Control Repository (FILE01) as Program Records.

Key structure is:

Column(s)	Value
1	P (literal)
2	Blank
3-8	File name
9	Record Type
10-14	Sequence number
15	Sequence code

Record Types are:

Value	Description
Blank	CSL/EL Source Code
A	Temporary use (P-RSEQ)
E	Error Recap Records
G	Generated CSL/EL Source Code
J	Sample Control Records and JCL
L	Menu list records
M	Documentation records
R	Messages (Reject/Warning/Memo)
S	Form Appearance Table Records
T	Demonstration (Test) data records
W	Cyborg Assembler Code
X	Object Code

Object Code(s):

Object	Object Description
P	P Records
P/M	P Records—Documentation
P-X	P Records—Except Object Records
P/R	P Records—Error messages
P/	P Records—CSL/EL source
P/S	P Records—Form Appearance Table
P/E	P Records—Reload messages
P/T	P Records—Test data
P/G	Generated CSL/EL Source Code
P/W	P Records—Assembler source
P/H	P Records—Change history
P/X	P Records—Object code
P/J	P Records—Sample JCL

RPT	Report
P/L	P Records—Menu List
RS	Report source

### 'PC' Records

Organization Number Report Validation records are used to specify which reports are to be produced for each Organization Number on your Employee Database (FILE02). The C12RPT form is an optional form that will contain a list of reports valid to be run for each Organization Number. The REPORT program assumes all reports are valid for every Control 1-2 unless the 'PC' (C12RPT Form) records are set up.

Key structure is:

Column(s)	Value
1-2	PC
3-8	Organization Number
9-14	Report Code
15	Blank
16-23	SECURITY verb

Object Code(s):

Object	Object Description
PC	C12RPT—Organization Number Report Validation Records

### 'PD' Records

Restricted Organization Number Report Scheduling Records are optional records, which are used to define which Organization Numbers are to be included in reporting, per the 'PE' (RUNREP Form) record. The RUNC12 form is used to create the list of Organization Numbers for reporting. The 'PD' record name must be identical to the 'PE'/RUNREP record name to which it applies.

Key structure is:

Column(s)	Value
1-2	PD
3-8	RUNREP Name
9-14	Organization Number
15-24	Organization Level 1 Name

Object Code(s):

Object	Object Description
PD	RUNC12—Organization Number Report Scheduling Records

### 'PE' Records

Report Scheduling Records are used to schedule reports. They must have a six-character file name that indicates the report job. 'PE' records are created through the Report Group form and contain a list of reports to be run together. Some reports may require special dates or other parameters in the PARAMETERS field. A header record is required on all report groups.

Key structure is:

Column(s)	Value
1-2	PE
3-8	Report Group Name
9-14	Report Code
15-35	Report Parameters (ALLOCATE-AREA)

Header structure is:

Column(s)	Value
1-2	PE
3-8	Report Group Name
9-23	blanks
24	H
25-64	Title

Object Code(s) are:

Object	Object Description
PE	Report Scheduling Records

**‘PR’ Records**

Position Control Table Records are used to access Position Control records.

Key structure is:

Column(s)	Value
1–2	PR
3–6	PC Control Number
7–16	Position Number
17	Record Type

**‘Q’ Records**

Alternate Key Records are index pointers created using the Master Record Key and other field information. These pointers allow the System Control Repository (FILE01) to access Employee Database (FILE02) records when a Query (on-line report) is executed.

Key structure is:

Column(s)	Value
1	Q (literal)
2–3	Key code
4–22	Key data
23–24	Duplicate Key indicator

Object Code(s):

Object	Object Description
Q	Alternate Key records

**‘R’ Records**

Report Format Records—When batch report source code is compiled, R records are created on the System Control Repository (FILE01).

Key structure is:

Column(s)	Value
1	R (literal)
2	Blank
3–7	Report Code and Type
8–9	Report Record Type

Object Code(s):

Object	Object Description
R	R Records—Report format records

### 'RQM', 'RRM', 'RSM', and 'RXM' Records

System Control Repository (FILE01) records for Solution View Specification are RQM, RRM, RSM, and RXM records. WRITER program to build the Report or Query Source code from the User's entry to the WRITER form(s).

Key structure is:

Column(s)	Value
1-3	RQM, RRM, RXM, or RSM
4-9	Query Report Name (File name)
10	Record Type
11-13	Segment and Segment Code
14	Sequence Number
15-24	Field name

### 'RT' Records

Report Output Position/Totaling Records are used to define output detail and total parameters for a report. Information on this record is used to obtain Field Name Table definition for printing field data on the report.

Key structure is:

Column(s)	Value
1-2	RT
3-7	Report Code and Type
8-9	Sequence Number
10-28	Field Name

Object Code(s):

Object	Object Description
RT	RTEDIT Records—Report Output Position/Totaling

### 'T' Records

Table Record records contain static company information displayed for an employee record based on a code present on the Master Record. The best way to view table records is to execute the table form in Inquiry mode.

Key structure is:

Column(s)	Value
1-2	Table ID
3-24	Table Key fields (varies by table)

Object Code(s):

Object	Object Description
T	Table Records

### 'Y' Records

Security records contain the security access for each individual sign-on sequence, as well as any security violations. Security Records have an encrypted key structure and are discussed in the Security Manual.

Object Code(s):

<b>Object</b>	<b>Object Description</b>
Y	Security Records
Y/V	Security Violation Records

### 'ZL' Records

Locked records are used by the system to determine whether a record is currently being updated. Locked records are temporary and will remain on FILE01 only if an abnormal termination of a program occurs. To remove ZL records from FILE01 use the UNLOCK program.

Key structure is:

<b>Column(s)</b>	<b>Value</b>
1-2	ZL
3-8	Control 1-2
9	Master Record Type

Object Code(s):

<b>Object</b>	<b>Object Description</b>
ZL	Locked Records

## Employee Database Key Structure(s)

### Report Generator Records

Report Generators reside first on FILE02 and are stored in executable form only. Although the Payroll Process uses Report Generators, they are maintained on the Employee Database for the purpose of performing an on-line pay calculation.

Key structure is:

Columns	Value
1-3	Binary record length
4-9	Report Generator Name

### Company Records

Company records contain information on the Company Name & Address, Company HEDs, Control Levels, Pay frequencies, and Pay Run parameters.

Key structure is:

Columns	Value
1-3	Binary record length
4-9	Organization Number
10	Record Type (D)
11-31	Blanks

### Tax Records

Tax Records (Record Type H) contain the tax body data used to calculate the deductions for the pay cycle.

Key structure is:

Columns	Value
1-3	Binary record length
4-9	Organization Number
10	Record Type (H)
11-14	Tax Body ID

## Employee Records

Employee Records follow the corresponding Company Records for the Organization Number on FILE02. The information contained in the Employee Records includes: Name and Address, Labor Split, HEDs, Taxes, and Human Resources data.

Key structure is:

Columns	Value
1-3	Binary record length
4-9	Organization Number
10	Record Type (M)
11-20	Employee Number
21-22	History Unique (Binary or 99)

## 'Other' Records

Record Type can be F or G for other records that follow the company header, or it can be W or X for other records that follow Employee Records. One example of 'Other' records are W records used for processing W2 tax information.

Key structure is:

Columns	Value
1-3	Binary record length
4-9	Control 1-2
10	Record Type (Various)
11-20	Other Key fields

## On-line Pay Calc Records

These are history/labor records created by the PAY-CP program. There are three ZQ records created for each session in which a PAY-CP is processed. These are reused within the session, so only the last calculation is displayed. The three records created contain:

Record 1—Input fields from PAY-CP form.

Record 2—Card stack created from PAY-CP form data: Batch Card, AE-SCR data, Timecard.

Record 3—Results of the calculation, for display.

Key structure is:

Column(s)	Value
1-3	Binary record length
4-5	ZQ
6-9	Session Number

### Executable Code Records

Executable code from the System Control Repository (FILE01) is copied to the Employee Database (FILE02) for each program executed on-line in batch. Allows for faster processing because fewer 'read' operations are required.

Key structure is:

Columns	Value
1-3	Binary record length
4-6	ZX
7-12	Program Name
13	Sequence number (tie breaker)

### Session Records

The first record is the Session Index Record. This record is used when logging on to assign new session numbers. There are other session records, one to a session, displaying the last form executed in the session. When the area directly after the Operator ID is blank it indicates the session was used for a batch job.

Key structure is:

Columns	Value
1-3	Binary record length
4-5	ZY
6-9	Session number
10-13	Operator ID

### Audit Records

Audit Records are form 'snap shots' of information entered in online sessions.

Key structure is:

Columns	Value
1-3	Binary record length
4-5	ZZ or ZI
6-9	Session Number
10-13	Audit Record Number (within the session)

### Time entry/Adjustment Transaction Records

Time entry and Adjustment Transaction 'ZZA' Records are written to FILE02 whenever a time card or adjustment form is entered. These records used in the pay cycle and are written to FILE10 by PAYXTR.

Key structure is:

<b>Columns</b>	<b>Value</b>
1-3	Binary record length
4-6	ZZA
7-12	Organization Number
13-16	Session Number
17-20	Audit Record Number (within session)

---

**NOTES**

---

## Appendix C: The TRACE Utility

---

### Table of Contents

The TRACE utility .....	222
Tracing an entire program online .....	224
Tracing a portion of the program—batch reports .....	225
Tracing paragraphs within a program—TRACE verbs .....	226
Additional trace tips .....	229
Reading the TRACE output .....	231
TRACE examples.....	232

## The TRACE utility

The TRACE utility provides a method that allows you to follow system processes at the conclusion of a program. You have the option of executing TRACE in either online or batch. There are two ways to run a trace on a program:

- Using the TRACE program
- Using the special TRACE- verbs
  - TRACE-ON
  - TRACE-OFF

### TRACE Output

At the conclusion of the program, the TRACE output is written to FILE03. The following discusses the various file outputs written by TRACE:

- In the delivered JCL for PC Solution Series the output, for traces completed in an RPT program, is written to REPORT03.LIS.
- For traces completed in the (-RP) special print program, the output is written to the report output file, RTPRNT03.LIS.
- For traces completed in an on-line program, the output is generally written to file SCREEN.03 where SCREEN is the name of the on-line program being traced. For example, the trace output for form program 40-SCR.03 would be written to 40-SCR.03.

### TRACE output options

You have the option in any of these cases of changing the name of the trace output file by either modifying the JCL to have FILE03 point to a different directory/filename or by typing in a new name on the on-line pop-up when it prompts you for the trace file name. This can be useful when you want to compare trace outputs from different runs of the same program.

- The TRACE program conditions a program for a trace based on the program name and a beginning paragraph label designated in the key field of the control record.
- You must be running either CBSVOT or CBSVBT (Trace version COBOL programs) to execute the TRACE program.

### Tracing an entire program in batch

To trace an entire program in batch, execute TRACE in a batch run. To do this, two control records are needed:

1. The TRACE Control Record—to condition the program for the trace.
2. The Program Control Record—to execute the program to be traced.

#### TRACE batch example

```

          1          2          3          4          5
1...5...0...5...0...5...0...5...0...5...0
P TRACE J00100 999999TRACE XX-RPT 100
          :
          : Beginning
          : paragraph label
          Program to be traced

```

```

          1          2          3          4          5
1...5...0...5...0...5...0...5...0...5...0
P TRACE J00100 999999TRACE XX-RPT 100
P TRACE J00100 999999XX-RPT
          :
          Program to be traced

```

The trace is turned on at the designated paragraph label and continues until the CBSVBT program is executed.

## Tracing an entire program online

### Executing TRACE

Complete the following steps in order to execute a trace:

1. Type TRACE in the Program field in the Command dialog box.
2. Type the name of the program you want to trace in the first six positions of the Key field.
3. Type the paragraph number you want to begin with in the last three positions of the Key field.
4. Press Enter.

**Result:** The system prepares the program for the trace by placing the program name and beginning paragraph number in the field TRACE-HOLD. This field resides in Pointer 7.

5. Type the name of the program you want to trace in the Program field in the Command Dialog.
6. Type any Key field data that the program requires for execution in the Key field.
7. Press Enter.

**Result:** The system begins its trace of the program, beginning with the paragraph number that resides in TRACE-HOLD. The trace continues until you exit CBSVOT by logging off the system.

8. Type GOODBY in the Program field to exit CBSVOT and end the trace.
9. Press Enter key.

**Result:** CBSVOT generates writes trace records to FILE03. Print FILE03 to review the trace records.

## Tracing a portion of the program—batch reports

The following trace example displays how extract records are being created for the report XX-RPT. The trace is done on the batch report during the first pass of the report print process (REPORT) only.

```

          1          2          3          4          5
1...5...0...5...0...5...0...5...0...5...0...5...0
P TRACE J00100 999999TRACE  XX-RPT 100
P TRACE J00100 999999REPORT WEEKLY

```

In order to trace a subroutine during the report print process, or second pass, the control records would look like the following example:

```

          1          2          3          4          5
1...5...0...5...0...5...0...5...0...5...0...5...0
P TRACE J00100 999999TRACE  CYBP10 100
P TRACE J00100 999999RTPRNT

```

In order to trace a special print options program, the control records would look like the following example:

```

          1          2          3          4          5
1...5...0...5...0...5...0...5...0...5...0...5...0
P TRACE J00100 999999TRACE  XX-RP 100
P TRACE J00100 999999RTPRNT

```

## Tracing paragraphs within a program—TRACE verbs

If you want to trace only a small section of a program, you may limit the trace by using special trace verbs:

- **TRACE-ON**

can be inserted anywhere within the source code of your program

- **TRACE-OFF**

must be placed before any RETURN verb and can be placed at the end of any logic you wish to trace

*Note: Remember to use the RELOAD program once the trace verbs have been inserted.*

The trace is executed by running the program to be traced under CBSVOT or CBSVBT. The trace begins at the next line of the paragraph logically following the TRACE-ON verb, and ends when the TRACE-OFF verb is encountered.

*Note: The Solution Series on a PC Network or NT platform do not have CBSVOT or CBSVBT. Instead, the presence of the trace verbs (after recompiling) initiates the trace and, if the TRACE-OFF verb is not used. Entering GOODBY in the Screen field will end the trace.*

Debugging notes are written to FILE03 (when you log off, if running CBSVOT).

## Tracing paragraphs within a program—TRACE verbs, continued

### Sample Program with TRACE verbs inserted

To turn a trace on in your program, insert the verb, TRACE-ON, at the starting point where you wish to trace. You may insert a TRACE-OFF verb to stop the tracing of code at a certain point as well. The program will be traced for all processing between these two verbs. The following provides examples of how to use TRACE- verbs:

```

P 1A-RPT 00000 SECURITY 'HR'. @ ALPHA LIST-ACTIVE & INACTIVE
EMPLOYEES XHR
P 1A-RPT 00001 @LAST MODIFIED ON: 11-03-95 BY: CE
AUTHOR: CE
P 1A-RPT 00003 @Report 1A-RPT provides you with a listing of
all active
P 1A-RPT 00004 @and inactive employees in alphabetical order.
P 1A-RPT 00020 DEFINE-REPORT NO-PE-DATES ALLOCATE-DATE NO-
VERSION-NUMBER.
P 1A-RPT 00040 HEADER-1 :52 'ALPHABETIC LISTING OF ACTIVE'.
P 1A-RPT 00060 HEADER-2 :55 'AND INACTIVE EMPLOYEES'.
P 1A-RPT 00080 HEADER-3 :38 'EMPLOYEE'.
P 1A-RPT 00100 HEADER-3 :51 'CTRL CTRL CTRL'.
P 1A-RPT 00120 HEADER-3 :75 'CTRL MAIL HIRE EMPLOYEE'.
P 1A-RPT 00140 HEADER-4 :3 'EMPLOYEE-NAME'.
P 1A-RPT 00160 HEADER-4 :38 'NUMBER THREE FOUR FIVE SIX'.
P 1A-RPT 00180 HEADER-4 :82 'DISTRIBUTION DATE STATUS'.
P 1A-RPT 00200 P200-SELECT.
P 1A-RPT 00210 TRACE-ON.
P 1A-RPT 00220 FIND-ACTIVITY.
P 1A-RPT 00240 IF NOT-FOUND TRACE-OFF RETURN
P 1A-RPT 00260 IF RESULTING-EMP-STATUS NOT EQUAL '0' AND '3'
AND '4'
P 1A-RPT 00280 TRACE-OFF RETURN.
P 1A-RPT 00300 SET RESULTING-EMP-STATUS TO SAVE.
P 1A-RPT 00320 P300-SORT-OUTPUT.
P 1A-RPT 00340 PRINT '11A-R' FORMS/REPORT-CODE PRINT-GRAND-
TOTAL
P 1A-RPT 00360 DOUBLE-SPACE-BEFORE CONTROL-1 BREAK-DEFAULT
P 1A-RPT 00370 SORT-LENGTH-42 CONTROL-2
P 1A-RPT 00380 EMPLOYEE-NAME-25 EMPLOYEE-NO '1' EMPLOYEE-
NAME
P 1A-RPT 00400 EMPLOYEE-NUMBER.
P 1A-RPT 00420 FIND-LOCATION.
P 1A-RPT 00440 IF NOT-FOUND SPACE-OVER :26
P 1A-RPT 00460 ELSE PRINT CTRL-3-THRU-6 MAIL-DISTRIBUTE-DATA.
P 1A-RPT 00480 SET RESULTING-EMP-STATUS TO SAVE.
P 1A-RPT 00500 PRINT ORIGINAL-HIRE-DATE RESULTING-EMP-STATUS
P 1A-RPT 00520 EMPLOYEE-STATUS.
P 1A-RPT 00540 MOVE :1 TO PERM-01-V0.
P 1A-RPT 00560 OUTPUT PERM-01-V0. WRITE-EXTRACT.
P 1A-RPT 00570 TRACE-OFF.

```

## Tracing paragraphs within a program—TRACE verbs, continued

Optimal version of using the trace verbs to run a trace against a select employee or employees:

```
P 1A-RPT 00210  IF EMPLOYEE-NUMBER EQUAL '1234' OR '3001'  
TRACE-ON  
P 1A-RPT 00220  ELSE TRACE-OFF RETURN.
```

If you only want to test the report by running against a sample set of employees, use the following code.

At the top of the program, initialize a counter field:

```
P 1A-RPT 00210  IF W6-01-035 EQUAL 'F'  
P 1A-RPT 00220  MOVE :0 TO PERM-01-V0.
```

Then loop through the program for as many times as seems appropriate:

```
P 1A-RPT 00210  IF PERM-01-V0 GREATER THAN :100'  
P 1A-RPT 00220  SET WORK TO :7  
P 1A-RPT 00240  MOVE 'X' TO WORK  
P 1A-RPT 00260  RETURN.  
P 1A-RPT 00270  CALCULATE PERM-01-V0 + :1 = PERM-01-V0.  
P 1A-RPT 00280  TRACE-ON.
```

The program will process the first 100 employees on FILE02 for the Control-1-2 specified on the RUNC12 form.

*Note: The first 100 employees will be processed, not necessarily the first 100 employees who meet the selection criteria of the program.*

After the program counter reaches 100, the program will terminate.

## Additional trace tips

Insert a new paragraph label in the program code after the TRACE-ON verb, but just before the area where the suspect code is. This makes finding the specific area in the trace output much easier. If you use an editor program to view the trace output, you can perform a find on the phrase 'LABEL XXX'. Imbed three blank spaces between the word LABEL and the label number, to move the editor program to the beginning of the label area.

Trace output often indicates that a MOVE is being performed but does not show the value that is being moved. A way to get around this and to be able to see the actual value is to create a dummy IF sentence that will never be valid in the program using the value in question. Review the following example:

```
P 1A-RPT 00210  MOVE RATING-VALUE TO W8-01-900.
P 1A-RPT 00220      IF W8-01-900 EQUAL 'Q'
P 1A-RPT 00240      GO TO P900-RETURN.
```

Sometimes you need to know the value in a field or working storage space but no part of the original code is manipulating the value, therefore nothing shows up in the trace. Again, insert a dummy statement in the program after the trace has been started, such as:

```
P 1A-RPT 00210  MOVE W8-01-800 TO W8-01-800.
```

When you use an editor to read the trace output, you can search for paragraph labels in the output by performing a find on phrase 'LABEL XXX' (where XXX is the paragraph number, P100, for example). There must be three spaces between the word 'LABEL' and the 'XXX' paragraph number.

### Tracing a CSL form program

Everything you have just learned will help you read a trace listing of a CSL form, however, there are three unique characteristics to keep in mind:

1. The first characteristic of tracing a GUI form is that the form program generally executes multiple times even though it appears to execute only once to the user. The first time through, the CSL code is typically creating an empty entry form while the second pass is typically accepting the values entered by the user and doing edit check routines.
2. A second characteristic is that some of the CSL code being executed is generated (Object type P/G or EL Source-Generated) based on the SAT (Solution Series Form Appearance Table, Object type P/S or Screen Items Table) when you RELOAD the form. The code generated differs, for instance, if a field is specified as an inquiry field as opposed to an entry field. A switch to generated code is indicated in the main CSL program when you see SCREEN-SECTION-x where the x is a number from 0 to 9. Screen section 0 is usually the line displaying the form title, and for employee level forms, the employee number. Screen section 1 is usually the main entry form, and section 8 is usually the inquiry form display.
3. A third characteristic is that paragraphs may be executed either before or after a display or entry field, based on the (.SAT) Solution Series Form Appearance

Table. That means there may not be a PERFORM statement in the base form program (Object type P/ - P/space - or EL Source) because it is executed by the generated code.

## Reading the TRACE output

### Column 1

This column indicates the program being traced including the CSL program and the CYB programs (if TRACE is allowed to run through to the end).

### Column 2

This column indicates the relative displacement into that specific program (CSL generator).

### Column 3

This column indicates the type of instruction being executed. For example, MOVE, FIND, IF.

### Column 4

This column is the trace of the instruction.

1	2	3	4				
1A-RPT	16	10	IF 1A-RPT	=			REPORT
1A-RPT	28	21	LABEL 200				
1A-RPT	40	24	MOVE Y				
1A-RPT	51	24	MOVE 202E10				
1A-RPT	63	24	MOVE LZC				
1A-RPT	64	07	SET PTR 36 TO 1ST				
1A-RPT	71	30	SEARCH LZC202E10		--LZC215D17		
1A-RPT	72	09	IF LZC202E10		LESS THAN LZC202E10		
1A-RPT	82	13	IF LZC	NOT =		LZC	
1A-RPT	95	13	IF 0	NOT =		0	
1A-RPT	114	07	SET PTR 36 TO SAVE				
1A-RPT	117	21	LABEL 300				
1A-RPT	120	04	MOVE 11A-R				
1A-RPT	128	04	MOVE 99				
1A-RPT	133	04	MOVE 9999				
1A-RPT	138	04	MOVE MEYER JUNE				
1A-RPT	143	04	MOVE 1001				
1A-RPT	147	04	MOVE 1				
1A-RPT	151	04	MOVE MEYER JUNE				
1A-RPT	156	04	MOVE 1001				
1A-RPT	170	24	MOVE 202E10				
1A-RPT	182	24	MOVE LZR				
1A-RPT	183	07	SET PTR 36 TO 1ST				
1A-RPT	190	30	SEARCH LZR202E10		--LZR215D17		
1A-RPT	191	09	IF LZR215D17		LESS THAN LZR202E10		
1A-RPT	201	13	IF LZR	NOT=		LZR	
1A-RPT	212	04	MOVE 3388448855086608				
1A-RPT	216	04	MOVE 11TH-4040				
1A-RPT	220	07	SET PTR 36 TO SAVE				
1A-RPT	227	20	CALC		19840915	19840915	V00
1A-RPT	230	20	EDIT		19840915	19840915	V00
1A-RPT	233	04	MOVE 01				
1A-RPT	241	35	INQUIRY				1
1A-RPT	303	20	CALC		1	1	V00
1A-RPT	311	20	=				
1A-RPT	314	04	MOVE				
1A-RPT	328	13	IF	NOT =			
1A-RPT	348	15	GO TO				
1A-RPT	396	21	LABEL 99				
1A-RPT	399	07	SET PTR 11 TO 1601				
1A-RPT	409	24	MOVE 11A-R999999MEYER JUNE				
1A-RPT	415	24	MOVE 01 338844885508660811TH-4040 09				
1A-RPT	416	22	FILE15 WRITE STATUS 00				
1A-RPT	419	07	SET PTR 11 TO 1601				
1A-RPT	433	24	MOVE Y				
1A-RPT	434	29	RETURN				

## TRACE examples

### Example 1

The following example illustrates how to use a trace in a report program for a single employee. If the results are not what are expected, usually it is preferable to trace a single employee who fits the criteria expected. This partial program selects employees in union 629 and uses their pay-rate to do a table look up to determine the monthly union dues amount.

**These are table records on FILE01 used in the following example:**

```

WUD9999202L280500 0300 00110  **Record that the program needs to read
WUD9999202L280501 0300 00210
WUD9999202L290500 0300 00110
WUD9999202L290511 0300 00110
WUD9999202L290524 0300 00110
WUD9999202L310700 0500 00350
    
```

```

P TS1~PT 00000 SECURITY ' '. @ Local 629 Union Dues Update XHR
P TS1~PT 00001 @LAST MODIFIED ON: 08-19-97 BY: AUTHOR: PGMR
P TS1~PT 00003 @THE UD~RPT will update the employee union dues HED with
P TS1~PT 00004 @the appropriate amount from the UD~SCR table.
P TS1~PT 00005 @
P TS1~PT 00100 DEFINE-REPORT NO-PE-DATES NO-VERSION-NUMBER ALLOCATE-08.
P TS1~PT 00200 HEADER-1 :52 ' LOCAL 629 UNION DUES '
P TS1~PT 00300 HEADER-2 :52 ' '
P TS1~PT 00400 HEADER-3 :01 'EMPLOYEE CONTROL 3 EMPLOYEE'
P TS1~PT 00500 HEADER-4 :01 'NUMBER CODE NAME '
P TS1~PT 00600 HEADER-3 :64 'DUES '
P TS1~PT 00700 HEADER-4 :64 'RATE MESSAGE '
P TS1~PT 00800 P100-INITIALIZE. @INITIALIZE COUNTERS HERE IF NEEDED.
P TS1~PT 00900 MOVE W7-30-176 TO W8-30-850.
P TS1~PT 01000 IF SPACES EQUAL W6-08-036 RETURN.
P TS1~PT 01100 P200-SELECT. @RECORD SELECTION LOGIC GOES HERE.
P TS1~PT 01110 IF EMPLOYEE-NUMBER EQUAL '1234' TRACE-ON. @TEST SUBJECT
P TS1~PT 01120 ELSE TRACE-OFF RETURN.
P TS1~PT 01200 IF UNION-CODE NOT EQUAL '629' RETURN.
P TS1~PT 01300 FIND PAY-RATE STARTING WITH '001'. @GET PAY-RATE
P TS1~PT 01400 IF NOT FOUND RETURN.
P TS1~PT 01500 MOVE PAY-RATE TO WORK-ANNUAL-SALARY. @4 DECIMAL TO 2
P TS1~PT 01600 READ-CONTROL-NUMBER. @FIND CONTROL NUMBER
P TS1~PT 01700 MOVE '17WUD' TO W7-05-008. @KEY LENGTH OF 17
P TS1~PT 01800 MOVE SALARY-GRADES TO W7-04-013.
P TS1~PT 01900 MOVE W6-08-036 TO RING-SAVE-DATE. @DATE FROM PARAMETER
P TS1~PT 02000 MOVE RING-SAVE-DATE TO WORK-DATE. @CENTURY FORMAT
P TS1~PT 02100 MOVE WORK-DATE TO W7-06-017.
P TS1~PT 02200 MOVE WORK-ANNUAL-SALARY TO WORK-AMT-5. @W8-06-724
P TS1~PT 02300 MOVE W8-04-726 TO W7-04-023.
P TS1~PT 02400 READ-UNIQUE FILE01.
P TS1~PT 02500 IF STAT-KEY GREATER THAN '01'
P TS1~PT 02600 RETURN.
P TS1~PT 02700 IF STAT-KEY EQUAL '00'
P TS1~PT 02710 TRACE-OFF
P TS1~PT 02800 MOVE '000000' TO W8-06-724
P TS1~PT 02900 MOVE W8-05-030 TO W8-05-725
P TS1~PT 03000 PERFORM P500-SORT
P TS1~PT 03100 PERFORM P700-UPDATE
P TS1~PT 03200 PERFORM P600-WRITE
P TS1~PT 03300 RETURN.
    
```

## TRACE examples, continued

```

@@@@@ REST OF PROGRAM FOLLOWS .....

ALL LINES THAT BEGIN WITH "***" ARE COMMENTS FOR ILLUSTRATION ONLY AND
ARE NOT A PART OF THE TRACE

TRACE OUTPUT:

** CHECKING UNION CODE, Program line 01200

TS1~PT 109 13 IF 629 NOT = 629

** FIND PAY-RATE, Program line 01300 - 01400

TS1~PT 114 08 SET PTR 32 TO 1ST
TS1~PT 126 24 MOVE H
TS1~PT 138 24 MOVE 001
TS1~PT 149 30 SEARCH H001 --H001
TS1~PT 150 13 IF H001 NOT = H001

** MOVE PAY-RATE TO FIELD WITH 2 DECIMAL PLACES, Program line 01500

TS1~PT 160 20 CALC 50000 50000 V04
TS1~PT 168 20 = 000000500

** READ CONTROL NUMBER AX-SCR, Program line 01600

TS1~PT 185 24 MOVE 16TZAX
TS1~PT 196 24 MOVE 999999
TS1~PT 207 24 MOVE 202E13
TS1~PT 208 22 FILE01 READ-SINGLE STATUS 01 TZAX999999202E13
TS1~PT 221 10 IF TZAX999999 = TZAX999999
TS1~PT 234 24 MOVE TZAX999999274L31 999999999999
TS1~PT 235 15 GO TO

** BUILD KEY TO READ FILE01, LENGTH IN W7-02-008, REST STARTS IN W7-00-010,
Program lines 01700-01900

TS1~PT 268 24 17WUD
TS1~PT 279 24 9999
TS1~PT 290 24 19970104

** CHANGE DATE TO CENTURY AND READ FILE01, Program lines 02000-02400

TS1~PT 296 20 CALC 19970104 19970104 V00
TS1~PT 304 20 = 202L28
TS1~PT 317 24 MOVE 202L28
TS1~PT 323 20 CALC 500 500 V00
TS1~PT 331 20 = 000500
TS1~PT 344 24 MOVE 0500
TS1~PT 345 22 FILE01 READ-UNIQUE STATUS 00 WUD9999202L280500

** CHECK STAT-KEY, Program lines 02500-02600

TS1~PT 358 11 IF 00 GREATER 01
TS1~PT 373 10 IF 00 = 00

** TURN TRACE OFF, Program line 02700

TS1~PT 385 24 MOVE

```

## TRACE examples, continued

### Example 2

This program will read all active employees. If the employees previous Annual Salary (the one before the current) is greater than thirty thousand dollars, a new 'LB6' segment is created for the employee and a report is created with that information.

### Problem

Even though the program uses MOVE-PLACE-TO HOLD1 and RESET-TO-HOLD1-PLACE, the report shows the current annual salary instead of the second, most recent, annual salary.

```
P 17~RPT 00000 SECURITY 'HR'. @ TEST PROGRAM XHR
P 17~RPT 00001 @LAST MODIFIED ON: 08-19-97 BY: S.O. AUTHOR:
P 17~RPT 00003 @Report 17~RPT reads the 'LZM' segment and creates a 'LB6'
P 17~RPT 00004 @segment for active employees whose previous salary is
P 17~RPT 00006 @greater than $30,000.00
P 17~RPT 00100 DEFINE-REPORT NO-PE-DATES ALLOCATE-08 NO-VERSION-NUMBER.
P 17~RPT 00200 HEADER-1 :48 ' TEST PROGRAM TITLE '.
P 17~RPT 00300 HEADER-2 :48 ' '.
P 17~RPT 00400 HEADER-3 :01 'EMPLOYEE EMPLOYEE PREVIOUS '.
P 17~RPT 00500 HEADER-4 :01 'NUMBER NAME ANNUAL-SALARY'.
P 17~RPT 00600 P100-SELECTION.
P 17~RPT 00700 FIND RESULTING-EMP-STATUS.
P 17~RPT 00800 IF NOT FOUND RETURN.
P 17~RPT 00900 IF ACTIVITY-CODE NOT EQUAL '00' RETURN.
P 17~RPT 01000 FIND ANNUAL-SALARY.
P 17~RPT 01100 IF NOT FOUND RETURN.
P 17~RPT 01200 SET ANNUAL-SALARY UP
P 17~RPT 01300 IF L-CARD-CODE NOT EQUAL 'ZF' RETURN.
P 17~RPT 01400 IF ANNUAL-SALARY NOT GREATER THAN :30000.00
P 17~RPT 01500 RETURN.
P 17~RPT 01600 PERORM P500-SORT.
P 17~RPT 01700 MOVE SALARY-CHANGE-TYPE TO W8-03-800.
P 17~RPT 01800 MOVE SALARY-EFFECTIVE TO SAVE-DATE.
P 17~RPT 01900 MOVE SAVE-DATE TO W8-06-810.
P 17~RPT 02000 MOVE-PLACE-TO-HOLD1.
P 17~RPT 02100 FIND RESTRICTION-DT-LB6 STARTING WITH W8-03-800.
P 17~RPT 02200 IF NOT FOUND
P 17~RPT 02300 INITIAL-SEGMENT-AREA
P 17~RPT 02400 MOVE 'LB6' TO W8-03-000
P 17~RPT 02500 MOVE W8-03-800 TO W8-03-003
P 17~RPT 02600 MOVE W8-06-810 TO W8-06-006
P 17~RPT 02700 INSERT-L-SEGMENT
P 17~RPT 02800 MOVE 'Y' TO RECORD-UPDATED.
P 17~RPT 02900 RESET-TO-HOLD1-PLACE.
P 17~RPT 03000 PERFORM P600-WRITE.
P 17~RPT 03100 RETURN.
P 17~RPT 03200 P500-SORT.
P 17~RPT 03300 SPACE-EXTRACT-RECORD.
P 17~RPT 03400 OUTPUT '117~R' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL
P 17~RPT 03500 CONTROL-1-2 NO-PRINT-SUBTOTAL
P 17~RPT 03600 SORT-LENGTH-21 EMPLOYEE-NUMBER.
P 17~RPT 03700 EXIT.
P 17~RPT 03800 P600-WRITE
P 17~RPT 03900 PRINT '1' EMPLOYEE-NAME EMPLOYEE-NUMBER ANNUAL-SALARY
P 17~RPT 04000 WRITE-EXTRACT.
P 17~RPT 04100 EXIT.
```



# TRACE examples, continued

## The trace

```

17~RPT 55 15 GO TO
17~RPT 70 08 SET PTR 36 TO 1ST @@ FIND ACTIVITY LZC SEGMENT
17~RPT 84 24 MOVE LZC
17~RPT 94 30 SEARCH LZC --LZC
17~RPT 95 13 IF LZC NOT = LZC
17~RPT 109 13 IF 00 NOT = 00
17~RPT 114 08 SET PTR 36 TO 1ST
17~RPT 128 24 MOVE LZC
17~RPT 138 30 SEARCH LZC --LZF
17~RPT 139 13 IF LZF NOT = LZF
17~RPT 167 24 MOVE ** 1ST READ **
17~RPT 177 24 MOVE 006000000 @@ FIRST SALARY 60,000.00
17~RPT 178 07 SET PTR 36 UP @@ SET SALARY UP
17~RPT 190 13 IF ZF NOT = ZF
17~RPT 199 20 CALC 5000000 5000000 V02
17~RPT 212 20 COMPARE 3000000 5000000 V02
@@ COMPARE TO 30,000.00
17~RPT 215 14 IF 5000000 NOT GREATER 3000000
17~RPT 246 24 MOVE ** AFTER SET UP ** @@ 2ND MOST RECENT SALARY
17~RPT 256 24 MOVE 005000000 @@ 50,000.00
17~RPT 257 16 PERFORM
17~RPT 638 21 LABEL 500
17~RPT 641 08 SET PTR 11 TO 1601
17~RPT 651 24 MOVE
17~RPT 652 24 MOVE
17~RPT 653 24 MOVE
17~RPT 654 07 SET PTR 11 TO 1601
17~RPT 659 04 MOVE 117 R
17~RPT 667 04 MOVE 999999
17~RPT 672 04 MOVE 1002
17~RPT 676 28 EXIT
17~RPT 269 24 MOVE A02
17~RPT 274 20 CALC 19841001 19841001 V00
17~RPT 282 20 = 841001
17~RPT 295 24 MOVE 841001
17~RPT 296 07 SET PTR 5 TO 36
17~RPT 310 24 MOVE H
17~RPT 341 24 MOVE ** AFTER SAVE PLACE ** @@ AFTER MOVE-PLACE-TO-HOLD1
17~RPT 351 24 MOVE 005000000 @@ AMOUNT STILL 50,000.00
17~RPT 352 08 SET PTR 36 TO 1ST
17~RPT 366 24 MOVE LB6
17~RPT 377 24 MOVE A02
17~RPT 387 30 SEARCH LB6A02 --LO4001 @@ FIND LB6 SEGMENT
17~RPT 388 13 IF LO4001 NOT = LB6A02
17~RPT 401 24 MOVE
17~RPT 407 24 MOVE
17~RPT 419 24 MOVE LB6
17~RPT 430 24 MOVE A02
17~RPT 441 24 MOVE 841001
17~RPT 446 20 CALC 5701 5701 V00
17~RPT 449 18 + 28 6 4149 9850 V00
17~RPT 458 20 + 35 9885 V00
17~RPT 461 18 = 15 7 0 9885 V00
17~RPT 469 20 CALC 5701 5701 V00
17~RPT 472 18 + 36 5 71 5772 V00
17~RPT 476 18 COMPARE 28 7 24957 5772 V00
17~RPT 480 11 IF 5772 GREATER 24957
17~RPT 500 20 =
17~RPT 503 07 SET PTR 36
17~RPT 506 08 SET PTR 8

```

**TRACE examples, continued**

```
17~RPT 509 25 INSERT LB6A02841001 @@ FIND LB6 SEGMENT
17~RPT 510 18 CALC 36 5 71 71 V00
17~RPT 514 18 = 5 4 0 71 V00
17~RPT 518 07 SET PTR 5 TO 36
17~RPT 523 21 LABEL 90
17~RPT 534 10 IF Y = Y
17~RPT 537 07 SET PTR 5 UP
17~RPT 540 18 CALC 5 4 71 71 V00
17~RPT 544 18 + 37 2 9884 9955 V00
17~RPT 548 18 CALC 5 4 71 71 V00
```

---

**NOTES**

---

## Glossary

---

## ACTIVATE-OK-CANCEL

The ACTIVATE-OK-CANCEL verb will activate the OK and CANCEL buttons in a GUI environment.

ACTIVATE-OK-CANCEL.

## AUDIT-nn-FIELD

The AUDIT-nn-FIELD verb causes an IS/WAS entry to be made in a table in memory. It is generated by the Generate Code from Form Appearance Table program (GENER8) as part of the code generated for a radio button.

- Before executing any of these verbs you must move the field name to W8-20-402, the old value to W8-30-300, and the new value to W8-30-270.

AUDIT-COMPANY-FIELD.

AUDIT-EMPLOYEE-FIELD.

AUDIT-OTHER-FIELD.

AUDIT-TAX-FIELD.

## AUTO-HEADERS

The AUTO-HEADERS verb specifies that a field name should appear as a heading on a form. AUTO-HEADERS automatically formats a field heading on a form by using specifications on the Field Name Table. These specifications are set by F-NAME when you add a new field to the system.

- Specify AUTO-HEADERS for each line of entry fields that you want displayed on a form.
- The AUTO-HEADERS verb must precede the ENTRY or INQUIRY verb.

AUTO-HEADERS.

## BEGIN-ENTRY

The BEGIN-ENTRY verb is used in conjunction with the ENTRY or INQUIRY verb and paired with the END-ENTRY verb and is used to find a particular segment occurrence in a multiple occurrence segment.

- BEGIN-ENTRY includes coding for AUTO-HEADERS and does not require it to be coded.
- BEGIN-ENTRY must precede the ENTRY and INQUIRY verbs.

BEGIN-ENTRY.

## BUTTON-ANSWER

The BUTTON-ANSWER verb must be coded at the beginning of any program that uses pushbuttons.

- This verb is coded just once at the beginning of the program.

BUTTON-ANSWER.

## BUTTON-GROUP

The BUTTON-GROUP is used, when a GUI is in use, to indicate that the previous one-position field is to be replaced with a radio button. The current value of the field (?Y' or ?N') determines whether or not it is selected. BUTTON-GROUP moves '%>' to the SCREEN area. A preceding PRINT statement must move a button group number ?n' to the SCREEN area. You may create up to ten different groups of buttons (?n' = 0-9).

- This is used with the **BUTTON-ON** and **BUTTON-OFF** verbs to create a radio button.

**BUTTON-ON**

The **BUTTON-ON** and **BUTTON-OFF** verbs enable you to create a **BUTTON-OFF** one-position entry field whose value is the contents of **YES-NO**.

**BUTTON-ON**      If it is the first pass in a form, **YES-NO** field is set to '?Y', otherwise **WORK** is moved to **YES-NO**.

**BUTTON-OFF**      If it is the first pass in a form, **YES-NO** field is set to '?N', otherwise **WORK** is moved to **YES-NO**.

- This is used with the **BUTTON-GROUP** verb to create a radio button.

**BUTTON-GROUP.****CHECKBOX**

The **CHECKBOX** verb allows you to indicate that the previous one-position field is to be replaced with a checkbox. The current value of the field (?Y' or ?N') determines whether or not it is checked. **CHECKBOX** moves '?:>' to the **SCREEN** area if a **GUI-CLIENT** is greater than a space. The GUI logic looks for the literal '?:>' and creates the checkbox.

- A checkbox should be used only for fields with a Yes/No value (?Y' or ?N').

**CHECKBOX.****CONFIRM-EMP-SEG-ADD**

The **CONFIRM-EMP-SEG-ADD** verb may be used to confirm that a segment will be added to an employee record. This verb should be placed after all field and relational edits have been performed. If a segment is not going to be added to an employee record, warning message PP012W will be displayed.

**CONFIRM-EMP-SEG-ADD.****CURSOR-ACTION-CODE**

The **CURSOR-ACTION-CODE** verb is used to place the cursor at the **ACTION** field on the Command Line, when the form is built.

**CURSOR-ACTION-CODE****CURSOR-ADDITIONAL**

The **CURSOR-ADDITIONAL** verb is used to place the cursor at the **ADDITIONAL-KEY** field on the Command Line, when the form is built.

**CURSOR-ADDITIONAL****CURSOR-CODE-1**

The **CURSOR-CODE-1** verb is used to place the cursor at the **CODE-1** field on the Command Line, when the form is built.

**CURSOR-CODE-1**

## **CURSOR-C1-2**

The CURSOR-C1-2 verb is used to place the cursor at the CONTROL-1-2 field on the Command Line, when the form is built.

CURSOR-C1-2

## **CURSOR-MARK**

The CURSOR-MARK verb is used to place the cursor at the first entry field found after an ENTRY verb, when the form is built. When used this verb must be coded prior to the ENTRY command.

CURSOR-MARK

## **CURSOR-SCREEN-NAME**

The CURSOR-SCREEN-NAME verb is used to place the cursor at the PROGRAM field on the Command Line, when the form is built.

CURSOR-SCREEN-NAME

## **CURSOR-THIS-KEY**

The CURSOR-THIS-KEY verb is used to place the cursor at the KEY field on the Command Line, when the form is built.

CURSOR-THIS-KEY

## **DASH-OFF**

The DASH-OFF verb is used to place a line of dashes across the form.

DASH-OFF

## **DISALLOW-DELETE**

The DISALLOW-DELETE verb will gray the 'Delete This Entry' option found under 'Actions'. This will not stop the user from using ZDELETE to delete an entry.

DISALLOW-DELETE.

## **EMPTY-BOX**

The EMPTY-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

## **END-ENTRY**

The END-ENTRY verb is paired with the BEGIN-ENTRY verb to determine the end of a forms Data Entry Line.

- BEGIN-ENTRY must follow the ENTRY and INQUIRY verbs for a Data Entry Line.

END-ENTRY

**END-ITEM**

The END-ITEM verb enables you to signal the end of the generated code for an item. It is generated only if the SKIP-ITEM phrase was used for this item.

- This verb must be used with the SKIP-ITEM verb.

*END-ITEM*

**END-LABEL**

The END-LABEL verb enables you to test the GUI switch. If a GUI is in use, it will PRINT '?>'. This verb is used after a label is printed. It marks the end of the label for the Windows program.

- This is used after a PRINT statement for a check box, group box, or radio button.

*END-LABEL*

**ENTRY**

The ENTRY verb enables you to create fields that permit users to enter data on a form. These fields are called data entry fields or unprotected fields. The ENTRY verb also causes the current value for the field to be displayed beneath it, unless the field is a Key field.

- Your program must execute an UPDATE verb before it executes the ENTRY verb.
- If you include the optional AUTO-HEADERS and START-LINE verbs in your program, they must be executed before the ENTRY verb.
- Place fields from different segments on separate lines unless they have the same segment Key elements.
- Use the START-LINE and REPEAT verbs with the ENTRY verb if you want the system to locate a stacked segment. Otherwise, the ENTRY verb assumes that you are pointing to the correct segment and position in the record.
- Use the NEXT-LINE verb to mark the end of an ENTRY line.

**ERRORS-EXIST**

The ERRORS-EXIST verb is used to check if any system level errors have occurred. This verb checks to see if SCREEN-ERROR equals an '@' at sign.

*IF ERRORS-EXIST*

*Imperative Statement . . .*

## **FIND**

The FIND verb enables you to locate a specific occurrence of a segment.

- Follow a FIND statement with an IF FOUND or IF NOT-FOUND statement to test the result of the FIND operation.
- Unless you use one of these valid qualifiers, FIND begins at the first segment in the pointer.
- FROM HERE—Begins the FIND operation where the pointer is currently positioned. Be sure that you know the position if you use this qualifier.
- STARTING WITH—Begins the FIND operation at a specified key value.

## **FORCE-OK-CANCEL**

The FORCE-OK-CANCEL verb will activate the OK and CANCEL buttons in a GUI environment and force the 'Save Changes?' prompt to appear.

FORCE-OK-CANCEL

## **FORMAT-CENTURY**

The FORMAT-CENTURY verb is used to format the century of a date. This verb should be used when you need to associate a century to a literal or a RUNREP date, particularly a date that is to be used as a 21st century date. This verb will assume that all dates that have a year within the range of 00-25 will be a 21st century date. The date to be formatted with the century must be moved in to WORK-DATE. Following the execution of FORMAT-CENTURY the new formatted date will reside in WORK-DATE.

For example, if a date were used in a report and supplied to the report via the RUNREP form and needed a 21st century reference, the coding may be as follows:

```
MOVE SPECIAL-YYMMDD TO WORK-DATE.  
FORMAT-CENTURY.  
MOVE WORK-DATE TO SPECIAL-YYMMDD.
```

## **FORMAT-INTER-DATE**

The FORMAT-INTER-DATE is used to convert a date stored in the YYMMDD format into an internationally accepted format of DAY MONTH YEAR. As an example, the date of January 1, 1999 would appear as 01 Jan 99 when printed on a form or report. To convert a date into this format it is necessary to, first, move the date into WORK-DATE and then execute the FORMAT-INTER-DATE verb. The result will reside in INTERNATIONAL-DATE.

```
MOVE SPECIAL-YYMMDD TO WORK-DATE.  
FORMAT-INTER-DATE.  
PRINT INTERNATIONAL-DATE.
```

---

**FORMAT-INTER-DATE-CC**

The **FORMAT-INTER-DATE-CC** is used to convert a date stored in the **YYMMDD** format into an internationally accepted format of **DAY MONTH CENTURY YEAR**. As an example, the date of January 1, 1999 would appear as 01 Jan 1999 when printed on a form or report. To convert a date into this format it is necessary to, first, move the date into **WORK-DATE** and then execute the **FORMAT-INTER-DATE-CC** verb. The result will reside in **INTERNATION-DATE-CC**.

*MOVE SPECIAL-YYMMDD TO WORK-DATE.*

*FORMAT-INTER-DATE-CC.*

*PRINT INTERNATION-DATE-CC.*

**GO-TO-NEXT-SEGMENT**

The **GO-TO-NEXT-SEGMENT** verb is used in forms that selectively display inquiry records. If no occurrences have been selected the **GO-TO-NEXT-SEGMENT** should be used.

Examples of this verb's usage is in the activity forms (01-SCR, 04-SCR, 08-SCR, 95-SCR and 96-SCR).

*GO-TO-NEXT-SEGMENT.*

**GRAYBUTTON**

The **GRAYBUTTON** verb, when a GUI is in use, allows you to indicate that the previous two positions in the **SCREEN** area contain the ID number of a pushbutton.

**GRAYBUTTON** moves '?@>' to the **SCREEN** area to signal the GUI logic to create the pushbutton as a gray non-selected button.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of '?00' to '?19'.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

**GUI-IN-USE**

The **IF GUI-IN-USE** verb tests whether a graphical user interface (GUI) is in use. It tests **W7-01-458** for greater than a space. If true, it means that this user has a graphical interface and the form is in graphics mode.

- This verb should be used only after the **NEW-SCREEN-STYLE** verb has been executed, as that is when **W7-01-458** is set to its proper value.

## GUI-ONLY

The GUI-ONLY verb enables you to write a form which will not run in a non-GUI environment. It tests W7-01-464 for greater than a space. If false, an error message is displayed and a RETURN is executed.

- This verb should be placed after sequence #00010, but before all other Cyborg Scripting Language/English Language (CSL/EL) statements.

## INDENTBOX

The INDENTBOX allows you to indicate that the previous five characters in the SCREEN area contain the height, width, and color of a rectangle that is to be drawn. INDENTBOX moves ?,>' to the SCREEN area. The GUI logic looks for the literal ?,>' and creates the rectangle.

- This verb should be used only when a GUI is in use.
- The 'hhwwc' represents the height (in rows), the width (in columns), and the interior color of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- The color values are:

0= transparent	5= yellow
1= red	6= light blue
2= green	7= reserved
3= blue	8= reserved
4= purple	9= entire rectangle is invisible (tab group)

## INQUIRY

The INQUIRY verb enables you to create protected, or display-only, fields. These fields do not permit users to enter data in them. They generally display the current or default value for the field.

- You can place up to 78 characters, including spaces, on each form line.
- The system must execute a READ or UPDATE verb before the INQUIRY verb.
- You can place both INQUIRY and ENTRY fields on the same form line.
- If you do not want to display data from the first occurrence of a segment.
- Precede the INQUIRY verb with a FIND statement.
- Use the START-LINE/REPEAT verbs and precede the INQUIRY verb with an ENTRY statement that contains all the key fields for the segment.
- Use INQUIRY-EMPLOYEE to display a name in last name, first name format. Position the output in the first 48 positions on the form line to allow for a name up to 30 characters long.

*INQUIRY field-name-1 literal.field-name-2...*

**INQUIRY-MODE**

The INQUIRY-MODE verb is used in an IF statement to check if the user selected inquiry mode. If CODE-1 (W7-01-252) equals an 'I' a true condition will be met.

IF INQUIRY-MODE

*Imperative Statements. . .*

**NEW-SCREEN-STYLE**

The NEW-SCREEN-STYLE verb signals the COBOL code ENTRY verb logic that a non-key field may have the existing value of the field within the entry box. It moves an ?N' to W7-01-460. This verb should be used in all new style forms, that is, those using the Form Appearance Table (SAT). The switch at W7-01-460 is initialized to an ?O' by CYB90 before any form is called.

- This must be executed before any SCREEN-SECTION verbs.

**NEXT-LINE**

The NEXT-LINE verb marks the end of a formatted line and tells the system to display that line on a terminal (output). It also acts as a carriage control indicator.

- End every form line with at least one NEXT-LINE verb, including prompts, entry and inquiry lines, and the Form Title.
- NEXT-LINE moves the form pointer to the beginning of the next form line.
- You can use NEXT-LINE to format a blank display line, or create double-spacing between lines by repeating the NEXT-LINE verb.

**NO-INQUIRY-SELECT**

The NO-INQUIRY-SELECT verb is used in conjunction with the GO-TO-NEXT-SEGMENT verb in the inquiry/selection (SCREEN-SECTION '8') of certain forms. This verb will check to see if any lines have been displayed for this section and if none have, the entry version of the form will be returned.

NO-INQUIRY-SELECT.

**NO-SAVE-CANCEL**

The NO-SAVE-CANCEL verb enables you to inactivate buttons labeled OK and Cancel at the bottom of a form. It places an ?N' in column three of row two if a GUI is in use. This signals the Windows code to not activate the ?OK' and ?Cancel' buttons at the bottom of the form when anything is changed.

- This must be executed prior to exiting form.
- If using form-sections, this must be executed after the SCREEN-SECTION verb.

## PRINT

The PRINT verb moves field data and literal values to Pointer 11 (SCREEN). Unlike the OUTPUT verb, the PRINT verb edits fields according to the edit length and edit routine specified on the Field Name Table.

- PRINT does not edit fields unless they have a specific edit length and edit pattern.
- You can use PRINT to cause field headings and spaces between fields to display on a form.
- If you use PRINT to move data, you must specify O in the PRINT TOTAL field on the RTEDIT form.

## PRINT-MESSAGE

The PRINT-MESSAGE verb is used to access and display memo messages.

## PRINT-REJECT

The PRINT-REJECT verb is used to access and display reject and file error messages.

## PRINT-WARNING

The PRINT-WARNING verb is used to access and display warning messages.

## PROCESS

The PROCESS verb enables you to establish a process loop. A process loop is a series of statements that are executed repetitively. Process loops are an effective way to check multiple-occurrence segments, because the PROCESS logic is executed for each occurrence of a particular segment code.

- If you do not specify an optional qualifier, the PROCESS verb begins at the first occurrence in the stack and continues until it processes all occurrences.
- Qualifiers restrict the process to specific start and end points. Valid qualifiers are:
  - FROM HERE—Begins processing where the pointer is positioned when the program encounters the PROCESS verb.
  - STARTING WITH—Begins processing at a specified key value.
  - ENDING WITH—Ends processing at a specified key value.
- Do not execute a RETURN verb in a process loop.
- You can use the BYPASS-ENTRY verb in a process loop. This verb causes processing to advance to the next occurrence.

---

## **PUSHBUTTON**

The **PUSHBUTTON** verb, when a GUI is in use, allows you to indicate that the previous two positions in the **SCREEN** area contain the ID number of a pushbutton. **PUSHBUTTON** moves ?;>' to the **SCREEN** area to signal the GUI logic to create the pushbutton.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of ?00' to ?19'.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

## **QUERY-FIRST-PASS**

The **QUERY-FIRST-PASS** verb is used with a conditional **IF** statement to determine the timing for initialization or any preprocessing tasks.

## **QUERY-HEADERS**

The **QUERY-HEADERS** verb instructs the program to use default heading information for the fields as defined on the Field Name Table. If used this verb must be specified before any fields are displayed on the form.

- The headings are not painted until the **INQUIRY** verb is used.

## **QUERY-ONLY**

The **QUERY-ONLY** verb instructs the system to inhibit execution of the program to only the **QUERY** Facility. This statement should be the first command coded in a query program.

## **RAISEDBOX**

The **RAISEDBOX** allows you to indicate that the previous five characters in the **SCREEN** area contain the height, width, and color of a raised rectangle that is to be drawn.

- This verb should be used only when a GUI is in use.
- The 'hhwvc' represents the height (in rows), the width (in columns), and the interior color of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- See **INDENTBOX** for color values.

## **READ-COMPANY**

The **READ-COMPANY** verb is used to read the company record data into working storage, and allow inquiry of the data. **READ-COMPANY** uses the data in the **CONTROL-1-2** field as the Key to the record.

## **READ-EMPLOYEE**

The **READ-EMPLOYEE** verb is used to read the employee record data into working storage, and allow inquiry of the data. **READ-EMPLOYEE** uses the data in the **CONTROL-1-2** field and **KEY** field as the Key to the record.

## READ-TAXES

The READ-TAXES is used to read the tax record into working storage, and allow for inquiry of the data. READ-TAXES uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

## RECTANGLE

The RECTANGLE verb allows you to indicate that the previous four characters in the SCREEN area contain the height and width of a group box that is to be drawn. RECTANGLE moves '?!>' to the SCREEN area. The GUI logic looks for the literal '?!>' and creates the group box.

- This verb should be used only when a GUI is in use.
- The '?hhww' represents the height (in rows) and the width (in columns) of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- To label the group box, follow the RECTANGLE verb with a PRINT statement for a literal ending with a greater than (>) symbol. The first position of the literal must not be a space.

## REPEAT :n TIMES

The START-LINE/REPEAT :n TIMES verb combination:

- Indicates the start of a line of form fields.
- Locates the correct segment in a FILE02 record.
- Specifies the number of times that a line of data should be repeated on a form.
- Determines the end of a line of fields to be displayed on a form.
- If you do not use the START-LINE/REPEAT :n TIMES verb combination, the ENTRY or INQUIRY verb assumes that you are pointed at the correct segment and position in the FILE02.
- To produce a single form line, code REPEAT :0 TIMES to end the form line.
- The code for each form line must end with at least one NEXT-LINE verb.
- You can use the verb BEGIN-ENTRY in place of the AUTO-HEADERS and START-LINE verb combination. END-ENTRY can replace REPEAT :n TIMES.

## RESET-RECORD-UPDATE

The RESET-RECORD-UPDATE verb is used to refresh a form. A space will be moved into the RECORD-UPDATED switch (W7-01-097) and an 'S' will be moved into the COMM-ACTION field (W7-01-301).

RESET-RECORD-UPDATE.

## RESET-SCREEN-STYLE

The RESET-SCREEN-STYLE verb is used to reset a form to a non-GUI form after the NEW-SCREEN-STYLE verb has been used.

RESET-SCREEN-STYLE.

---

**SAVE-CANCEL**

The SAVE-CANCEL verb enables you to activate buttons labeled OK and Cancel at the bottom of a form. It places an 'S' in column three of row two if a GUI is in use. This signals the Windows code to activate the 'OK' and 'Cancel' buttons at the bottom of the form when anything is changed. The Generate Code from Form Appearance Table program (GENER8) generates this verb for any form section containing an ENTRY field.

- This must be executed prior to exiting form.

**SCREEN-SECTION**

The SCREEN-SECTION verb is used to display a specific section of a form that was created using Form Builder. This command **MUST** be on its own line. No other code can appear before or after this verb on the same line. This command cannot be used within an IF statement.

SCREEN-SECTION 'X'. 'X' represents the form section from the Form Builder.

**SELECTION-MODE**

The SELECTION-MODE verb is used in an IF statement to check if the user selected the selection mode. If AUTO-KEY-SWITCH (W7-01-448) equals a 'Z' a true condition will be met.

IF SELECTION-MODE

*Imperative Statement . . .*

**SET SCREEN**

The SET SCREEN TO :nn verb changes the current or active address TO :nn of Pointer 11 (SCREEN).

- Positions are counted relative to one (1).
- SET manipulates the fields on the Pointer Table. Therefore, it is important to know where to position the address before you code the SET statement.

**SET-2ND-PANEL**

The SET-2ND-PANEL-UPDATE verb is used in multi panel-UPDATE forms. This verb will force the 2nd panel to update and return the 2nd panel form to the terminal.

SET-2ND-PANEL-UPDATE.

**SET-AUTO-KEY**

The SET-AUTO-KEY verb will move a 'Y' to W7-01-448. This verb is used in forms that have no key fields to insure the data entered for one employee is not passed to the next employee. This verb should be executed in the prompts paragraph.

SET-AUTO-KEY.

## SET-FIELD-ERROR

The SET-FIELD-ERROR verb is used during the field error subroutine. If an error is to be produced by the FIELD-EDIT-ROUTINE this verb sets the indicators to display the message.

This verb moves an '@' at sign into SCREEN-ERROR, W8-01-480, and W8-01-330.

SET-FIELD-ERROR.

## SET-FOR-MESSAGES

The SET-FOR-MESSAGES verb issues 0, 1, or 2 NEXT-LINEs, depending on how close to line 24 you are. You should issue it just before any relational edits are done which might cause error messages. When SET-FOR-MESSAGES is used, the position at which the printed message will appear is set at 0, 1 or 2 lines below the last form field. The number of lines depends on how much room is left on the form after the last field.

- This is used after an ENTRY verb or a SCREEN-SECTION containing an ENTRY verb.

## SET-GRAY

The SET-GRAY verb moves '@' to BUTTON-STATE (W8-01-394).

- These verbs are used to set the state of a user defined pushbutton.

## SET-NORMAL

The SET-NORMAL verb moves ';' to BUTTON-STATE (W8-01-394).

- These verbs are used to set the state of a user defined pushbutton.

## SET-PGxx-y

SET-PGxx-y is a group of verbs where 'xx' is either UP or DN, and 'y' is either GRAY or NORMAL. The use of one of these verbs forces a PGUP or PGDN button on the button bar to an 'on' (normal) or 'off' (gray) setting.

- Typically, these would not be used; the GENER8 program would automatically generate the SET-nn-UPDN verbs.
- If used, these would be used only to code utilities. They would not be used for standard forms, that is, employee- or company-level form.

The SET-PGDN-GRAY verb sets column one of line two to a 'G'.

The SET-PGDN-NORMAL verb sets column one of line two to a 'Y'.

The SET-PGUP-GRAY verb sets column two of line two to a 'G'.

The SET-PGUP-NORMAL verb sets column two of line two to a 'Y'.

## SET-PGDN-OFF

The SET-PGDN-OFF verb will set the left arrow button into a non-selective state.

SET-PGDN-OFF.

**SET-PGUP-OFF**

The SET-PGUP-OFF verb will set the right arrow button into a non-selective state.

SET-PGUP-OFF.

**SET-SCREEN-TO-ENTRY**

The SET-SCREEN-TO-ENTRY verb forces the form into entry mode.

SET-SCREEN-TO-ENTRY.

**SET-SCREEN-TO-INQRY**

The SET-SCREEN-TO-INQRY verb forces the form into inquiry mode.

SET-SCREEN-TO-INQRY.

**SET-nn-UPDN**

SET-nn-UPDN is a group of verbs where 'nn' is the number of a pointer for an application segment which can occur multiple times.

The Generate Code from Form Appearance Table program (GENER8) generates this verb after the 'REPEAT :0 TIMES' verb. It determines if there are additional segments of the same type, both up and down from the current occurrence. Then, if a GUI is in use and there is more than a single occurrence, columns one and two of the second line of the form are set to 'Y' (normal) or 'G' (gray). The Windows logic will generate pushbuttons labeled Page Up, Page Down, and Select. If column two of line two is a 'G' the Page Up button will be grayed.

If the program contains more than one page, the PGDN button will be normal when a page other than the last page is displayed. The PGUP button will be normal when a page other than the first page is displayed.

- The GENER8 program automatically generates this.

**SKIP-ITEM**

The IF [condition] SKIP-ITEM verb allows control to be transferred down to the next END-ITEM statement. The Generate Code from Form Appearance Table program (GENER8) generates this verb.

- This verb may be used to skip over form formatting and positioning verbs such as ENTRY, PRINT, and SET SCREEN. It is not recommended as a way of skipping over other types of verbs.

**SPACE-OVER :nn**

The SPACE-OVER :nn verb moves the specified number of spaces to Pointer 11 (SCREEN).

- :nn is a two-digit numeric literal value from 01 through 60.

## START-LINE

The START-LINE/REPEAT :n TIMES verb combination:

- Indicates the start of a line of form fields.
- Locates the correct segment in a FILE02 record.
- Specifies the number of times that a line of data should be repeated on a form.
- Determines the end of a line of fields to be displayed on a form.
- If you do not use the START-LINE/REPEAT :n TIMES verb combination, the ENTRY or INQUIRY verb assumes that you are pointed at the correct segment and position in the FILE02.
- To produce a single form line, code REPEAT :0 TIMES to end the form line.
- The code for each form line must end with at least one NEXT-LINE verb.
- You can use the verb BEGIN-ENTRY in place of the AUTO-HEADERS and START-LINE verb combination. END-ENTRY can replace REPEAT :n TIMES.

## STORE-INQUIRY-ONLY

The STORE-INQUIRY-ONLY verb will move the content of the COMM-CHAR field (W7-01-298) in to W6-01-298. It is used to test for an inquiry only condition when executing a table form.

*STORE-INQUIRY-ONLY.*

## TIME-TO-PRINT-TITLE

The TIME-TO-PRINT-TITLE verb is used with a conditional IF statement to determine the timing for printing the Query Title at the top of each form.

## UPDATE-COMPANY

The UPDATE-COMPANY verb is used to read the company record data into working storage, and allow updating of the data. UPDATE-COMPANY uses the data in the CONTROL-1-2 field as the Key to the record.

## UPDATE-EMPLOYEE

The UPDATE-EMPLOYEE verb is used to read the employee record data into working storage, and allow updating of the data. UPDATE-EMPLOYEE uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

## UPDATE-TAXES

The UPDATE-TAXES verb is used to read the tax record into working storage, and allow updating of the data. UPDATE-TAXES uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

**USER-BUTTON**

The USERBUTTON verb, when a GUI is in use, allows you to indicate that the previous two positions in the SCREEN area contain the ID number of a pushbutton. USERBUTTON moves BUTTON-STATE followed by '?>' to the SCREEN area to signal the GUI logic to create the pushbutton.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of ?00' to ?19'.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

**VALUE-BELOW-BOX**

The VALUE-BELOW-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

VALUE-BELOW-BOX.ENTRY field-name-1 literal...field-name-2...

**VALUE-IN-BOX**

The VALUE-IN-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

VALUE-BELOW-BOX.ENTRY field-name-1 literal field-name-2...

**WARNINGS-EXIST**

The WARNINGS-EXIST verb is used to check if any warnings exist. This verb will check if SCREEN-WARNING (W7-01-096) equals a 'W'.

IF WARNINGS-EXIST

*Imperative Statement . . .*

---

**NOTES**

---





- 'A' Records ..... 206  
 'B' Records ..... 206  
 'C' Records ..... 207  
 'D' Records ..... 207  
 'ECM' Records ..... 208  
 'F' Records ..... 208  
 'MCL' Records ..... 209  
 'Other' Records ..... 217  
 'P' Records ..... 210  
 'PC' Records ..... 211  
 'PD' Records ..... 212  
 'PE' Records ..... 212  
 'PR' Records ..... 213  
 'Q' Records ..... 213  
 'R' Records ..... 213  
 'RQM', 'RRM', 'RSM', and 'RXM' Records ..... 214  
 'RT' Records ..... 214  
 'T' Records ..... 214  
 'Y' Records ..... 215  
 'ZL' Records ..... 215  
 5M-RP program ..... 147  
 Activating user-defined files ..... 195  
 Applying the transaction file ..... 135  
 Audit Records ..... 218  
 Batch transaction updating ..... **135**  
 BATCHL program ..... 135  
 Bypass employees ..... 163  
 Bypass Employees ..... 157  
 CALC-CODE-SET ..... 67  
 Calculation option list ..... 59  
 Calling option list logic ..... 67  
 CLOSE ..... 33  
 Company Records ..... 216  
 Compile calc option list ..... 65  
 Compile Calculation Code Set form (RECALC) ... 65  
 Compile Edit Code Set form (REEDIT) ..... 65  
 Compile edit option list ..... 65  
 Compiling option list logic ..... **65**  
 Course introduction ..... **3**  
 Course logistics ..... **5**  
 Course materials ..... **7**  
 Create A Read Verb form (RDVERB) ..... 27  
 Creating option list logic ..... **61**  
 Custom READ- verbs ..... 27  
 DELETE ..... 45  
 DELETE rules ..... 131  
 DELETE-GLOBAL ..... 47  
 Deleting a segment ..... **131**  
 Deleting records ..... **45**  
 Display System Control Repository utility (DSP01)107  
 Edit Utility (EDIT) ..... 61  
 EDIT-CODE-SET ..... 67  
 Employee Database ..... 15  
 Employee Database Key Structure(s) ..... 216  
 Employee Records ..... 217  
 Entry mode design ..... 101  
 Executable Code Records ..... 218  
**Executing TRACE** ..... 224  
 Extract Program ..... 149, 153  
 Field Name/Maintenance utility (F-NAME) ..... 87  
 File access ..... 37  
 FILE01 READ- verb ..... 29, 109  
 FILE01 reports ..... 163  
 FILE02 alternate ..... 23  
 FILE02 READ- verb ..... 29  
 FILE23 ..... 199  
 FILE23 READ- verb ..... 29  
 FILE24 ..... 201  
 FILE25 ..... 203  
 Form Body ..... 101  
 Form design ..... 101  
 Form Header ..... 101  
 Form image layouts ..... 135  
 Form image transactions ..... 135  
 Function of option lists ..... 59  
 Group delete ..... 47  
 INITIAL-SEGMENT-AREA verb ..... 123  
 Input and output functions ..... 13  
 Input only files ..... 13  
 Input/output files ..... 13  
 INSERT rules ..... 123  
 INSERT- verbs ..... 123  
 INSERT-PTR36 BATCH ..... 127  
 Key field design ..... 83  
 Key field order ..... 83  
 Multi-part records ..... 111  
 Multi-part table layout ..... 81  
 Multiple employee format ..... 153  
 On-line Pay Calc Records ..... 217  
 OPEN ..... 33  
 Option list logic ..... 63  
 Option lists overview ..... **59**  
 Other key fields ..... 83  
 Output only files ..... 13  
 P100 ..... 155, 161  
 P140 ..... 161  
 P200 ..... 151, 155  
 P260 ..... 155  
 Parameter Validation ..... 157  
 Percentage of total report ..... **149**  
 Print Position Record (RTEDIT) ..... 149, 153  
 Print report parameters ..... 161  
 Random files ..... 31, 39  
 READ ..... 31  
 Read status ..... 111  
 READ- verb utility ..... 109  
 READ-COMPANY ..... 23  
 READ-EMPLOYEE ..... 23  
 READ-HISTORY ..... 25  
 READ-HL ..... 25  
 Reading FILE01 ..... 19  
 Reading FILE02 ..... 19  
 Reading FILE23 ..... 19

Reading files sequentially.....	31	Table identifier .....	81
Reading random files.....	17	Table key fields.....	81
Reading the TRACE output.....	231	Table layout	
READ-LABOR.....	25	example.....	85
READ-TAXES.....	23	Table record layout .....	81
READ-UNIQUE.....	19	Tax Records .....	216
Recommended update technique .....	137	The Solution Series files .....	13
Record identifier.....	81	The TRACE utility.....	222
Record length.....	39	Time entry/Adjustment Transaction Records.....	219
Record retrieval random files .....	17	<b>TRACE batch example</b> .....	223
Record WRITE.....	127	TRACE examples.....	232
Relational edit option list.....	59	<b>TRACE Output</b> .....	222
Report Generator Records .....	216	<b>TRACE output options</b> .....	222
Report schedule parameters.....	157	trace tips.....	229
REWRITE .....	43	TRACE-OFF .....	226
Rewrite a record .....	43	TRACE-ON.....	226
<b>Sample Program with TRACE verbs inserted</b>	227	<b>Tracing a CSL form program</b> .....	229
Screen error .....	25	Tracing a portion of the program—batch reports.....	225
SCREEN work area.....	137	<b>Tracing an entire program in batch</b> .....	223
Segment deletion .....	131	Tracing an entire program online .....	224
Segment updating .....	123	Tracing paragraphs within a program—TRACE verbs .....	226
Select/Inquiry mode design .....	101	UNLOCK .....	37
Sequential files .....	33	UNLOCK- verbs .....	129
Session Records.....	218	Unlocking files .....	37
Special.....	155	UPDATE- verbs .....	129
Special print option paragraphs .....	147	User-defined files.....	15
Special Print Options Program .....	149, 153	User-defined table entry form .....	99
Special Print Program paragraphs.....	151	User-defined tables.....	77
Standard Sort .....	159	Variable Sort .....	159
STAT-KEY.....	21, 33	Verify form and fields.....	105
Supervisor read verb.....	109	Verify table layout .....	107
System Control Repository.....	15	WRITE.....	39
System Control Repository Key Structure(s) .....	206	WRITE- .....	41
Table analysis.....	79	WRITE- verbs .....	129
Table data .....	81	WRITE-EXTRACT.....	41
Table data requirements.....	79	WRITE-FILE10.....	41, 137
Table field definitions.....	87	Writing and rewriting records .....	39

---

# **Cyborg Scripting Language Customization Participant Guide**

Document Issue: 5.0

CYBORG

Document Issue Status: Fifth Release  
Document Issue Level: 5.0  
Document Issue Date: June 2003  
Software Version: 5.1

### **Copyright Notice**

Copyright© 2003 by Cyborg Systems Inc. All rights reserved.

No part of this publication may be copied or distributed, transmitted, stored in a retrieval system or translated into any human or computer language in any form or by any means, electronic, mechanical, magnetic, manual or otherwise, or disclosed to third parties without the express written permission of Cyborg Systems, Inc.

### **Disclaimer**

This document relates to the current published version of the product. Every effort has been made to supply complete and accurate information. However, all information found herein is subject to change without notice and Cyborg Systems, Inc. shall not be held accountable for any subsequent changes or modifications made to the contents of this document.

All information contained herein is as accurate as is humanly possible. If any statement is unclear, please contact Customer Support for clarification. No statement is intended to be vague or misleading.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series® are registered trademarks of Cyborg Systems, Inc.

The ASP Solution™ is a trademark of Cyborg Systems, Inc.

All other names are trademarks or registered trademarks of their respective owners.



Cyborg Systems, Inc., Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

# Contents

<b>Contents</b>	<b>i</b>
<b>Section 1: Course Overview</b>	<b>1</b>
Course introduction .....	3
Course logistics .....	5
Course materials .....	7
<b>Section 2: Designing a Form</b>	<b>9</b>
Overview .....	11
Form design applications and The Solution Series .....	13
Designing a new form .....	17
Starting the form design application .....	19
Naming the new form .....	21
Establishing modes .....	25
Designing the form .....	27
Creating the header .....	29
Creating the body .....	33
Tab sequence .....	49
Section 2 exercise 1 .....	51
Select/Inquiry form section .....	53
Section 2 exercise 2 .....	59
Section summary .....	61
<b>Section 3: Customization Basics</b>	<b>63</b>
Overview .....	65
The form design application and CSL .....	67
Form programming .....	69
Context-sensitive menu records .....	79
Extracting the form .....	81
Generating the form .....	83
Extracting the form in batch .....	85
Generating the form in batch .....	87
Section summary .....	89
Section 3 exercise .....	91

<b>Section 4: Defining Fields</b>	<b>93</b>
Overview.....	95
The Solution Series field reports .....	97
Creating option lists .....	99
Field definitions .....	101
Section summary.....	127
Section 4 exercise .....	129
<b>Section 5: User-Defined Segments</b>	<b>133</b>
Overview.....	135
User-defined segments overview .....	137
Data requirements analysis .....	139
User-defined segment layout .....	141
User-defined field definitions .....	145
User-defined segment layout—continuation segment.....	153
User-defined entry form.....	155
Verify user-defined definitions.....	157
Query-related programs .....	161
Section summary.....	167
Section 5 exercise .....	170
<b>Section 6: Customization Basics Plus</b>	<b>173</b>
Overview.....	175
Program processing.....	177
Form messages.....	179
Creating report parameter entry forms (RGMSTR) .....	193
Section summary.....	195
Section 6 exercise .....	197
<b>Section 7: Online Query Programming</b>	<b>199</b>
Overview.....	201
Query programming.....	203
Maintaining alternate keys.....	207
Section summary.....	221
Section 7 exercise .....	223
<b>Appendix A: Exercise Answers</b>	<b>225</b>
Section 2 exercise 1 .....	226
Section 2 exercise 2 .....	229
Section 3 exercise .....	233
Section 4 exercise .....	234
Section 5 exercise .....	237
Section 6 exercise .....	245
Section 7 exercise .....	247

---

<b>Appendix B: Solution View</b>	<b>249</b>
The new fields definition form series .....	251
Selecting the action .....	253
Titling and describing the new form .....	255
Defining the fields .....	259
Accessing the new form .....	263
New form sample .....	265
<b>Appendix C: Quick Solution—Technical Reference Guide</b>	<b>267</b>
System Control Repository object codes .....	268
Operating system codes .....	269
Edit command list .....	270
<b>Appendix D: Pointer 7 Fields</b>	<b>275</b>
Pointer 7 fields .....	276
Pointer 7 fields in pointer order: .....	277
Pointer 7 fields in alphabetical order: .....	278
Pointer 7 field definitions .....	279
<b>Appendix E: Form Footers Using Section 9</b>	<b>289</b>
Creating the screen footer .....	291
<b>Glossary</b>	<b>293</b>
<b>Index</b>	<b>311</b>

---

---

**NOTES**

---

# Section 1: Course Overview

---

## Table of Contents

Course introduction.....3  
Course logistics.....5  
Course materials .....7

## Course Introduction

- **Purpose and benefits**
- **Audience**
- **Versions**
- **Prerequisites**
- **Goals**
- **Expectations**

---

NOTES

---

## Course introduction

### **Purpose**

This course will teach you the skills necessary to design Cyborg Scripting Language (CSL) data entry programs.

### **Benefits**

The benefit of learning this information is that you will acquire experience in programming data entry programs.

### **Audience**

This course has been designed for project team members or data processing personnel who are responsible for creating data entry forms, online reports, and user-defined fields.

### **Prerequisites**

Before taking this course you should have completed the following Cyborg courses:

- Using The Solution Series: Administrative Solutions
- Introduction to Cyborg Scripting Language

### **Goals**

At the conclusion of this course you should be able to:

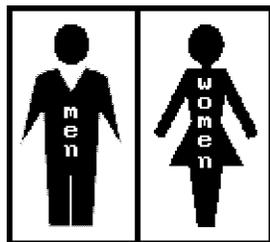
- Create online data entry forms for existing or user-defined fields.
- Create Cyborg Scripting Language programs to display lists of data on a screen (Query).

### **Expectations**

To achieve the goals of this course you should:

- Ask questions.
- Share examples of your own Cyborg-related experiences. This sharing of information among participants enhances the learning process.
- Ask where to obtain additional information if you have an interest in a point that is introduced.

# Logistics



---

## NOTES

---

## Course logistics

Use the space below and in the right column to take notes about the course logistics.

### Meals

---

---

---

---

### Breaks

---

---

---

---

### Telephones

---

---

---

---

### Restrooms

---

---

---

---

### Security questions

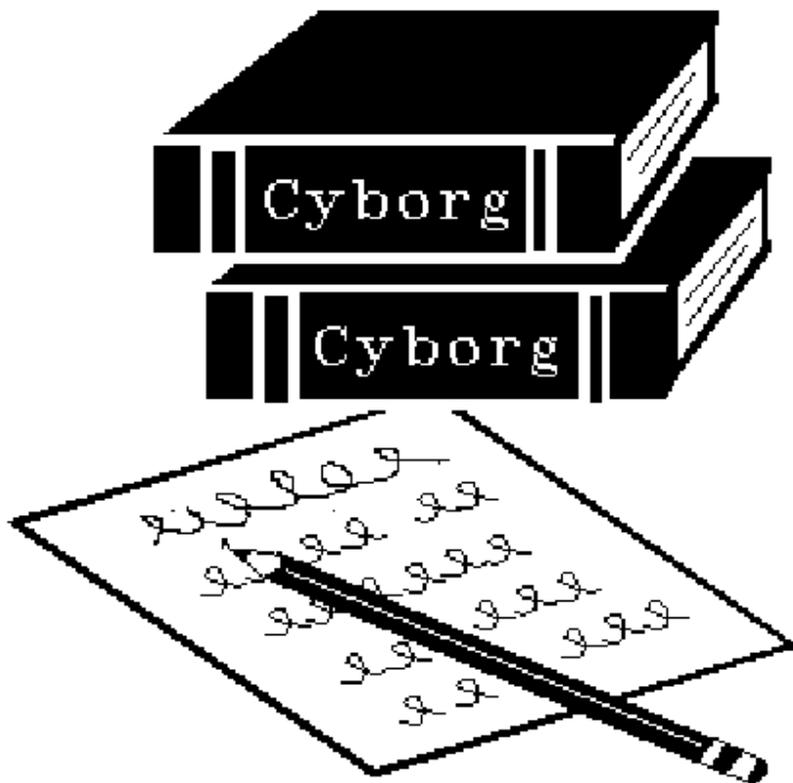
---

---

---

---

# Course Materials



---

**NOTES**

---

## Course materials

### Course materials

All form illustrations are shown for the Windows version of The Solution Series. Instructions to access or complete a form are provided using Windows. Keyboard (non-mouse) users can refer to the ‘Command:’ text under each step, where applicable.

### Table of contents

Each section has a table of contents listing on the section title page.

### Text layout

This guide is designed in the following manner:

- Left pages typically contain copies of overhead transparencies or forms.
- Right pages contain information about the overhead transparency or form and an area for your note taking.

### Section exercise

Exercises give you an opportunity to practice what you have learned in each section. All sections except the course overview section have exercises.

### Appendixes

The appendixes are in the back of your participant guide and contain the following:

- Exercise Answers—Answers to section exercises.
- Solution View—Creating A User-Defined Segment and Entry Form.
- Quick Solution—Technical Reference Guide.
- Pointer 7 Fields
- Form Footers Using Section 9

### Glossary

Glossary and syntax for the CSL verbs.

### Index

An alphabetical listing of content cross-referenced to page numbers.

---

**NOTES**

---

## Section 2: Designing a Form

---

### Table of Contents

Overview.....	11
Form design applications and The Solution Series.....	13
Designing a new form.....	17
Starting the form design application.....	19
Naming the new form .....	21
Establishing modes .....	25
Designing the form .....	27
Creating the header .....	29
Creating the body.....	33
Tab sequence.....	49
Section 2 exercise 1 .....	51
Select/Inquiry form section.....	53
Section 2 exercise 2 .....	59
Section summary.....	61

## Objectives

- **Identify the role a form design application plays in the design and development of a form program**
- **Identify and create form components**
- **Identify the purpose of form sections**
- **Use the form design application to design a form**

---

NOTES

---

## Overview

### **Purpose**

In this section, you will learn how to use a form design application to design a form.

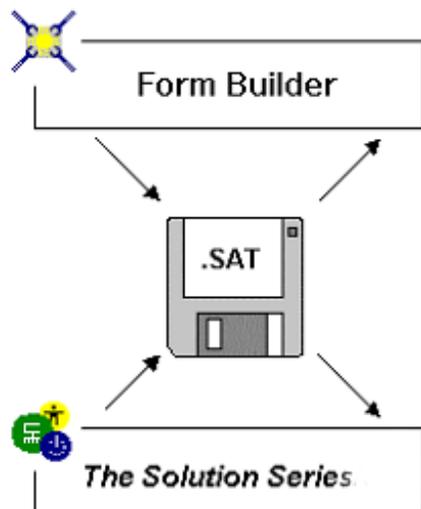
### **Objectives**

When you complete this section, you will be able to:

- Identify the role the form design application plays in the design and development of a form program.
- Identify and create The Solution Series form components:
  - Header
  - Body
  - Select/Inquiry
- Identify the purpose of sections.
- Use a form design application to design a form.

# Form Design Applications

## Form Builder



---

## NOTES

---

## Form design applications and The Solution Series

### Design applications and Cyborg Scripting Language

The Form Builder software is used in conjunction with Cyborg Scripting Language (CSL) programming to create and maintain The Solution Series forms.

- Both design applications are visual programming tools that allow you to create and maintain form attributes and layout.
- CSL is used to display the form design, validate the data entered into the form, and update the database with the valid results.

The relationship between design applications and The Solution Series is the Form Appearance Table (SAT) for each form.

- The Form Appearance Table contains the attributes and layout of a form.
- The Solution Series uses the Form Appearance Table to generate the CSL code that displays the attributes and layout you create.

The focus of this section is to design a basic form with a Header, Body, and Select/Inquiry.

☞ *Refer to Section 3: Customization Basics for completion of the form development process.*

# Entry Mode

Company Earnings

HED> 001  
Description: REGULAR PAY

Category: Basic-Normal Tax	Pay On Vacation Pay: <input checked="" type="radio"/> Yes <input type="radio"/> No
Frequency: Inactive	Priority Overrides:
Taxability: Fully Taxable	Permanent Order: 001 <input type="checkbox"/>
Calc Method: Use HED 001 Rate	Temporary Order: 001 <input type="checkbox"/>
Amount/Percent: <input type="text"/>	
TE-2 Hours: Not In Use	TE-2 Amt: Not In Use

# Select/Inquiry Mode

HED	Description
<input type="checkbox"/>	001 REGULAR PAY
<input type="checkbox"/>	002 FUTURE RAISE
<input type="checkbox"/>	003 OVERTIME PAY
<input type="checkbox"/>	004 OVERTIME PAY
<input type="checkbox"/>	005 BONUS
<input type="checkbox"/>	006 SICK PAY
<input type="checkbox"/>	007 TIPS
<input type="checkbox"/>	008 VACATION
<input type="checkbox"/>	009 HOLIDAY
<input type="checkbox"/>	011 2ND SHIFT PREM
<input type="checkbox"/>	012 3RD SHIFT PREM
<input type="checkbox"/>	013 1ST SHIFT
<input type="checkbox"/>	015 EXPENSE 1
<input type="checkbox"/>	023 COMMISSIONS
<input type="checkbox"/>	027 ANNUAL BONUS
<input type="checkbox"/>	034 TRANS SUBSIDY
<input type="checkbox"/>	035 HOUSING ALLOW
<input type="checkbox"/>	044 CALENDAR MEMO
<input type="checkbox"/>	047 FISCAL MEMO

---

## NOTES

---

## Form design applications and The Solution Series, continued

### Form sections

The Solution Series forms are divided into sections with each section containing a specific portion of the form. Normally, The Solution Series forms are defined using at least three sections, however more complex forms can be defined using up to ten sections. Sections are used to create:

- Header—Section 0
- Form Body—Sections 1–7
- Select/Inquiry—Section 8

*Note: Section 9 may contain screen prompts that provide links to related screens. Further information on Section 9 may be found in Appendix E: Form Footers Using Section 9. Context-sensitive menus are recommended instead of prompts on all new development.*

### Form display modes

There are two display modes that must be accounted for when designing a form. The form display modes are:

- Entry mode—Displays the form with all fields displayed and ready for update:
  - This is the mode when a form is initially displayed.
  - This mode displays the same for single-occurrence and multiple-occurrence segment forms.
- Select/Inquiry mode—Displays the form in one of two formats dependent on whether there is one or multiple occurrences for the segment
  - One Occurrence of the Segment—The form displays the same as entry mode.
  - Multiple Occurrences of the Segment—The form displays a summary of each occurrence that can be selected for full form entry mode.

## **Steps to Designing a New Form**

- **Starting the form design application**
- **Naming the new form**
- **Designing the form**
- **Saving the form**

---

**NOTES**

---

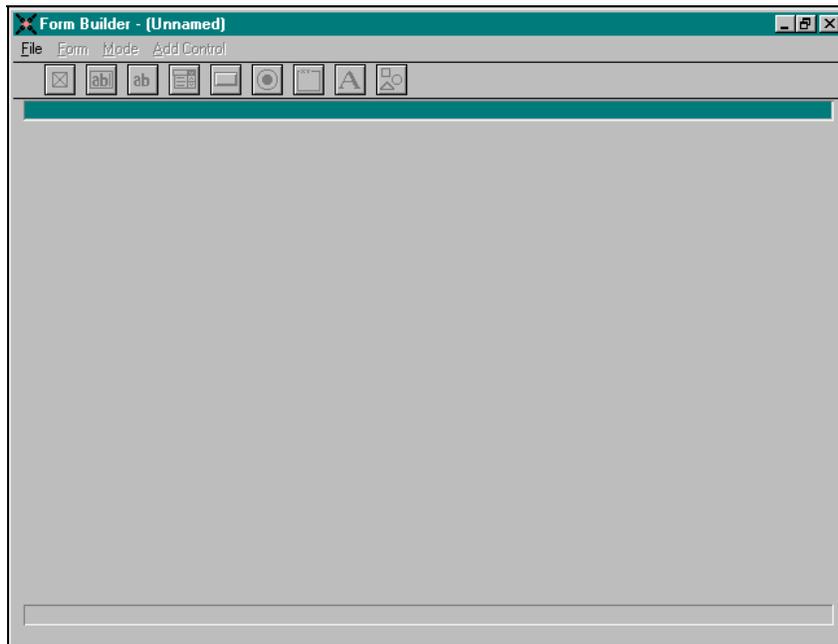
## Designing a new form

### Form design steps

The procedures to design a new form include:

- Starting the form design application  
Form Builder is a Windows application.
- Naming the new form  
The name of the form must be the same for both the form design application and The Solution Series.
- Designing the form  
The three primary activities in designing a form is to create the title, body, and select/inquiry.
- Saving the form  
The form design must be saved after development is complete.

# Form Builder Window



---

## NOTES

---

## Starting the form design application

### Start the form design application

To start the design application, select the Form Builder icon.

### The design application window

The form design application window contains several components. These include:

- Menu Bar
- Workspace
- Button Bar
- Status Bar

### Menu bar

The menu bar displays the form design application options.

- The File menu contains the New, Open, Save, Save As, Exit, and About options.
- The Form menu contains the Sections 1–7 and View Composite options.
- The Mode menu contains the Character, Graphical, Bilingual, Primary Lang, Alternate Lang, and Group Move options.
- The Add Control menu contains the items that can be selected to design a Form.

### Button bar

The button bar displays an Icon for each item that can be selected to design a Form.

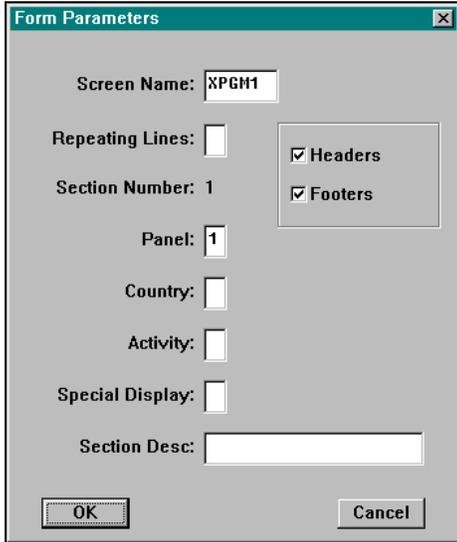
### Workspace

The workspace is where you create or modify the form design. The workspace corresponds to 80 columns (width) and rows 3 through 24 of a 24-row Form (height). One row position is the size of a character.

### Status bar

The status bar displays the type of item selected, its row and column position, the language mode, a list of the sections developed, and the current section number. The list of section numbers used will be grey if the section is not currently displayed.

# Form Parameters Dialog



<u>Value</u>	<u>Action</u>
Blank	Do not repeat this section.
1-8	Repeat the entry and inquiry fields 1-8 times dependent upon the value.
9	Repeat the entry and inquiry fields on all remaining lines (down to line 23). This technique is used in the Select/Inquiry Form section.

---

## NOTES

---

## Naming the new form

Before you can start designing your form, you must create name and form parameters. From the form design application menu select one of the following:

In Form Builder, select the following from the menu:

File ► New

**Result:** The Form Parameters dialog is displayed.

### **Form name**

This text box contains the name of the program used by The Solution Series. The form name must not exceed six characters and should be all uppercase characters.

### **Repeating lines**

This text box determines the number of times this section will be repeated. If you want the selected section to have repeating lines of entry and inquiry fields enter one of the values listed above in this field.

### **Headers**

This check box determines whether the form design application will display inquiry or entry fields that are marked as headers (Section 0).

### **Footers**

This check box determines whether the form design application will display inquiry or entry fields that are marked as footers (Section 9).

### **Section number**

This field contains the form section you are editing.

# Form Parameters Dialog

<b>Country</b>	
<u>Code</u>	<u>Country</u>
Blank	Not Country specific
0	US
1	Canada
2	UK
3	Ireland (Erie)
4	Malaysia
5	Singapore
6	Australia

<b>Special Display</b>	
<u>Value</u>	<u>Description</u>
Blank	Special Display not applicable.
S	Form section requires the special display limits.
R	Form section is a regular display when another Form section is defined as a special display.

---

## NOTES

---

## Naming the new form, continued

### **Batch layout section header**

The four remaining optional fields are used by the Batch Layout Report program (BATCHL).

### **Panel**

This text box is used by the Batch Layout program to determine if this is a multi-panel form. A multi-panel form is one that contains a push button that will produce a second or subsequent form.

### **Country**

This text box determines if a form section is country specific. Leave this field blank if the form section is not country-specific; otherwise enter a valid country code.

### **Activity**

This text box determines if there are different methods for processing the form. Leave this field blank unless a form displays different sections for add and change functions. If this is so, enter A for a section that is used for adding, or C for a section used to make a change.

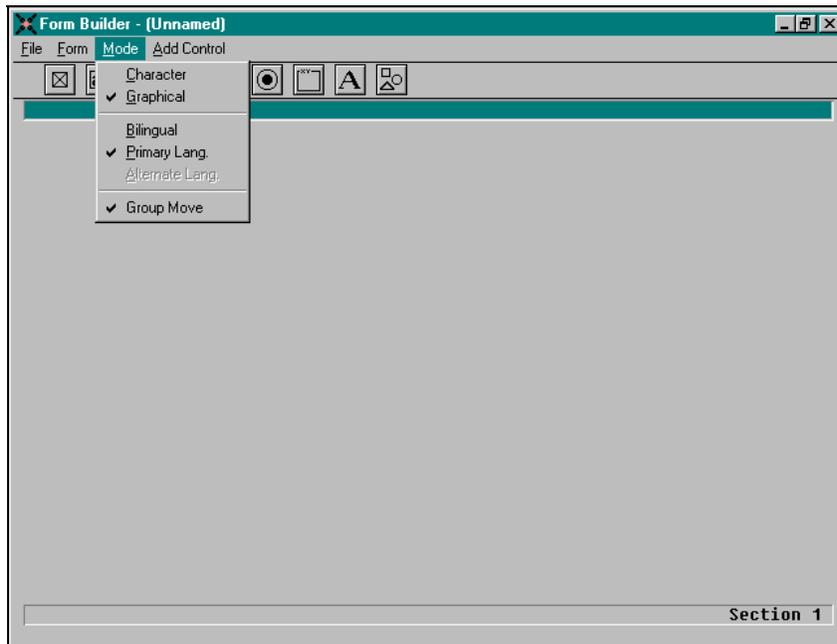
### **Special display**

This text box is used to limit the number of entry fields that display on a character-mode terminal. Some character-mode terminals cannot handle the number of entry fields on some of The Solution Series forms.

### **Section description**

This field is to describe the type of section, in other words, header, body, or select/inquiry.

# Form Builder Modes



---

## NOTES

---

## Establishing modes

### Design application modes

The Form Builder application is used to develop The Solution Series forms by users who interface using a Graphical User Interface (GUI), as well as those using a text-based terminal. Additionally, these form design applications accommodate two different languages per form: the primary language and an alternate language.

Before you start to develop a form, you will want to establish the mode in which you use The Solution Series. In order to establish modes, complete one of the following sets of steps:

In Form Builder, select the following from the menu:

Mode ► Graphical or Character

**Result:** Establishes the Solution Series interface.

Mode ► Bilingual or Primary Lang

**Result:** Establishes the language mode.

Mode ► Group Move

**Result:** Enables or disables the moving of a group boxes and contents.

## Form Design Activities

- **Create the header**
- **Create the body**
- **Create the select/inquiry**

---

### NOTES

---

## Designing the form

### Form design

The three primary activities in designing a form are:

- Creating the header
- Creating the body
- Creating the select/inquiry

### The header

The form header provides a form title and other information regarding the form. If the form is an employee level form, the employee's name is displayed with the form title.

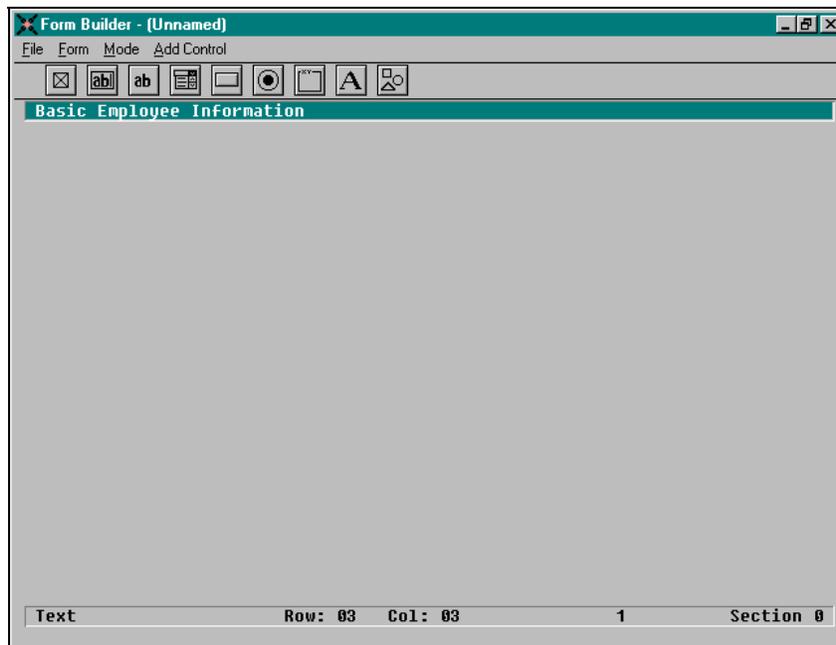
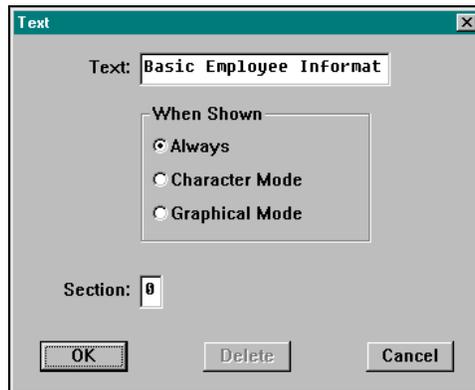
### The body

The form body provides either update or inquiry access to a field. Fields can be represented as entry fields, inquiry fields, radio buttons, check boxes, or list boxes.

### Select/Inquiry

The select/inquiry component provides a summary of each occurrence of multiple occurring segments that may then be selected for full display.

## Form Title—Text Dialog



---

## NOTES

---

## Creating the header

### The form header

As you recall, the form header provides a form title and other form information, such as an employee name. The form header is defined in form section 0.

### The form title

The form title is created using the text dialog box. In order to access the text dialog box perform one of the following:

In Form Builder, click the Add Text button or select the following from the menu:

Add Control ► Text

**Result:** The Text dialog box displays.

### Text field

This field is used to type up to 60 characters to display on the form builder workspace. Cyborg form titles are a maximum of 38 characters, which corresponds with the CSL title in sequence 00000.

### When Shown buttons

The When Shown radio buttons are used to control the display of items based on the mode.

### Section

The Section field identifies a text item as being part of a header, footer, or form body. Since the title is part of the form header, type 0 for the form section.

### Completing the title

To complete the text dialog, fill in each field as needed to describe the form title. Position the title starting in Row 3, Column 3 of the workspace using your pointer.



## Creating the header, continued

### Employee name

In order to include additional information, such as the employee name, complete one of the following:

In Form Builder, click the Add Display Box button or select the following from the menu:

Add Control ► Display Box

**Result:** The Display dialog box is displayed.

### The display dialog

The display dialog contains the attributes of how a field will display on the form. Data in these fields display on the form, but cannot be changed.

### Field name

The field name identifies the data dictionary name to be used with the form item. If you are not sure of the data dictionary name to include in the form item, you may display a list of available field names by typing a question mark (?) and, if possible, the beginning letters of the field name, into the field and choosing OK.

### Label

The Label text box is used to identify the dialog box or inquiry field displayed on the workspace. Though the form design application allows you to enter more than 20 characters in this text box, only the first 20 characters display.

### Exit routines

The exit routines contain 3-digit CSL paragraph numbers. The numbers indicate specific subroutines in the form program that are required to be executed before or after this item is displayed.

### When Shown buttons

The When Shown radio buttons are used to control the display of items based on the mode.

### Label location

The label location determines the placement in relation to the field.

### Section

The section field identifies a text item as being part of a header, footer, or body. Since the title is part of the heading, type a 0 in the section field.

### Completing the name

To complete the dialog, fill in each field as needed to describe the employee name. Position the name starting in row 3, column 49 of the workspace.

# Single Column

The screenshot shows a software window titled "Form Builder - DA-SCR.SAT". The menu bar includes "File", "Form", "Mode", and "Add Control". The toolbar contains icons for various form controls. The main area is titled "Disciplinary Actions" and contains the following fields:

- Date Of Problem: XXXXXXXXXXXX
- Type Of Problem: XXXXXXXXXXXXXXXXXXXXXXXX
- Disciplinary Action:
  - Action Taken: XXXXXXXXXXXXXXXXXXXXXXXX
  - Date Recorded: XXXXXXXXXXXX
  - Effective Date: XXXXXXXXXXXX
- Review Information:
  - Type: XXXXXXXXXXXXXXXXXXXXXXXX
  - Date: XXXXXXXXXXXXXXXXXXXXXXXX

At the bottom, there is a checkbox labeled "Grievances" and a status bar showing "1 9 Section 9".

# Double Column

The screenshot shows a software window titled "Form Builder - 15-SCR.SAT". The menu bar includes "File", "Form", "Mode", and "Add Control". The toolbar contains icons for various form controls. The main area is titled "Emergency Medical Information" and contains the following fields:

- Effective Date: XXXXXXXXXXXX
- Disability: XXXXXXXXXXXXXXXXXXXXXXXX
- Blood Type: XXXXXXXXXXXXXXXXXXXXXXXX
- Last Donation: XXXXXXXXXXXX
- Will Donate Blood
- Employee Smokes
- Allergies:
  - XXXXXXXXXXXXXXXXXXXXX
  - XXXXXXXXXXXXXXXXXXXXX
  - XXXXXXXXXXXXXXXXXXXXX
  - XXXXXXXXXXXXXXXXXXXXX
  - XXXXXXXXXXXXXXXXXXXXX
- Religion: XXXXXXXXXXXXXXXXXXXXXXXX
- Language: XXXXXXXXXXXXXXXXXXXXXXXX
- Emergency Contact/Physician
- Injury/Illness And Work Restriction

## NOTES

## Creating the body

### Form body layout

There are two basic layouts you can use for your form body:

- Single Column
- Double Column

### Key fields

Regardless of the layout you choose in your form design, key fields must adhere to the following rules:

- All key fields must be grouped together in segment layout order.
- Non-key data must follow the key fields.
- All key field labels will end with a greater than symbol (>) denoting that this field is a key field.
- An empty row should exist between the last key field and the first non-key field on the form, when possible.

### Form body items

There are several classifications of fields within The Solution Series which determine what items may be used in creating the form body.

<u>Field Type</u>	<u>Item Type</u>
Date	Entry or Inquiry Field
Numeric	Entry or Inquiry Field
Name	Entry or Inquiry Field
Alphanumeric	Entry or Inquiry Field
Option List	Text Box, Display Box, Option Button, List Box, or Check Box

The following pages discuss each item.

*Note: Before starting to build your form body, you must change the form section to Section 1.*

# Text Box Dialog

Text Box

Field Name:   Spin Button?

Label:

Section:

When Shown

- Always
- Character Mode
- Graphical Mode

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

Exit Routines

Before:

After:

Current Value

- Don't Show
- Show Inside Box
- Show Below Box

---

## NOTES

---

## Creating the body, continued

### Text box dialog

A text box item is defined when the end-user must be able to type information into a text box, such as an employee name or telephone number.

In Form Builder, click the Add Text Box button or select the following from the menu:

Add Control ► Text Box

**Result:** The Text Box dialog is displayed.

### Field name

The field name identifies the data dictionary name to be used with the form item. If you are not sure of the data dictionary name to include in the form item, you may display a list of available fields names by typing a question mark (?) and, if possible, the first characters of the field name, in the field and choosing OK.

### Label

The label text box is used to identify the dialog box displayed on the workspace. Though form design application allows you to enter more than 20 characters in this text box, only the first 20 characters display.

### Section

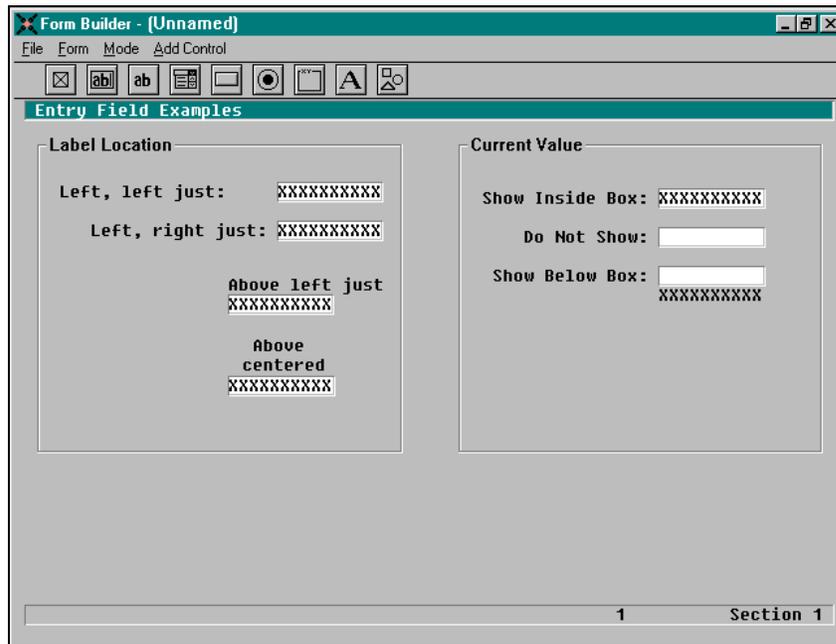
The section field identifies a text item as being part of a header, footer, or body. Type a one (1) in the Section field to define this as the form body.

### Spin button

An up and down arrow is placed next to the text box so the end-user can either type a number directly into the field or use the buttons to cycle from 00 to 99. You may use a spin button only when the field name you enter:

- Is a one- or two-digit numeric field.
- Has no decimal places.
- Is not tied to an Option List.

# Text Box Examples



---

## NOTES

---

## Creating the body, continued

### **When shown**

The When Shown option buttons are used to control the display of items based on the mode.

### **Exit routines**

The exit routines contain 3-digit CSL paragraph numbers. The numbers indicate specific subroutines in the form program that are required to be executed before or after this item is displayed.

### **Label location**

The label location determines the placement in relation to the field.

### **Current value**

The current value determines the display of the field's current value:

- Do Not Show  
Do not display the current value of this field.
- Show Inside Box  
Show the current value of this field in the box.
- Show Below Box  
Show the current value of this field on the line below the box. When the new data is entered, the old value is replaced.

### **Completing the text box**

Complete the text box dialog. Position the field in the workspace using your pointer.

# List Box Dialog

**List Box**

Field Name:

Label:

Section:

Exit Routines

Before:

After:

When Shown

Always

Character Mode

Graphical Mode

Big Codeset

Label Location

No Label

Left, Left Justified

Left, Right Justified

Above, Left Justified

Above, Centered on 2 Lines

---

## NOTES

---

## Creating the body, continued

### List box dialog

You can use a list box in place of a text box for any field that is validated against an option list. Instead of entering the option list value in a text box, the user selects a description from a list.

In Form Builder, click the Add List Box button or select the following from the menu:

Add Control ► List Box

**Result:** The List Box dialog is displayed.

### Field name

The field name identifies the data dictionary name to be used with the form item. If you are not sure of the data dictionary name to include in the form item, you may display a list of available field names by typing a question mark (?) and, if possible, the first characters of the field name, into the field and choosing OK.

### Label

The label text box is used to identify the dialog box displayed on the workspace. Though form design application allows you to enter more than 20 characters in this text box, only the first 20 characters display.

### Section

The section field identifies a text item as being part of a header, footer, or form body. Type a one (1) in the section field to define this as the form body.

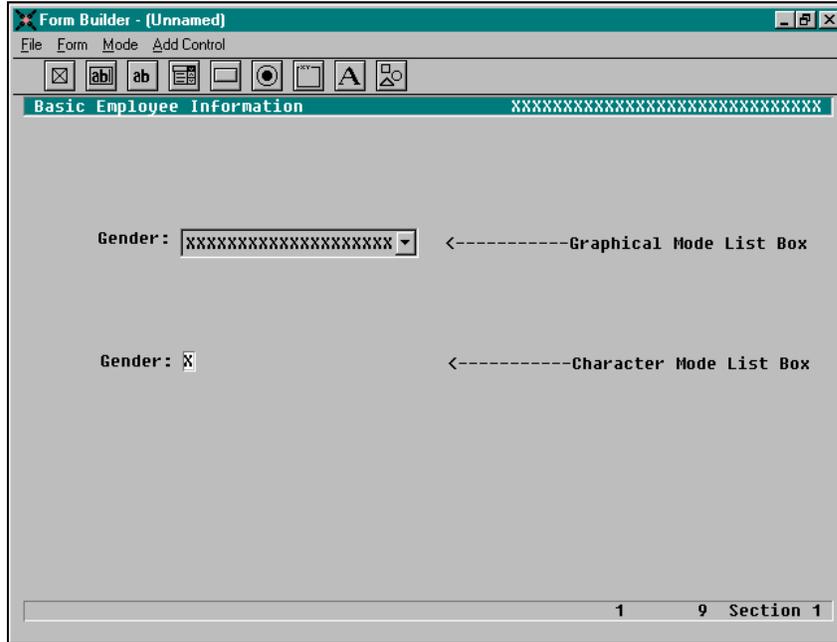
### Big option list

If an option list has more than 40 entries, select the big option list check box. This allows the end-user to more easily search a large option list.

### Label location

The label location determines the placement in relation to the field.

# List Box Example



---

## NOTES

---

## Creating the body, continued

### **When shown**

The When Shown option buttons are used to control the display of items based on the mode.

### **Exit routines**

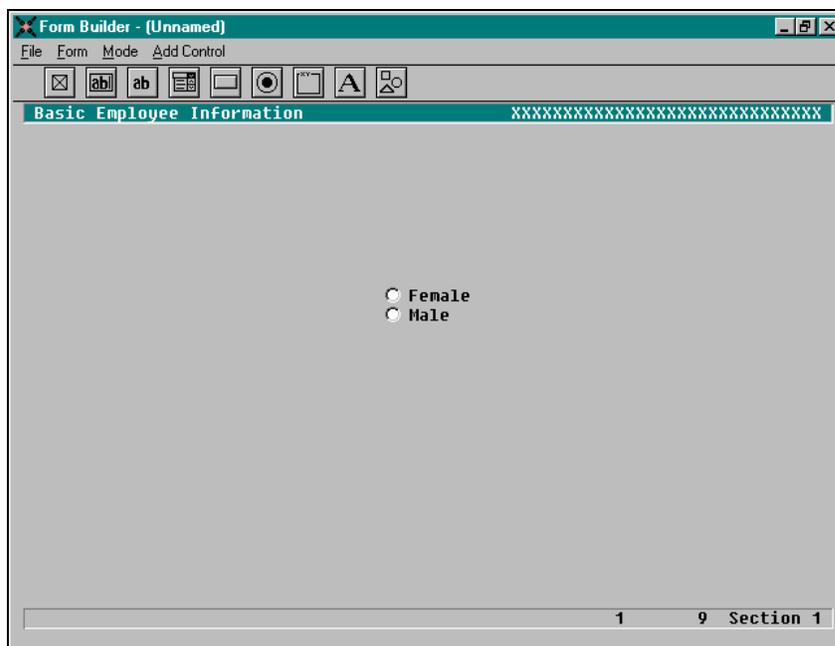
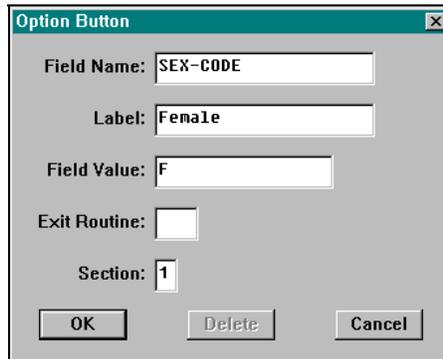
The Exit Routines area is used for CSL paragraph numbers. The numbers indicate specific subroutines that are required to be executed before or after this item is displayed.

### **Completing the list box**

To complete the List Box dialog box, fill in each field as needed to describe the list box. Position the list box in the workspace, using your pointer.

*Note: Character Mode users may use this item, however it converts to a Text Box because List Boxes are not supported in character mode. To display the description next to the entry field, use the field's Option List description name as an inquiry field.*

# Option Button Dialog



---

## NOTES

---

## Creating the body, continued

### Option button

An option button may be used in place of a text box or a list box for any field that is validated against an option list and has a finite set of valid responses (two to five).

In Form Builder, click the Add Option button or select the following from the menu:

Add Control ► Option Button

**Result:** The Option Button dialog is displayed.

### Field name

The field name identifies the data dictionary name to be used with the form item. If you are not sure of the data dictionary name to include in the form item, you may display a list of available field names by typing a question mark (?) and, if possible, the first characters of the name field, in the field and choosing OK.

### Label

The label text box identifies the option list value description represented by the button. Place a question mark (?) in this field to display a list of valid values and descriptions. Selecting a value populates both the label and field value. Labels always display to the right of the button.

### Field value

The Field Value identifies the option list value to be selected by the button.

### Exit routine

The exit routine is for a CSL paragraph number. The number indicates the specific subroutine that is to be executed before this item is displayed.

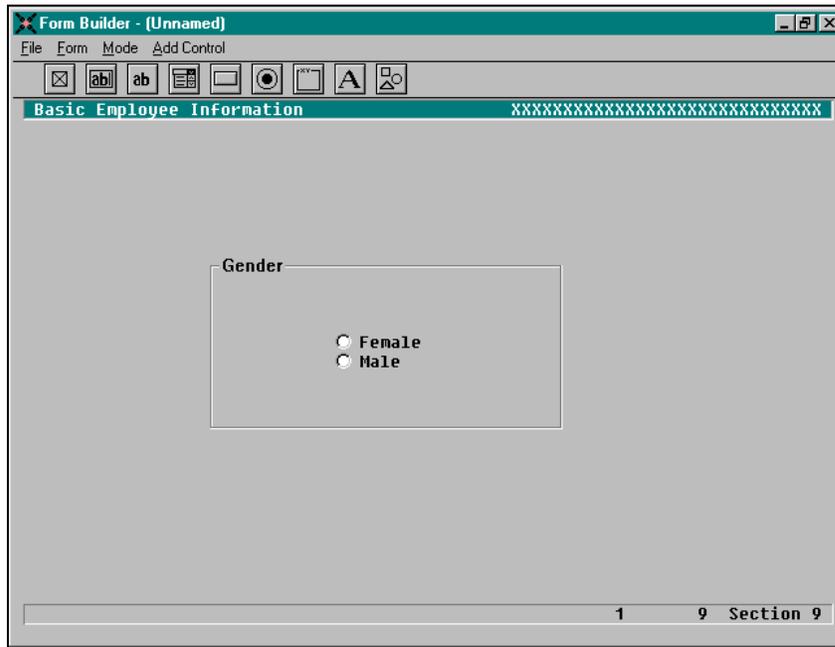
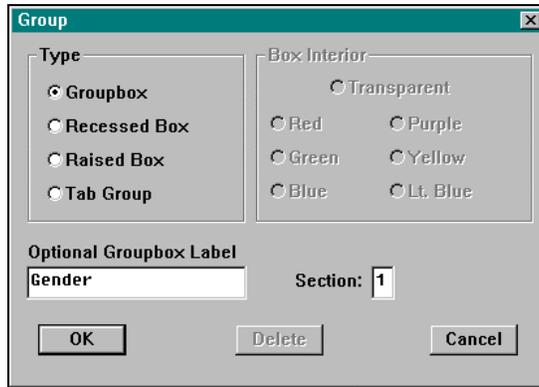
### Section

The section field identifies a text item as being part of a header, footer, or body. Type a one (1) in the section field to define this as the form Body.

### Completing the button

Complete the button dialog. Position the button in the workspace using your pointer.

# Group Dialog



---

## NOTES

---

## Creating the body, continued

### Grouping buttons

When more than one set of buttons display on your form, you will need to group each set to make them independent from one another. In order to group buttons, perform one of the following:

In Form Builder, click the Add Group button or select the following from the menu:

Add Control ► Group

**Result:** The Group dialog is displayed.

### Type

Select one of the four options to determine the formatting of the box:

- Groupbox
- Recessed Box
- Raised Box
- Tab Group

### Optional group box label

This field is used to identify the data grouped within the box. The box label may be up to 20 characters.

### Section

The section field identifies a text item as being part of a header, footer, or form Body. Type a one (1) in the section field to define this as the form body.

### Completing the group box

Complete the dialog box. Position the group box in the workspace using your pointer.

### Size a group box

To change the size of the group box perform the following steps:

1. Select the group box using the pointer.
2. Move the pointer to the edge of the group box so that it becomes a double-headed arrow.
3. Hold the pointer and drag the group box to the appropriate size then release the button.



## Creating the body, continued

### Check Box item

A check box is used when a Yes or No answer is required for a field. Additionally, the field must be validated against an option list. In order to add a check box item, perform one of the following:

In Form Builder, click the Add Check Box button or select the following from the menu:

Add Control ► Check Box

**Result:** The Check Box dialog is displayed.

### Field name

The Field Name identifies the Data Dictionary name to be used with the Form item. If you are not sure of the Data Dictionary name to include in the Form item, you may display a list of available field names by typing a question mark (?) and, if possible, the first character of the field name, into the field and choosing OK.

### Label

The Label text box is used to identify the Dialog Box displayed on the workspace. Though Form Builder allows you to enter more than 20 characters in this text box, only the first 20 characters display. For check boxes, label text always displays to the right of the check box.

### Exit routine

The Exit Routine is for a CSL paragraph number. The number indicates the specific subroutine in the Form's program that is to be executed before this item is displayed.

### Section

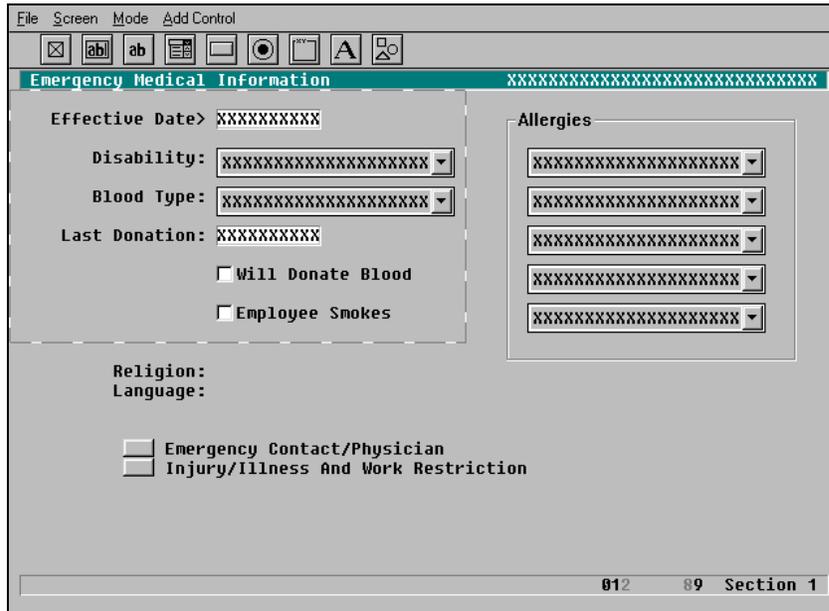
The Section field identifies a text item as being part of a Header, Footer, or Form Body. Type a one (1) in the Section field to define this as the Form body.

### Completing the check box

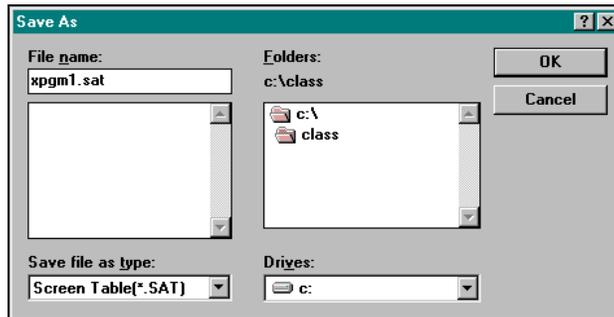
Complete the check box. Position the check box in the workspace using your pointer.

*Note: Check boxes can only be associated with fields that have Option Lists HR00, PP00, SC02 (Yes, No).*

# TAB Key Sequence



# Save Dialog



## NOTES

## Tab sequence

### Tabbing

When you draw any type of box around a group of form elements, all items within the group box are accessed before any items outside of it. Cursor movement via TAB is always left to right, then top to bottom through a form. When a group box is encountered, the system moves through all entries in the box before continuing with the next item on the form.

### Grouping

Use the group dialog box to define groups of items for tabbing and add graphical elements to your form. There are three distinct items you can add using the group dialog box:

- Group box
- Recessed box
- Raised box
- Tab group

### Moving a group of items

To move a group box and its contents perform the following:

In Form Builder, click the Add Check Box button or select the following from the menu:

Mode ► Group Move

**Result:** The Group Move option is selected.

Once the Group Move option is selected, click the box and drag the group to its new position or use the arrow keys to reposition it.

### Saving the design

The final step in designing the form is to save the design to a file. Make sure that your form design file has an extension of .SAT, and remember to save it in all capitals.

## Section 2 Exercise 1

RECORD-DATE	This is a key field that contains the effective date of the form's information.
MARITAL-CODE	This code identifies the employee's current marital status. The description for this field is MARITAL-STATUS.
CITIZENSHIP-CODE	This code contains the country where the employee holds citizenship. The description for this field is CITIZEN-COUNTRY.
TOTAL-DEPENDENTS	This is a 2-digit field that records the number of the employee's dependents.
PRIOR-NAME	This field contains an employee or applicant prior surname due to marriage or other legal means.
ID-VERIFIED	This field contains the identification procedure status. The status denotes whether or not the procedure has started and if identification could be produced. The description for this field is ID-VERIFIEDS.
ID-PROVIDED	This field contains the code for the type of identification document produced. The description for this field is ID-PROVIDEDS.

---

### NOTES

---

## Section 2 exercise 1

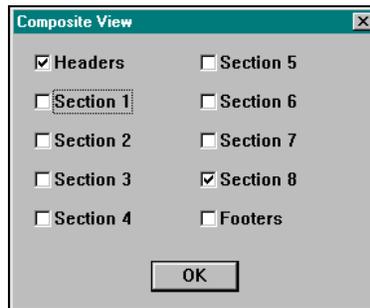
### Purpose

This exercise gives you practice using the information you have learned in the section.

Design a form that displays the fields listed below and allows them to be updated in both graphical and character modes.

- Use the title Personal and Identification Information.
- Display the employee name in the form heading.
- Display the following fields and allow them to be updated:
  - RECORD-DATE
  - MARITAL-CODE
  - CITIZENSHIP-CODE
  - TOTAL-DEPENDENTS
  - PRIOR-NAME
  - ID-VERIFIED
  - ID-PROVIDED

# Composite View Dialog



---

## NOTES

---

## Select/Inquiry form section

### Change the composite view

Once you have completed the form parameters for Section 8, you will notice that the form display does not change. This is due to a feature known as Composite View, which allows you to edit multiple sections of a form at a time.

For Form Section 8, we will be creating a Select/Inquiry version of the form, and will need to change the composite view to edit only Section 8. In order to change the composite view, perform one of the following:

In Form Builder, select the following from the menu:

Form ► Composite View

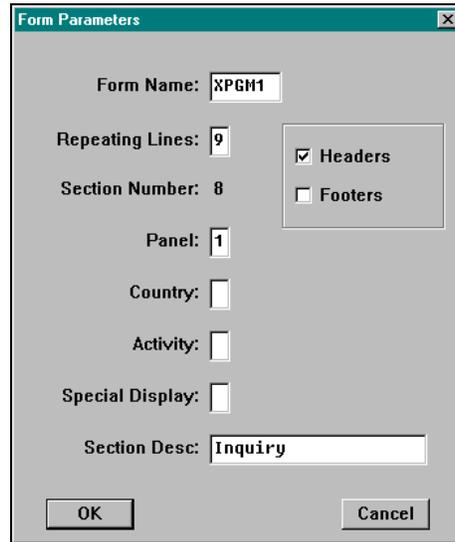
**Result:** The Composite View dialog is displayed.

In order for the workspace to display only Section 8 and the Headers, perform the following from the Composite View dialog:

1. Deselect all sections other than Headers and Section 8.
2. Choose OK or press Enter.

*Note: The Composite View Feature allows you to view and edit items that have already been created. Any items that are added when viewing multiple sections will be added to the section that appears in the status bar.*

## Select/Inquiry—Section 8



The image shows a 'Form Parameters' dialog box with the following fields and options:

- Form Name:
- Repeating Lines:
- Section Number:
- Panel:
- Country:
- Activity:
- Special Display:
- Section Desc:
- Headers:
- Footers:
- OK button
- Cancel button

---

### NOTES

---

## Select/Inquiry form section, continued

### Select/Inquiry section

As you recall, there are two display modes for a Form: Entry mode and Select/Inquiry mode. This topic focuses on the creation of the Select/Inquiry section.

### Section 8

In order to edit form Section 8, perform one of the following:

In Form Builder, select the following from the menu:

Form ► Section 8

**Result:** The Form Parameters dialog is displayed.

### Form name

This text box contains the name of the program you entered when you named the form. Do not change this field.

### Repeating lines

This field determines the number of times this section is repeated. Type '9' to repeat the fields on all remaining lines (down to line 23).

### Headers

This check box determines whether the form design application will display fields that are marked as headers. Select this button if it is not already selected.

### Footers

This check box determines whether the form design application will display inquiry or entry fields that are marked as footers. Deselect this button if it is already selected.

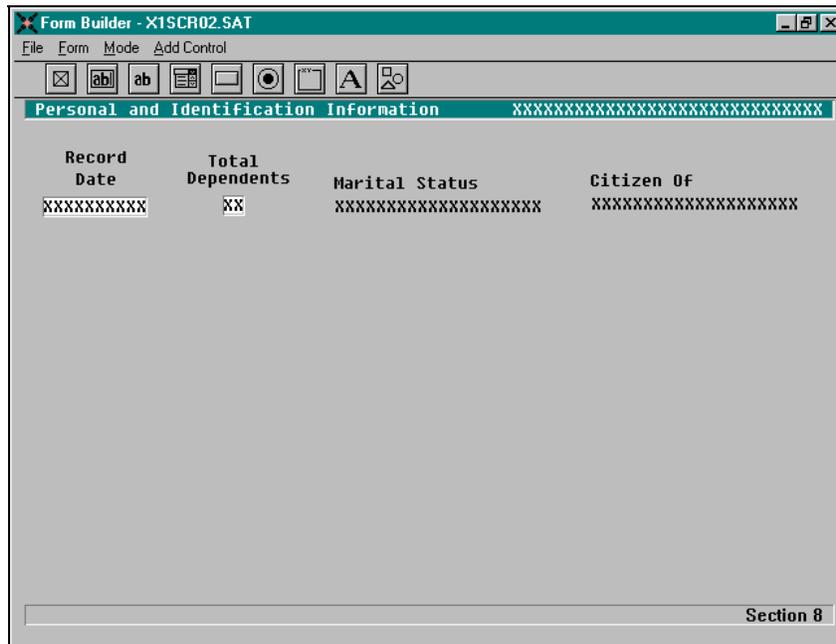
### Section number

This field contains the form section you are editing. It must contain an 8.

### Batch layout section header

The remaining fields are used by the Batch Layout Report (BATCHL) program. Refer to the Naming the Form topic for details.

## Section 8—Select/Inquiry Form



---

### NOTES

---

## Select/Inquiry Form Section, continued

### Format rules

The format of the Select/Inquiry Form section must contain the following format:

- Headings appear on two lines above the text box with the data displayed inside the unprotected areas.
- All key fields must appear in order from left to right as entry fields.
- One non-key field must appear following the key fields as an entry field.
- All remaining non-key fields that will fit on one line will appear as inquiry fields.
- It is a requirement that labels are painted on lines 5/6 with the data fields falling on line 7.

## Section 2 Exercise 2

RECORD-DATE	This is a key field that contains the effective date of the form's information.
MARITAL-CODE	This code identifies the employee's current marital status. The description for this field is MARITAL-STATUS.
CITIZENSHIP-CODE	This code contains the country where the employee holds citizenship. The description for this field is CITIZEN-COUNTRY.
TOTAL-DEPENDENTS	This is a two-digit field that records the number of the employee's dependents.
PRIOR-NAME	This field contains an employee or applicant prior surname due to marriage or other legal means.
ID-VERIFIED	This field contains the identification procedure status. The status denotes whether the procedure has started, and if identification could be produced. The description for this field is ID-VERIFIEDS.
ID-PROVIDED	This field contains the code for the type of identification document produced. The description for this field is ID-PROVIDEDS.

---

### NOTES

---

## Section 2 exercise 2

### **Purpose**

This exercise gives you practice using the information you have learned in the section.

Modify the form you created in Section 2, Exercise 1 to include the following:

- Add a tab group to the Form (Graphical Mode only).
- Create a Select/Inquiry section of the Form using Form Section 8.

## Section Summary

- **The form design application and The Solution Series**
- **Designing the form**
- **Creating the header**
- **Creating the body**
- **Saving the form**
- **Select/Inquiry form section**

---

NOTES

---

## Section summary

In this section, you learned to use Form Builder. Specifically, you learned:

### **The form design application and The Solution Series**

---

---

### **Designing the form**

---

---

### **Creating the form header**

---

---

### **Creating the form body**

---

---

### **Saving the form**

---

---

### **Select/Inquiry form section**

---

---

---

**NOTES**

---

## Section 3: Customization Basics

---

### Table of Contents

Overview.....65

The form design application and CSL .....67

Form programming .....69

Context-sensitive menu records.....79

Extracting the form .....81

Generating the form .....83

Extracting the form in batch .....85

Generating the form in batch .....87

Section summary.....89

Section 3 exercise .....91

## Objectives

- **Identify the relationship between the form design application and Cyborg Scripting Language (CSL)**
- **Identify and use programming verbs**
- **Create context-sensitive menu records**
- **Create and run a form program**

---

## NOTES

---

## Overview

### **Purpose**

In this section, you will learn the technique of creating a form program.

### **Objectives**

When you complete this section, you will be able to:

- Identify the relationship between the form design application and Cyborg Scripting Language (CSL).
- Identify and use programming verbs.
- Create context-sensitive menu records.
- Create and run a form program.

# The Form Design Application and CSL

## The Form Design Application

- Form Layout
- Field Presentation
- Form Navigation

## CSL

- Database Access
- Form Processing
- Error Processing

---

## NOTES

---

## The form design application and CSL

### The relationship between the application and CSL

As you recall, Form Builder is the tool for designing the look and feel of Solution Series forms. However, CSL programming is required to complete the development process of a form.

- The form design application provides the capabilities of:
  - Form layout
  - Field presentation
  - Form navigation
  
- CSL provides the capabilities of:
  - Database access
  - Form processing
  - Error processing

## **Form Programming Structure**

- **Record access statement**
- **Form style statement**
- **Form title statement**
- **Data entry line statements**
- **Data verification statements**

---

### **NOTES**

---

## Form programming

### Program structure

Forms are programmed using a structure common to all form programs. The program structure should include the following components:

- Record access statements
- Form styles statement
- Form title statement
- Data entry line statements
- Data verification statements

As you can see, this structure parallels the form components. Additionally, the form program must be coded so that each of the above sets of statements is executed one after the other.

# Record Access

## Format

UPDATE-COMPANY.

UPDATE-EMPLOYEE.

UPDATE-TAXES.

### Record Access:

```
P100-START-SCREEN.  
KEY-REQUIRED.  
UPDATE-EMPLOYEE.  
NEW-SCREEN-STYLE.  
SCREEN-SECTION '0'.  
IF INQUIRY-MODE OR SELECTION-MODE  
GO TO P990-INQUIRY-SCREEN.  
P200-ENTRY-SCREEN.  
SCREEN-SECTION '1'.  
P300-VERIFY.  
SET-FOR-MESSAGES.  
IF ERRORS-EXIST RETURN.  
IF RECORD-NOT-UPDATED  
RETURN.  
CALL 'XXXXXX'. @ Edit Routine  
RETURN.  
P990-INQUIRY-SCREEN.  
SCREEN-SECTION '8'.  
RETURN.
```

---

## NOTES

---

## Form programming, continued

### Record access

As you recall, the READ- verbs allow you to access records on the Employee Database for inquiry use only. Therefore, UPDATE- verbs are provided to access the employee, company and tax records for update. The UPDATE- verbs:

- Access the entire Employee Database record into working storage.
- Must be the first commands coded in a form. The verbs used depend on the type of data being accessed.
- More than one access verb can be used in a program.
- Lock the record while the record is read into memory or rewritten to the file.

There are separate verbs for accessing employee, company and tax records. They are:

- UPDATE-COMPANY  
Reads the company record data into working storage and allows updating of the data. UPDATE-COMPANY uses the data in the CONTROL-1-2 field as the key to the record.
- UPDATE-EMPLOYEE  
Reads the employee record data into working storage and allows updating of the data. UPDATE-EMPLOYEE uses the data in the CONTROL-1-2 field and KEY field as the key to the record.
- UPDATE-TAXES  
Reads the tax record into working storage and allows updating of the data. UPDATE-TAXES uses the data in the CONTROL-1-2 field and KEY field as the key to the record.

# Form Style and Title

## Format

*NEW – SCREEN – STYLE.*

<i>SCREEN – SECTION</i> { <i>fieldname</i> } <i>literal</i> }
--

Record Access:

**Form Style:**

**Form Title:**

```
P100-START-SCREEN.  
KEY-REQUIRED.  
UPDATE-EMPLOYEE.  
NEW-SCREEN-STYLE.  
SCREEN-SECTION '0'.  
IF INQUIRY-MODE OR SELECTION-MODE  
GO TO P990-INQUIRY-SCREEN.  
P200-ENTRY-SCREEN.  
SCREEN-SECTION '1'.  
P300-VERIFY.  
SET-FOR-MESSAGES.  
IF ERRORS-EXIST RETURN.  
IF RECORD-NOT-UPDATED  
RETURN.  
CALL 'XXXXXX'. @ Edit Routine  
RETURN.  
P990-INQUIRY-SCREEN.  
SCREEN-SECTION '8'.  
RETURN.
```

---

## NOTES

---

## Form programming, continued

### **NEW-SCREEN-STYLE**

The NEW-SCREEN-STYLE verb must be used in all forms that were developed using the form design application. The verb signals the CBSV COBOL logic that a non-key field may have data within the entry box.

Additionally, this verb tests for selection or inquiry mode. If either of these modes is encountered, it prevents radio buttons, check boxes, drop down list boxes, and other controls from being displayed. This is necessary because all of these controls can be created only in conjunction with entry fields.

This command **MUST** be on its own line. No other code can appear before or after this verb on the same line.

This command cannot be used within an IF statement.

### **Form title**

The form section 0 contains the form title information and is displayed to the form using the SCREEN-SECTION verb. This is a reserved section.

Form Section 0's content will contain any text or display boxes that were defined as a header.

### **SCREEN-SECTION**

The SCREEN-SECTION verb is used to display a specific section of a form that was created using the form design application:

- This command **MUST** be on its own line. No other code can appear before or after this verb on the same line.
- This command cannot be used within an IF statement.

# Data Lines

## Form Sections 1–8

Record Access:	P100-START-SCREEN. KEY-REQUIRED. UPDATE-EMPLOYEE. NEW-SCREEN-STYLE. SCREEN-SECTION '0'. <b>IF INQUIRY-MODE OR SELECTION-MODE</b> <b>GO TO P990-INQUIRY-SCREEN.</b>
Form Style:	<b>P200-ENTRY-SCREEN.</b>
Form Title:	<b>SCREEN-SECTION '1'.</b>
<b>Form Section 8:</b>	P300-VERIFY. SET-FOR-MESSAGES. IF ERRORS-EXIST RETURN. IF RECORD-NOT-UPDATED RETURN. CALL 'XXXXXX'. @ Edit Routine RETURN.
<b>Form Sections 1-7</b>	<b>P990-INQUIRY-SCREEN.</b> <b>SCREEN-SECTION '8'.</b> <b>RETURN.</b>

---

## NOTES

---

## Form programming, continued

### Data entry lines

The coding for the data entry line(s) is dependent upon the number of form sections you have created.

- Form sections 1 through 7 are used to create the different areas of images that will appear on the form. These include entry, inquiry, and option list fields.

### Data inquiry lines

The coding of the data inquiry line is conditionally dependent upon whether the form has been placed into inquiry or selection mode.

- Form Section 8 is used to create the inquiry version of the form. This is a reserved section.
- The Boolean field INQUIRY–MODE can be checked to determine if the form is in inquiry mode.
- The Boolean field SELECTION–MODE can be checked to determine if the form is in select mode.

### SCREEN–SECTION

Remember, the SCREEN–SECTION verb:

- Must be on its own line. No other code can appear before or after this verb on the same line.
- Cannot be used within an IF statement.

# Data Verification and Form Prompts

```
Record Access:      P100-START-SCREEN.  
                   KEY-REQUIRED.  
Form Style:        UPDATE-EMPLOYEE.  
Form Title:       NEW-SCREEN-STYLE.  
Form Section 8:   SCREEN-SECTION '0'.  
                   IF INQUIRY-MODE OR SELECTION-MODE  
                   GO TO P990-INQUIRY-SCREEN.  
Form Sections:1-7 P200-ENTRY-SCREEN.  
                   SCREEN-SECTION '1'.  
Verification:    P300-VERIFY.  
                   SET-FOR-MESSAGES.  
                   IF ERRORS-EXIST RETURN.  
                   IF RECORD-NOT-UPDATED  
                   RETURN.  
                   CALL 'XXXXXX'.    @ Edit Routine  
Form Section 8:   RETURN.  
                   P990-INQUIRY-SCREEN.  
                   SCREEN-SECTION '8'.  
                   RETURN.
```

---

## NOTES

---

## Form programming, continued

### Data verification

The next group of statements included in our form program is the data verification statements. The minimum statements to include for form verification insure that error messages appear at the bottom of the form and make sure that system-detected errors are shown.

- SET-FOR-MESSAGES insures that any messages will be placed at the bottom of the form.
- The ERRORS-EXIST Boolean checks the SCREEN-ERROR field for an @. If there are no errors, SCREEN-ERROR contains an F.
- The RECORD-NOT-UPDATED Boolean checks the value of the RECORD-UPDATED field for a blank. This field is initially set to a blank when a form is painted. If any data entry is detected it is set to Y.
- Additional relational field edits may be placed in a called subroutine or coded directly into this part of the program.

# Context-sensitive Menu Records

COMMAND:

Screen/Sq	ScrNam	GrLnKy	Menu	Screen Title
40-SCR 01	41-SCR		N	Salary Information
40-SCR 02	49-SCR		N	Performance Appraisal Results
41-SCR 01	40-SCR		N	Salary Assignments/Changes
41-SCR 02	42-SCR		N	Projected Salary Changes
42-SCR 01	40-SCR		N	Salary Assignments/Changes
93BSCR 01	93XSCR		N	Absences
93BSCR 02	93TSCR		N	Absence Inquiry By Type
93BSCR 03	93PSCR		N	Absence Points
93XSCR 01	93BSCR		N	Absence Entitlement
93XSCR 02	93TSCR		N	Absence Inquiry By Type
93XSCR 03	93PSCR		N	Absence Points
EF-SCR 01	GG-SCR		N	Home Location Pay Allocation
JJ-SCR 01	GG-SCR		N	Payroll Home Location/Pay Allo
JJ-SCR 02	JR-SCR		N	Reciprocal Tax Setup
M10SCR 01	M10SCR		Y	Job Set Up
M10SCR 02		L		
M10SCR 04	M13SCR		Y	Job Skills Required
M10SCR 05	M15SCR		Y	Job Education Required
M10SCR 06	M1TSCR		Y	Job Training Required

Salary Assignment/Changes      AUSTIN, STEVEN

Effective Date: 01-01-2000  
 Key Separator: 1st Occurrence  
 Type Of Change: Increase-Merit  
 Pay Frequency: Weekly

Process Override  
 Override Calculation  
 History-No PR Update  
 Not 1st Chg for Date  
 No Override(Default)

Calculation Entries	Calculated Results
Hours Per Period: <input type="text"/>	30.00
Hourly Rate: <input type="text"/>	9.6153
Salary Per Period: <input type="text"/>	288.46
Annual Salary: <input type="text"/>	15,000.00
Amount Change: <input type="text"/>	6,073.16
Percent Change: <input type="text"/>	68.03
Months Since Prior: <input type="text"/>	181

Save This Form  
 Cancel This Form  
 Select An Employee...  
 Show Selection  
 Salary Information  
 Performance Appraisal Results

History

## NOTES

## Context-sensitive menu records

Context-sensitive menus provide the user with the ability to navigate to related forms directly without using the application Menus or Navigator. Context Sensitive Menu (ECM) records are stored on the system control repository (FILE01).

### Editing ECM records

The Edit Utility form (EDIT) is used to create, update, and delete context-sensitive menu records.

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** System Control Repository Utilities  
**Task:**  Edit Control Repository Object

**Result:** The Edit Utility form (EDIT) is displayed.

<b>Object:</b>	<b>Context Menus</b>	Edit ECM Records
<b>Object Key:</b>	xxxSCR	Form name

### Edit Utility form (EDIT) format

**(Unlabeled)** The left column is for the line commands of Add, Change or Delete.

**Screen/Sq** Form ID of the host form followed by 1 blank and a 2-digit sequence number to keep all the records for the same form unique. This is the key for the ECM record.

**ScrNam** The form ID to be included in the context-sensitive menu.

**Gr** Group code. (Not used at this time. Leave blank.)

**Ln** An 'L' will insert a horizontal line at this point in the menu. Used to group related forms in the menu list. The rest of the line will be blank if inserting a line with 'L'.

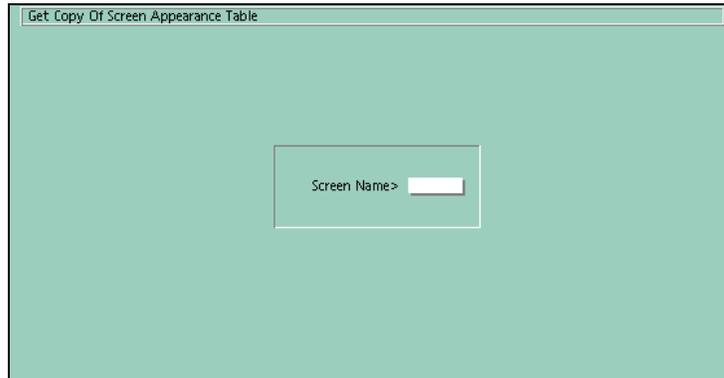
**Ky** An 'N' will retain the current key and additional key information (employee number, and so forth).  
 A 'Y' or blank will result in a prompt for the key information prior to the form's display.

**Menu Screen Title** The form title to be used in the shortcut menu.

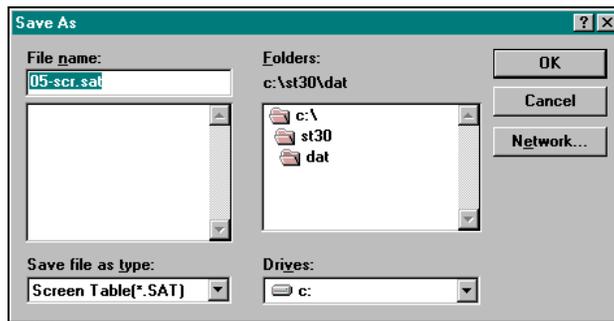
### Access

The context-sensitive menu is in the lower portion of the Form's shortcut menu accessed by right clicking on a blank part of the form.

# Extracting the Form



# Save As Dialog



---

## NOTES

---

## Extracting the form

### Extract the form

In order to complete development of a form program, you must first extract it using the Get Copy of Screen Appearance Table program (GETSAT).

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Programming Utilities
- Task:**  Extract Form Appearance Table

**Result:** The Get Copy Of Screen Appearance Table (GETSAT) form displays.

After you have accessed the GETSAT program, complete the following steps:

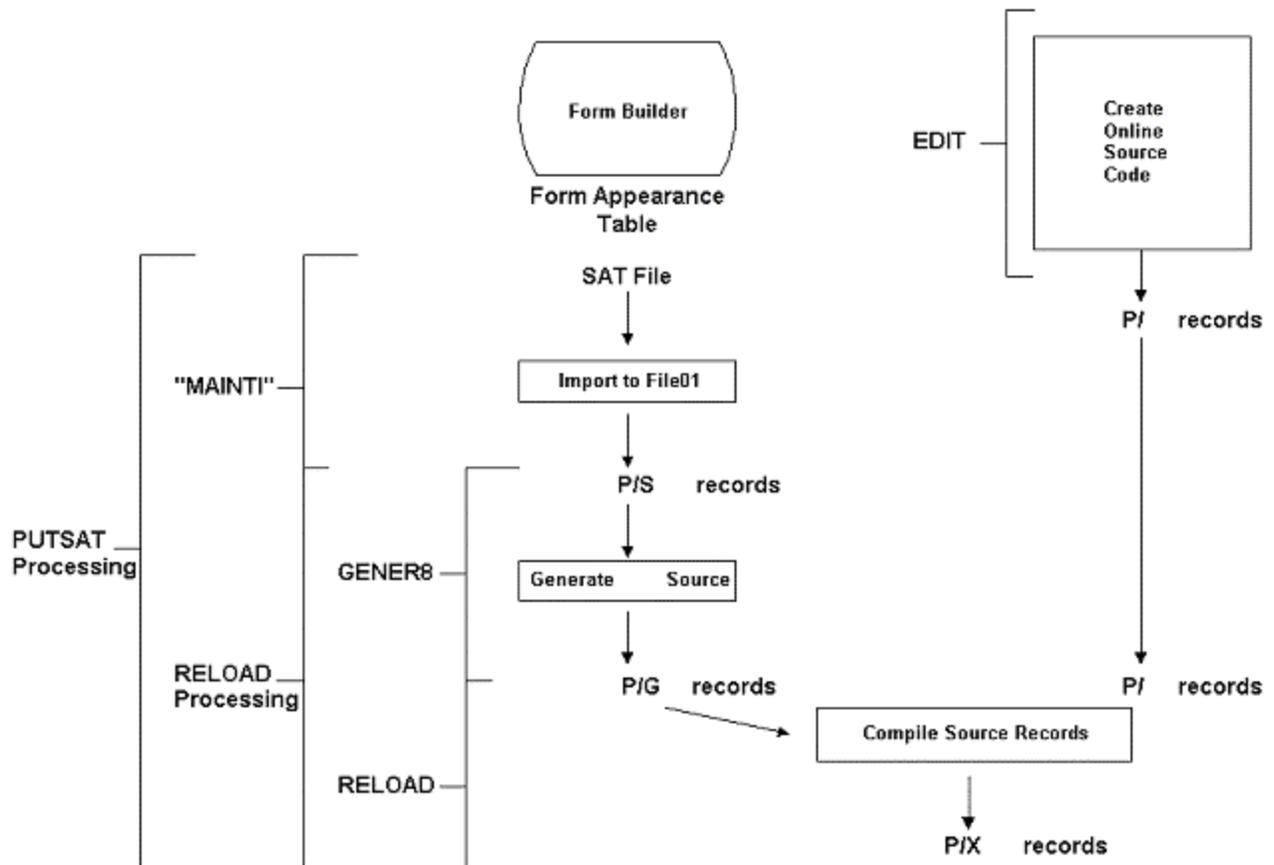
1. Enter the form ID of the form and click OK or press Enter.

**Result:** The Save As dialog box displays.

2. Enter the name of the Screen Appearance File (.sat) and click OK or press enter to save it.

**Result:** The Screen Appearance File will be extracted from The Solution Series.

# Generating the Form



---

## NOTES

---

## Generating the form

### Generate the form

To complete the development of the form program, you must combine the form appearance data with your CSL program. This task is completed using the Update Form Appearance Table program (PUTSAT):

Make the following selections from the Navigator:

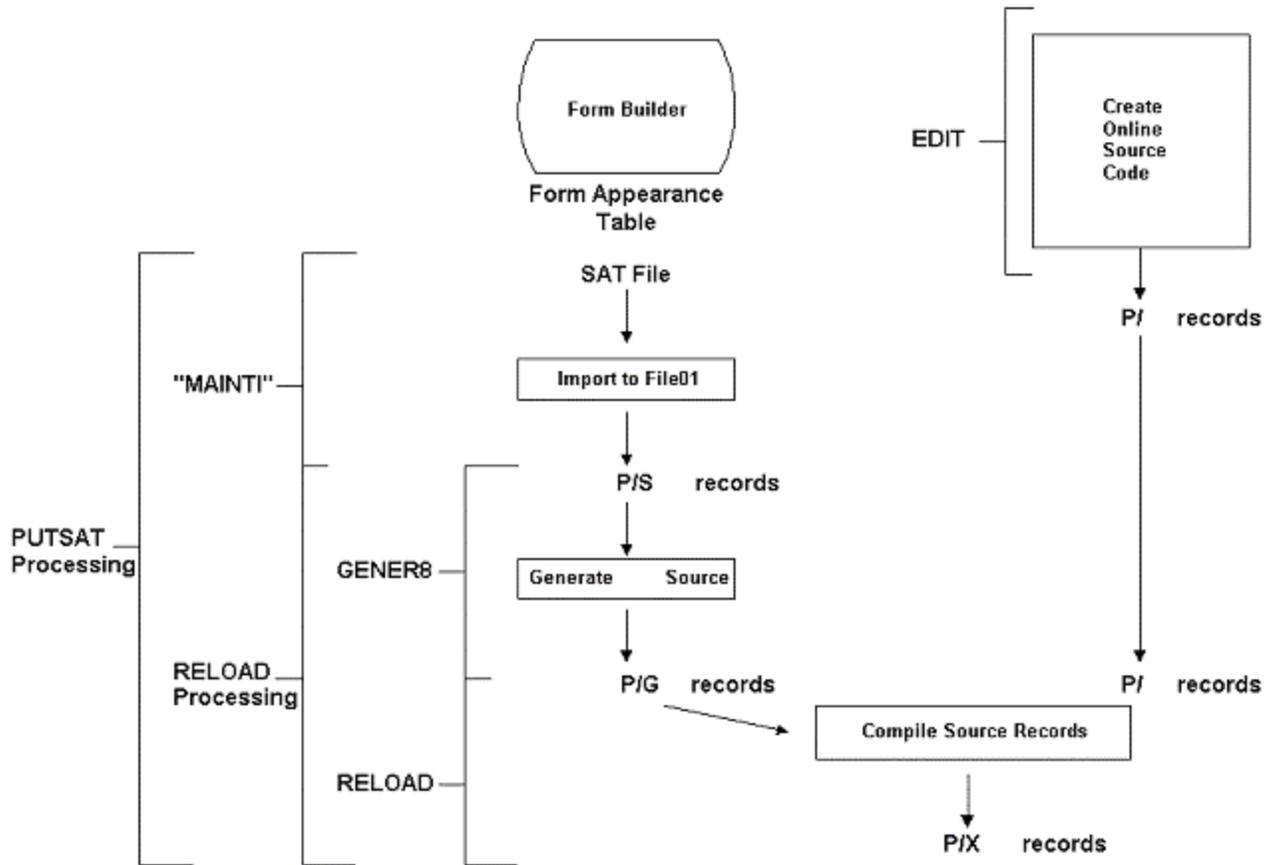
- Component:**  Development Tools
- Process:** Programming Utilities
- Task:**  Update Form Appearance Table

**Result:** The Open dialog box is displayed.

Use the Files list box to locate and select the Form Appearance file (.SAT extension), then choose OK or press ENTER.

**Result:** The Form Appearance File is imported into The Solution Series. The Form Appearance logic is generated and the data entry form program is compiled. When the compile is complete, the RELOAD IS OK message appears.

# Generating the Form



---

## NOTES

---

## Extracting the form in batch

### Batch extraction

The GETSAT program extracts the Screen Appearance File from the System Control Repository (Control File; FILE01).

1. Execute the GETSAT utility in batch as follows:

INPUT	FILE01 FILE04	System Control Repository Control Record File
OUTPUT	FILE03 FILE10	Audit/Message File Screen Appearance Table (*.SAT)
EXECUTE	CBSVB	

The control record on FILE04 has the following syntax:

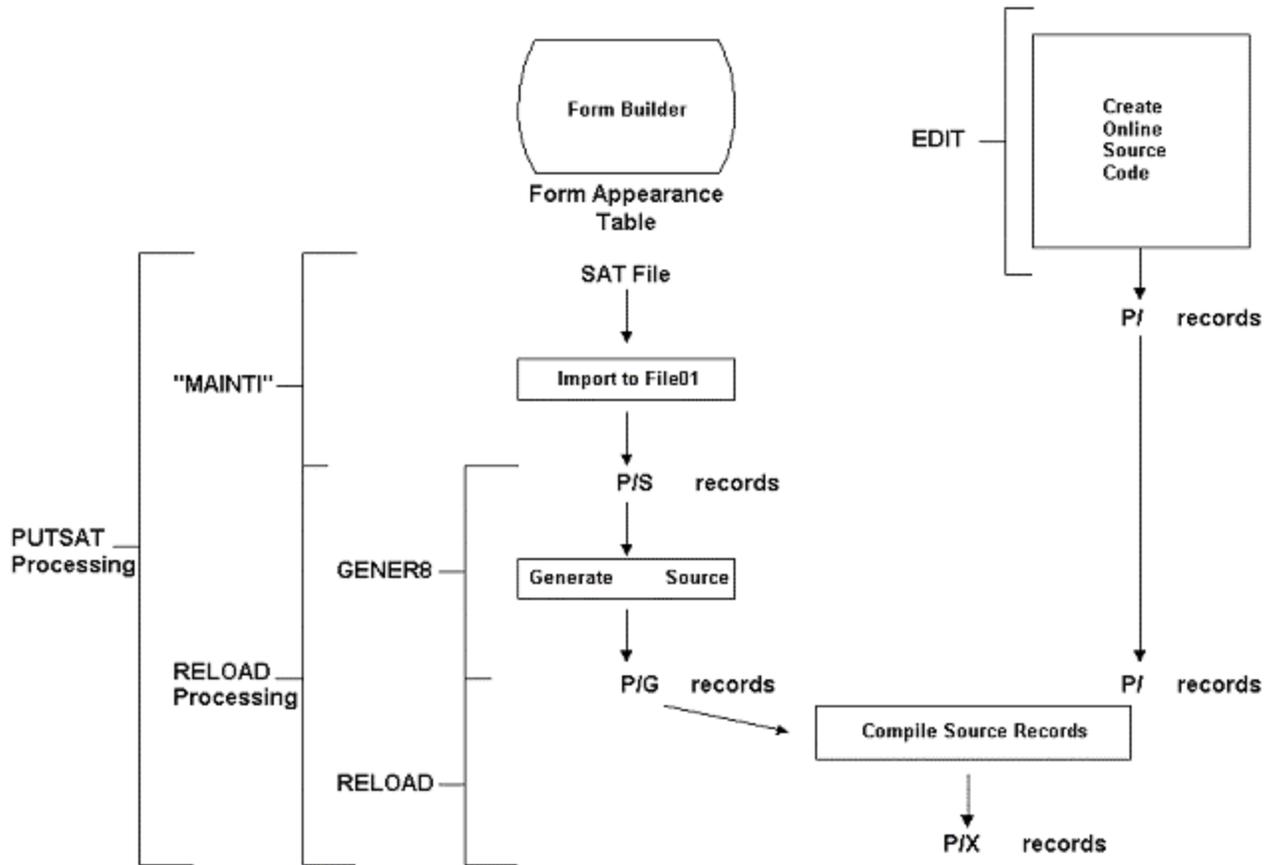
Positions	Entry	Description
23-28	GETSAT	Program name
31-40	Form program name (for example, '05-SCR')	Form program name

Control record example:

1	2	3	4	5
1...5...0...5...0...5...0...5...0...5...0...5		GETSAT	05-SCR	

2. Launch form design application
3. Revise the screen display
4. Save the revised Screen Appearance Table
5. Return to The Solution Series
6. Generate the form (\*.SAT file) in batch

# Generating the Form



---

## NOTES

---

## Generating the form in batch

### RELOAD processing

Execute the Update Form Appearance Table program (PUTSAT) in batch. The PUTSAT program updates the Screen Appearance Table file, runs the GENER8 program to create the screen appearance logic, and runs the RELOAD program to compile the screen code. Type the name of your form program in the control statement KEY field and use FILE05 to input the .SAT file.

If you make changes to either the CSL source code records (P/) or Form Appearance Table records (P/S) using the online Edit Utility (EDIT), the RELOAD compiler must be executed. The RELOAD processing includes the generation of CSL source code records (P/G) from the form appearance table records and the creation of the executable program records (P/X).

Execute this utility in batch as follows:

INPUT	FILE04 FILE05	Control Record File Screen Appearance Table (*.SAT)
OUTPUT	FILE03	Audit/Message File
EXECUTE	CBSVB	

The control record on FILE04 has the following syntax:

In these positions	Enter	Description
23-28	PUTSAT	Program name

Control record example:

1	2	3	4	5
1 . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 5	PUTSAT			

Access the form in The Solution Series and review the layout to make sure it is acceptable and conforms to the specification.

## Section Summary

- **Form components**
- **Form programming**
- **Create context-sensitive menu records**
- **Generating the form**

---

NOTES

---

## Section summary

In this section, you learned the basic techniques for creating an online form program. Specifically, you learned:

### Form components

---

---

---

### Form programming

---

---

---

### Create context-sensitive menu records

---

---

---

### Generating the form

---

---

---

## Section 3 Exercise

---

**NOTES**

---

## Section 3 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. Create a form program that displays the form you designed using Form Builder.
2. Create context-sensitive menu records for your form. Include the following forms:
  - 03-SCR—Telephone Information
  - 24-SCR—Automobile Information
  - 16-SCR—Emergency Contact/Physician
3. Generate the form program using the Update Form Appearance Table program (PUTSAT).
4. Run and test the form program.

---

**NOTES**

---

## Section 4: Defining Fields

---

### Table of Contents

Overview.....	95
The Solution Series field reports .....	97
Creating option lists .....	99
Field definitions .....	101
Section summary.....	127
Section 4 exercise .....	129

## Objectives

- **Recall The Solution Series field reports**
- **Create an option list for field validation**
- **Identify the components of a field definition**
- **Create a field definition**

---

### NOTES

---

## Overview

### **Purpose**

In this section, you will learn about how to define a field in The Solution Series as well as utilities used to maintain field definitions.

### **Objectives**

When you complete this section, you will be able to:

- Recall field reports
- Create an option list for field validation
- Identify the components of a field definition
- Create a field definition

## Field Reports

- **Field table list (FTLIST)**
- **Field table menu (F-MENU)**
- **Screen label to field name cross-reference (FLABEL)**
- **Segment layout report (SRTFLD/F-SEGM)**
- **Field to program cross-reference (CROSSX/CROSSP)**
- **Program memory map (MAPRPT)**

---

NOTES

---

## The Solution Series field reports

### Field reports

As you recall from the Introduction to Cyborg Scripting Language course, there are six reports that supply CSL programmers with valuable information:

- **Field Table List (FTLIST)**  
Online program that enables you to view the field definitions in alphabetic order by segment/pointer. The online display is in scrolling format. A batch run produces a printout with headings and page numbers.
- **Field Table Menu (F-MENU)**  
Online program that allows you to view the attributes of the data fields in the Field Name Table in a user-friendly, menu-driven format in displacement order by segment/pointer.
- **Screen Label To Field Name Cross Reference (FLABEL)**  
Cross-reference between the field labels used on a form and their Field Name Table field names.
- **Segment Layout Report (SRTFLD/F-SEGM)**  
Batch programs that produce a report that displays each segment's layout in position order.
- **Field to Program Cross Reference (CROSSX/CROSSP)**  
Batch programs that produce a report that cross-references where all fields and verbs are used within a program.
- **Program Memory Map (MAPRPT)**  
Batch program that produces a report that displays the memory usage within a program.



## Creating option lists

### Option lists

As you recall, an option list is a list of values that are valid for a particular field on a given form. To create a new option list:

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Programming Utilities
<b>Task:</b>		Update Control Repository Objects

**Result:** The Edit prompt form is displayed.

Use the following Edit Utility form (EDIT) fields to create an option list:

<b>Object</b>	Select Codeset Value (C/V).
<b>Object Key</b>	Type the Option List name. Press Enter.

**Result:** The Option List Edit form is displayed.

### Edit columns

The Edit Utility form (EDIT) for option list values is in a different format than the source code. The column definitions are:

- **Line Command**  
Specifies whether you are adding, changing or deleting the line.
- **Code**  
The code value that will be stored in the record.
- **Name**  
The name of the option list or the description of the option value. The first line of all option lists must contain the option list name with a blank value.
- **Other Values**  
The alternate language description of the option value.

## Field Name Table

- **Data fields**
- **Key fields**
- **Date fields**
- **Option list fields**

---

**NOTES**

---

## Field definitions

### Fields

There are several different kinds of fields used in The Solution Series:

- Data fields  
can be alphanumeric or numeric.
- Key fields  
uniquely identify segments and records.
- Date fields  
can be century/complement, regular, or time-span format.
- Option list fields  
access a list of valid values.

Before you can refer to a field in a program, the field must be defined to the system. This means that the definition must reside on the field name table.

### Field name table

The field name table resides on the system control repository (FILE01). When delivered, it contains the names and definitions for all existing entry and inquiry fields, CSL words and verbs, and defined files. If you want to create, modify, or delete fields, you can do so through the Field-Name Entry/Maintenance form (F-NAME).

Relational database users should use caution when re-defining existing fields using the field name table. The redefinition will only exist within The Solution Series since relational table columns may only have one name.

# Field Maintenance and Edit

Field Maintenance And Edit

Action:   
Field Name:

Field Location

Pointer:   
Storage Length:   
Displacement:

Field Options

Propagate:   
Rounding:   
Header Switch:   
 RDBMS Field

Field Properties

Data Type:   
Field Type:   
Template:   
Lengths: Display:  Entry:   
Module:   
Structure:   
Seg/Table ID:  Table Separator:   
Codeset:   
Edit Routine:

---

## NOTES

---

## Field definitions, continued

### Field maintenance

The Field Maintenance And Edit form (F-NAME), contains a group of fields that together provide a unique definition for every field in The Solution Series. In order to access the Field Maintenance And Edit form (F-NAME) perform one of the following:

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Fields and Verbs
- Task:**  Define a Field

**Result:** The Field Maintenance And Edit form (F-NAME) is displayed.

### F-NAME fields

The following pages detail each of the fields on the Field Maintenance and Edit form (F-NAME). Together, these fields uniquely identify a field in The Solution Series.



## Field definitions, continued

### **ACTION**

This one-character field lets you create, modify, or delete a field definition.

- A (ADD)—to create a new field
- C (CHANGE)—to modify an existing field
- D (DELETE)—to remove a field from the system.

### **FIELD NAME**

This 20-character Key field contains the field name. It serves as the key to the Field Name Table record.

- Separate words in a multi-word name with hyphens (-).
- Begin all user-defined field names with the letter X to distinguish them from Cyborg-delivered fields.
- Alternatively, user-defined multi-word field names may begin with any letter/number if an asterisk (\*) is used to separate the words.

*Note: Certain Relational Database systems limit the size of field names to 18 characters.*

### **POINTER**

This two-digit field defines the field's location and the segment to which it belongs. Currently, valid values are 05 to 45. For example, a field that stores basic data about employees resides in Pointer 29 (E segment).

### **STORAGE LENGTH**

This three-digit field physically defines the field's storage length. Fields in The Solution Series can be up to 60 characters long. Valid values are 000 to 060.

### **DISPLACEMENT**

This three-digit field defines the field's physical location within the pointer by defining where the field starts. Displacement is relative to zero (0). In user-defined fields, the minimum value for this field is 003 to allow for the segment type and segment code.

## FIELD TYPE

P	The field is a partial segment key, but is not the last field in the key.
K	The field is the only segment key, or is the last field in a string of segment key fields.
D	The field is an Option List description.
J	The field is the first data field in a single-occurrence segment code.
G	The field is a required, non-key field. (None). The field is a regular data entry field.

---

## NOTES

---

## Field definitions, continued

### **DATA TYPE**

This two-character field defines the kind of information and number of decimals that the field stores. Several data types are reserved for Cyborg use only.

### **FIELD TYPE**

This one-character field indicates whether the field is a segment key, an option list description, a required field, or a data entry field.

### **TEMPLATE**

This two-digit field defines the edit routine or mask that the system uses when it displays the field. Numeric and date fields require these values.

The actual number of decimal places is defined in the DATA TYPE field. Therefore, the value that you select in the DATA TYPE field must agree with the value that you select in the TEMPLATE field, or the system may truncate the data.

### **DISPLAY LENGTH**

This two-digit field defines the number of output characters that is displayed on a form or report, based on the template.

### **ENTRY LENGTH**

This two-digit field defines the number of characters that you can enter in a field. It is used for numeric and date fields. Valid values are numbers from 01 through 60 and blank. This field defines the length of the entry box on a form.

*Note: When this field is left blank and the DATA TYPE is numeric, the STORAGE LENGTH is used to determine the size of the ENTRY LENGTH.*

*Note: When this field is left blank and the DATA TYPE is a date, the larger of the STORAGE LENGTH or DISPLAY LENGTH is used to determine the size of the ENTRY LENGTH.*

## STRUCTURE

K	The field resides within a multiple-occurrence segment that is uniquely identified by the Segment Type and Segment Code (Company C segment, Employee E and L segments).
8	The field resides within a single-occurrence segment that is uniquely identified by the Segment Type (Company A and B segments, Employee F, G, H, J, and P segments).

## SEGMENT CODES

For user-defined segments, the valid segment codes are:

- Segment C (Company-level data) Bx and Cx
- Segment L (Employee-level data) Lx, Mx, and Nx where x is 1 through 9 or A through Z.

---

## NOTES

---

## Field definitions, continued

### **MODULE**

This two-character field defines the application to which the field belongs. If you use the PC Solution you must enter one of the valid values.

### **STRUCTURE**

This one-character field defines how the system maintains the field by specifying whether the field resides on a single- or multiple-occurrence segment type.

### **SEG/TABLE ID**

This four-character field serves two purposes: to define the segment that a company, tax or employee field resides within, or to identify the Table ID for fields that reside in table records (Pointer 40).

### **TABLE SEPARATOR**

This one-character field is used to identify the fifth position of a Table ID when a logical table spans more than one physical record.

### **CODESET**

This five-character field contains the name of the option list record group that lists the valid values for the field.

### **EDIT ROUTINE**

This field contains the name of the CSL program that will be called each time a value is entered in this field via a data entry form.

# ROUNDING

## Rounding Methods

- U Based on the rounding digit value, increase the number by one.
- D Based on the rounding digit value, truncate the number to that position.
- 5 If the rounding value is 5 or greater, increase the number by one. If the rounding value is less than 5, truncate the number to that position.

## Rounding Digit

- 0–9 The rounding digit specifies what place in the number is rounded, such as the penny or dollar places in a dollar amount. Positions are counted right to left, and valid values are 0 through 9.

Example: The result of a calculation is 101.454; however, the field that stores the value has only 2 decimals. This example shows the effect of rounding:

ROUNDING Field	Stored Result
50 (default)	101.45
52	101.00
U0	101.46
U1	101.50
U2	102.00
D0	101.45
D1	101.40
D2	101.00

---

## NOTES

---

## Field definitions, continued

### **PROPAGATE**

This field defines a field's propagation parameters. Propagation is a system function that permits you to update an Employee Database (FILE02) record without entering duplicated information again. It is valid for fields that reside in stacked segments. If you enter a value in this field, the system tries to match the segment type, segment code, and the specified number of bytes of key data for the segment when you update the record. If it finds a match on an existing occurrence, it inserts the existing data into the new segment's field automatically.

Valid entries are to match the Segment ID, the Segment Code, and from 0 to 9 bytes of Key data. Specifying 0 means that you want the system to match only the segment type and segment code. Choose (None) or leave the field blank if you do not want to propagate the field.

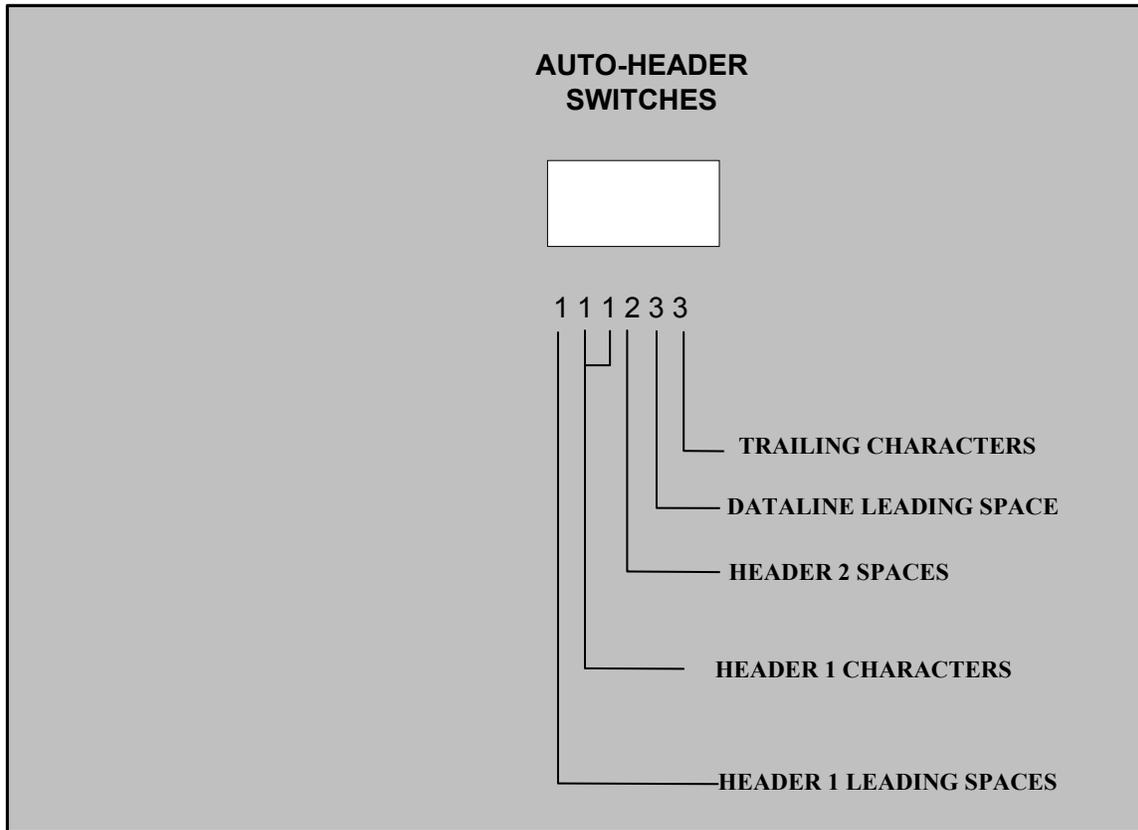
### **ROUNDING**

This two-character field defines how a numeric field is decreased when the system must truncate it. The ROUNDING field is made up of two distinct values:

- Rounding method
- Rounding digit

Rounding is used by the MOVE and CALCULATE verbs when the receiving field is smaller than the sending field.

# HEADER SWITCH FIELDS



---

## NOTES

---

## Field definitions, continued

### HEADER SWITCH

The header switches are used by online forms and reports to create column headers from field names. When you add a new field, leave this field blank and the system automatically supplies appropriate values.

- Position 1  
This field lets you specify the number of spaces that you want inserted before the first heading line. Minimum of 1.
- Positions 2–3  
This field lets you specify the number of characters that should appear in the first heading line. The remaining characters are moved to the second heading line.
- Position 4  
This field lets you specify the number of spaces that you want inserted before the second heading line. Minimum of 1.
- Position 5  
This field lets you specify the number of spaces that should precede the field and position it under the heading.
- Position 6  
This field lets you specify the number of spaces that should follow the field.

### RDBMS field

For relational database only, this field requires a Y when the field is to be used in creating a column entry in a relational table.

# Partial Key Field

The screenshot shows the 'Field Maintenance And Edit' dialog box for a 'Partial Key Field'. The 'Field Name' is 'PARTIAL-KEY'. Under 'Field Location', the 'Pointer' is 36, 'Storage Length' is 006, and 'Displacement' is 003. Under 'Field Options', 'Propagate' and 'Rounding' are dropdown menus, 'Header Switch' is 207410, and the 'RDBMS Field' checkbox is unchecked. The 'Field Properties' section includes: 'Data Type' (Century/Complement D), 'Field Type' (Key-Part of Key), 'Template' (Date), 'Lengths' (Display: 10, Entry: unchecked), 'Module' (Payroll/HRMS), 'Structure' (Seg has ID and Code), 'Seg/Table ID' (M8), 'Table Separator' (unchecked), 'Codeset' (empty), and 'Edit Routine' (empty).

# Key Field

The screenshot shows the 'Field Maintenance And Edit' dialog box for a 'Key Field'. The 'Field Name' is 'KEY-FIELD'. Under 'Field Location', the 'Pointer' is 36, 'Storage Length' is 004, and 'Displacement' is 003. Under 'Field Options', 'Propagate' and 'Rounding' are dropdown menus, 'Header Switch' is 100210, and the 'RDBMS Field' checkbox is checked. The 'Field Properties' section includes: 'Data Type' (Alphanumeric), 'Field Type' (Key-Ending/Only Key), 'Template' (empty), 'Lengths' (Display: unchecked, Entry: unchecked), 'Module' (Payroll/HRMS), 'Structure' (Seg has ID and Code), 'Seg/Table ID' (M8), 'Table Separator' (unchecked), 'Codeset' (empty), and 'Edit Routine' (empty).

---

## NOTES

---

## Field definitions, continued

### Field examples

The next several pages show the generic field characteristics for the most common field types.

### Partial key field

Partial key fields are part of the key that uniquely identifies a multiple occurrence segment. Partial key fields:

- May be any type of field type: alphanumeric, numeric, name, or option list fields.
- Must reside between displacement 003 and 015 for user-defined segments (displacement 003 if it is the first partial key field).
- Must be identified with a FIELD TYPE of Key–Part of Key (P).

### Key field

Key fields are the last or the only key that uniquely identify a multiple occurrence segment. Key fields:

- May be any field type: alphanumeric, numeric, name, or option list fields.
- Must reside between displacement 003 and 015 for user-defined segments (displacement 003 if it is the only key field).
- Must be identified with a FIELD TYPE of Key–Ending/Only Key (K).

# Alphanumeric Field

Field Maintenance And Edit

Action:    
Field Name: ALPHANUMERIC - FIELD

Field Location

Pointer: 36   
Storage Length: 030   
Displacement: 015

Field Options

Propagate:    
Rounding:    
Header Switch: 112575   
 RDBMS Field

Field Properties

Data Type: Alphanumeric   
Field Type:   
Template:   
Lengths: Display:  Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: M8 Table Separator:   
Codeset:   
Edit Routine:

# Numeric Field

Field Maintenance And Edit

Action:    
Field Name: NUMERIC - FIELD

Field Location

Pointer: 36   
Storage Length: 004   
Displacement: 048

Field Options

Propagate:    
Rounding:    
Header Switch: 107220   
 RDBMS Field

Field Properties

Data Type: Numeric 0 Decimals   
Field Type:   
Template: ---,---,---,---,---   
Lengths: Display: 05 Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: M8 Table Separator:   
Codeset:   
Edit Routine:

## NOTES

## Field definitions, continued

### **Alphanumeric field**

Alphanumeric fields may contain both alpha and numeric data. Alphanumeric fields:

- Are identified by a DATA TYPE of alphanumeric (00).
- Do not require an ENTRY LENGTH, DISPLAY LENGTH, or TEMPLATE.

### **Numeric field**

Numeric fields contain only numbers. Numeric fields:

- Are identified by a DATA TYPE of numeric with 0 to 6 Decimals (10–16).
- Require an ENTRY LENGTH equal to the STORAGE LENGTH or blank and the default is to the STORAGE LENGTH.
- Require a DISPLAY LENGTH; this will be dependent on the TEMPLATE.
- Require a TEMPLATE.

# Name Field

The screenshot shows the 'Field Maintenance And Edit' dialog box. The 'Field Name' is 'NAME-FIELD'. Under 'Field Location', the 'Pointer' is 36, 'Storage Length' is 030, and 'Displacement' is 015. Under 'Field Options', 'Propagate' and 'Rounding' are dropdown menus, 'Header Switch' is 204210, and 'RDBMS Field' is checked. Under 'Field Properties', 'Data Type' is Name, 'Field Type' is empty, 'Template' is empty, 'Lengths' has 'Display' and 'Entry' checkboxes, 'Module' is Payroll/HRMS, 'Structure' is Seg has ID and Code, 'Seg/Table ID' is M8, 'Table Separator' is empty, 'Codeset' is empty, and 'Edit Routine' is empty.

# Required Field

The screenshot shows the 'Field Maintenance And Edit' dialog box. The 'Field Name' is 'REQUIRED-FIELD'. Under 'Field Location', the 'Pointer' is 36, 'Storage Length' is 004, and 'Displacement' is 048. Under 'Field Options', 'Propagate' and 'Rounding' are dropdown menus, 'Header Switch' is 107220, and 'RDBMS Field' is checked. Under 'Field Properties', 'Data Type' is Alphanumeric, 'Field Type' is Required Non-key, 'Template' is empty, 'Lengths' has 'Display' and 'Entry' checkboxes, 'Module' is Payroll/HRMS, 'Structure' is Seg has ID and Code, 'Seg/Table ID' is M8, 'Table Separator' is empty, 'Codeset' is empty, and 'Edit Routine' is empty.

---

## NOTES

---

## Field definitions, continued

### **Name field**

Name fields may contain both alpha and numeric data.

Name fields:

- Are identified by a DATA-TYPE of NAME (AO).
- Typically have a FIELD-LENGTH of 30.
- Do not require an ENTRY LENGTH, DISPLAY LENGTH, or TEMPLATE.

### **Required field**

Required fields may be any data type. Required fields:

- Are identified by a FIELD TYPE of Required Non-Key (G).
- Follow the rules for the particular DATA TYPE.

# Code Field

The screenshot shows a 'Field Maintenance And Edit' window for a field named 'CODE-FIELD'. The window is divided into several sections:

- Action:** A small icon.
- Field Name:** CODE-FIELD
- Field Location:** Pointer: 36, Storage Length: 003, Displacement: 045
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 104220,  RDBMS Field
- Field Properties:** Data Type: Alphanumeric, Field Type: (dropdown), Template: (dropdown), Lengths: Display:  Entry:  Module: Payroll/HRMS, Structure: Seg has ID and Code, Seg/Table ID: M8, Table Separator: (dropdown), Codeset: SCXX, Edit Routine: (dropdown)

# Code Description Field

The screenshot shows a 'Field Maintenance And Edit' window for a field named 'CODE-DESCRIPTION'. The window is divided into several sections:

- Action:** A small icon.
- Field Name:** CODE-DESCRIPTION
- Field Location:** Pointer: 36, Storage Length: 003, Displacement: 045
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 100310,  RDBMS Field
- Field Properties:** Data Type: Alphanumeric, Field Type: Codeset Description, Template: (dropdown), Lengths: Display: 20, Entry:  Module: Payroll/HRMS, Structure: Seg has ID and Code, Seg/Table ID: M8, Table Separator: (dropdown), Codeset: SCXX, Edit Routine: (dropdown)

---

## NOTES

---

## Field definitions, continued

### **Code field**

Code fields must be established in conjunction with the option list in which the value resides. Code fields:

- Must be defined as a data type of alphanumeric (00).
- Require a CODESET entry to identify where to validate values.
- Have a maximum STORAGE LENGTH of 14.

### **Code description field**

Code description fields display the description from the option list in which the code field resides and are not part of the segment layout. Code description fields:

- Must have the same field location and field properties as the code field, with the following exceptions:
  - Have a DISPLAY LENGTH of up to 20.
  - Are identified with a FIELD TYPE of Codeset Description (D).

# Regular Date Field

The screenshot shows the 'Field Maintenance And Edit' window for a 'Regular Date' field. The window is divided into several sections:

- Action:** A small icon.
- Field Name:** REGULAR-DATE
- Field Location:** Pointer: 36, Storage Length: 006, Displacement: 015
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 207310,  RDBMS Field
- Field Properties:** Data Type: Regular Date, Field Type: (dropdown), Template: Date, Lengths: Display: 08, Entry: (checkbox), Module: Payroll/HRMS, Structure: Seg has ID and Code, Seg/Table ID: M8, Table Separator: (checkbox), Codeset: (dropdown), Edit Routine: (dropdown)

# Time-span Date Field

The screenshot shows the 'Field Maintenance And Edit' window for a 'Time-span Date' field. The window is divided into several sections:

- Action:** A small icon.
- Field Name:** TIME-SPAN-DATE
- Field Location:** Pointer: 36, Storage Length: 006, Displacement: 015
- Field Options:** Propagate: (dropdown), Rounding: (dropdown), Header Switch: 109420,  RDBMS Field
- Field Properties:** Data Type: Time Span Date, Field Type: (dropdown), Template: Date, Lengths: Display: 08, Entry: (checkbox), Module: Payroll/HRMS, Structure: Seg has ID and Code, Seg/Table ID: M8, Table Separator: (checkbox), Codeset: (dropdown), Edit Routine: (dropdown)

---

## NOTES

---

## Field definitions, continued

### Regular date field

Regular date fields are used when the date is not a key or partial-key field.

Regular date fields:

- Must be defined with a DATA TYPE of Regular Date (80).
- Require an ENTRY LENGTH equal to the STORAGE LENGTH or blank and the default is to the greater of the STORAGE LENGTH or DISPLAY LENGTH. Typically the ENTRY LENGTH is left blank.
- Require a DISPLAY LENGTH. DISPLAY LENGTHs can be
  - 05 MMDD storage format
    - DD–MM edit format
  - 08 YYMMDD storage format
    - MM–DD–YY edit format for US and Canada
    - DD–MM–YY edit format elsewhere
  - 10 CCYYMMDD storage format
    - MM–DD–CCYY edit format for US and Canada
    - DD–MM–CCYY edit format elsewhere
- Require a TEMPLATE of Date (34).

### Time–span date field

Time–span date fields are used to display the elapsed number of days, months and years. Time–span dates are not stored and should be the result of a date calculation. Time–span dates:

- Must be defined with a DATA TYPE of Time Span Date (81).
- Require an ENTRY LENGTH equal to the STORAGE LENGTH or blank and the default is to the greater of the STORAGE LENGTH or DISPLAY LENGTH. Typically the ENTRY LENGTH is left blank.
- Require a DISPLAY LENGTH of 08 (YYMMDD storage format, YY–MM–DD edit format).
- Require a TEMPLATE of Date (34).

# Century Date Field

# Century Date Calculation

**CENTURY** is calculated as:  
 22 - xx, where xx equals the century being translated. If the DISPLAY LENGTH is not equal to 10, the 20th Century is assumed.

**YEAR** is calculated as:  
 99 - yy, where yy equals the year to be translated.

**MONTH** is calculated using the following alpha character:

Jan = L	Jul = F
Feb = K	Aug = E
Mar = J	Sep = D
Apr = I	Oct = C
May = H	Nov = B
Jun = G	Dec = A

**DAY** is calculated as:  
 32 - zz, where zz equals the day.

## NOTES

## Field definitions, continued

### Century date field

Century date fields are usually used when the date is a key or partial key field. Century/compliment dates cause the most recent occurrence to be shown first. Because the latest occurrence is the most frequently referenced, this is an efficient arrangement. Century date fields:

- Must be defined with a DATA TYPE of Century/Complement Date (82).
- Require an ENTRY LENGTH equal to the STORAGE LENGTH or blank and the default is to the greater of the STORAGE LENGTH or DISPLAY LENGTH. Typically the ENTRY LENGTH is left blank.
- Require a DISPLAY LENGTH. DISPLAY LENGTHs can be
  - 05 MMDD storage format
    - DD–MM edit format
  - 08 CYYMDD storage format
    - MM–DD–YY edit format for US and Canada
    - DD–MM–YY edit format elsewhere
  - 10 CYYMDD storage format
    - MM–DD–CCYY edit format for US and Canada
    - DD–MM–CCYY edit format elsewhere
- Require a TEMPLATE of Date (34).
- Translate automatically when used with the ENTRY, INQUIRY, or PRINT verb.

## Section Summary

- **The Solution Series field reports**
- **Option Lists—field validation**
- **Field definitions**
- **Documenting fields**

---

NOTES

---

## Section summary

In this section, you learned the components of defining a field to The Solution Series. Specifically, you learned:

**The Solution Series field reports**

---

---

**Option lists—field validation**

---

---

**Field definitions**

---

---

# Section 4 Exercise

---

**NOTES**

---

## Section 4 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. Use any of the Online Field Utilities (FTLIST, F-MENU, or F-NAME) to answer the following questions:

- a) What is the Segment Key Field(s) for the Employee Name & Address Segment, Pointer 30?  
\_\_\_\_\_

- b) What type of date is SALARY-EFFECTIVE?  
\_\_\_\_\_

- c) To what does the ENTRY LENGTH field default when left blank?  
\_\_\_\_\_

2. Create an Employee Supplies Option List SCZZ as follows:

<u>Codes</u>	<u>Description</u>
001	Black Pens/Medium (Box of 10)
002	#2 Pencils (Box of 12)
020	Legal Pad of Paper
034	Post-it Note Pads (3x5)
035	Scotch Tape
040	Staples

## Section 4 Exercise, continued

---

**NOTES**

---

**Section 4 exercise, continued**

3. Create new field definitions for existing Cyborg fields in order to access shortened versions of the fields or a specific part of the fields.

<b><u>Cyborg Field</u></b>	<b><u>New Field</u></b>	<b><u>Description</u></b>
OTHER-NAME	XOTHER-NAME-20	The first 20 positions of the Employee's Spouse / Dependent name.
EMPLOYER-NAME	XEMPLOYER-NAME-20	The first 20 positions of the Employee's Spouse / Dependent employer.

---

**NOTES**

---

## Section 5: User-Defined Segments

---

### Table of Contents

Overview.....	135
User-defined segments overview.....	137
Data requirements analysis.....	139
User-defined segment layout.....	141
User-defined field definitions.....	145
User-defined segment layout—continuation segment.....	153
User-defined entry form.....	155
Verify user-defined definitions.....	157
Query-related programs.....	161
Section summary.....	167
Section 5 exercise.....	170

## Objectives

- **Identify the steps to create user-defined fields/segments**
- **Analyze user-defined data requirements**
- **Create a user-defined segment layout**
- **Create new field definitions for a user-defined segment**
- **Create a data entry form for a user-defined field/segment**
- **Verify the user-defined definitions**
- **Create a FIND– verb**
- **Create Solution View (WRITER) cross-reference records**

---

### NOTES

---

## Overview

### **Purpose**

In this section, you will learn the steps to create user-defined segment fields and a form for data entry of the fields.

### **Objectives**

When you complete this section, you will be able to do the following:

- Identify the steps to create a user-defined field/segment
- Analyze user-defined data requirements
- Create a user-defined segment layout
- Create new field definitions for a user-defined segment
- Create a data entry form for a user-defined field/segment
- Verify the user-defined definitions
- Create a FIND- verb
- Create Solution View (WRITER) cross-reference records

## **Creating User-Defined Segments**

- **Analyze the data requirements**
- **Create the segment layout**
- **Create the field definitions**
- **Create the data entry form**
- **Enter data into the fields  
through the data entry form**

---

### **NOTES**

---

## User-defined segments overview

### User-defined segments

The Solution Series provides the capability to create user-defined segments for the company and employee records. To create user-defined segments complete the following procedure:

- Analyze the data requirements
- Create the segment layout
- Create the field definitions
- Create the data entry form
- Enter data into the fields through the data entry form

### RDBMS users

If user-defined segments are created using the Field Maintenance And Edit form (F-NAME), the COBOL programs that access the fields will need to be re-generated and the relational tables must be re-built before the new fields/tables will be accessible to The Solution Series.

The Solution View New Fields Definition program (NEWSCR) may be used to create user-defined fields/tables dynamically.



*Refer to the Technical Administration documentation for further information.*

# User-Defined Request and Analysis

## MEMO

To: Mike Pahud, Cyborg Online Programmer  
From: Joy Ross, HRMS Coordinator  
Subject: Tracking Work-At-Home equipment

With the implementation of our new Work-At-Home policy, it will be necessary to track any company equipment being used by employees in their home offices.

To keep the data entry and maintenance requirements to a minimum, we would like to integrate this data into The Cyborg Solution Series system since it already contains information such as the employee's number, name, and address.

The Work-At-Home information to be maintained should include a description of the equipment, an equipment category, a serial number, the date the equipment was issued, and the date the equipment is returned. It is feasible that each employee will have several different types of equipment in their home, such as computer, fax machine, printer, and so forth. I have already created an Option List named WH01 that contains the equipment categories to be used.

If you have any questions, please contact me at extension 1231.

---

## NOTES

---

## Data requirements analysis

### Data requirements

The first step in creating user-defined segment fields is to analyze the data requirements. This analysis will include questions to help understand how and where the data is to be stored:

- Does the data need to be stored in each employee's record or simply at the company level?
- Are there existing Cyborg-delivered segments/fields to accommodate the requirements?
- Does the data require more than one instance or a history of past information?
- If the segment is multiple-occurrence, what field(s) uniquely identifies one segment of data from another?
- Do any fields require validation? Are any fields required?

### Example analysis

Analysis of the above memorandum using the analysis questions provides the following data requirements for the user-defined segment:

- The data is to be stored in each employee's record.
- There are no existing Cyborg fields to satisfy this request.
- Employees may have more than one item of equipment at their home. Therefore, the data requires a multiple-occurrence segment structure.
- A counter could be used to uniquely identify each occurrence of Work-At-Home equipment.
- The equipment category will be validated using an option list, and the serial number is required.



## User-defined segment layout

### Segment layout

The second step in creating user-defined segment fields is to construct a segment layout for the user-defined segment.

- The segment layout must always begin with the segment type and segment code beginning at displacement 000:

Company Level	84 maximum length
	C segment type, pointer 23
	Bx or Cx segment code
Employee Level	71 maximum length
	L segment type, pointer 36
	Lx, Mx or Nx segment code

Where x is an alphanumeric character.

- Segment key fields, if any, must follow the segment type/code on the segment layout beginning at displacement 003. If you are creating a single-occurrence segment, place your data fields after the segment type/code and designate the first field with a FIELD-TYPE = J.
- Segment keys can be a maximum of 15 characters long, however the segment type and code already use 3 displacements leaving 12 for the user-defined segment key.
- Data should not be placed in the last two positions of the segment. These positions may be used by payroll processing to determine if changes have occurred in the segment.



## User-defined segment layout, continued

### Segment layout example

The above example shows the result of populating the segment layout with the information derived from the analysis and memo requirements.

- The segment layout shows the physical structure of the segment.
- This step is critical in our process to create a user-defined segment since it will be used in subsequent steps, for example, creating the field definitions.

# Equipment Number

Field Maintenance And Edit

Action:    
Field Name: XEQUIP-NBR

Field Location

Pointer: 36  
Storage Length: 002  
Displacement: 003

Field Options

Propagate:    
Rounding:    
Header Switch: 106332   
 RDBMS Field

Field Properties

Data Type: Numeric 0 Decimals   
Field Type: Key-Ending/Only Key   
Template:   
Lengths: Display:  Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: L1 Table Separator:   
Codeset:   
Edit Routine:

# Equipment Description

Field Maintenance And Edit

Action:    
Field Name: XEQUIP-DESC

Field Location

Pointer: 36  
Storage Length: 030  
Displacement: 005

Field Options

Propagate:    
Rounding:    
Header Switch: 100110   
 RDBMS Field

Field Properties

Data Type: Alphanumeric   
Field Type:   
Template:   
Lengths: Display:  Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: L1 Table Separator:   
Codeset:   
Edit Routine:

---

## NOTES

---

## User-defined field definitions

### Define fields

The third step in creating user-defined segment fields is to define the fields to the Field Name Table. This step is accomplished by using the Field Maintenance And Edit form (F-NAME).

### Field example

The segment layout is used to help determine the field's definition. The following describes each field's definition:

#### **XEQIP-NBR**

This field is a key field that contains two digits. Its display length is two positions with 0 decimals and is edited as: 99.

#### **XEQIP-DESCRIPTION**

This field is an alphanumeric field that contains 30 characters.

# Equipment Category

Field Maintenance And Edit

Action:    
Field Name: XEQUIP-CATEGORY

Field Location

Pointer: 36   
Storage Length: 005   
Displacement: 035

Field Options

Propagate:    
Rounding:    
Header Switch: 206131   
 RDBMS Field

Field Properties

Data Type: Alphanumeric   
Field Type:   
Template:   
Lengths: Display:  Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: L1 Table Separator:   
Codeset: WH01   
Edit Routine:

# Equipment Category Description

Field Maintenance And Edit

Action:    
Field Name: XEQUIP-CATEGORY-DESC

Field Location

Pointer: 36   
Storage Length: 005   
Displacement: 035

Field Options

Propagate:    
Rounding:    
Header Switch: 100110   
 RDBMS Field

Field Properties

Data Type: Alphanumeric   
Field Type: Codeset Description   
Template: (None)   
Lengths: Display: 20 Entry:    
Module: Payroll/HRMS   
Structure: Seg has ID and Code   
Seg/Table ID: L1 Table Separator:   
Codeset: WH01   
Edit Routine:

## NOTES

## User-defined field definitions, continued

### **XEQUIP-CATEGORY**

This field is an alphanumeric field that contains five characters. It is edited against option list WH01.

### **XEQUIP-CATEGORY-DESC**

This field is an alphanumeric field that contains 20 characters. It is a value found in option list WH01. This field displays the description from the option list translation of the XEQUIP-CATEGORY field.

*Note: The option list description is not part of the segment layout. The option list description is found in the option list and is referenced by the use of the option list field.*

# Equipment Serial Number

Field Maintenance And Edit

Action:   
Field Name: XEQUIP-SERIAL-NBR

Field Location

Pointer: 36  
Storage Length: 010  
Displacement: 040

Field Options

Propagate:   
Rounding:   
Header Switch: 306110  
 RDBMS Field

Field Properties

Data Type: Alphanumeric  
Field Type: Required Non-key  
Template:   
Lengths: Display:  Entry:   
Module: Payroll/HRMS  
Structure: Seg has ID and Code  
Seg/Table ID: L1 Table Separator:   
Codeset:   
Edit Routine:

# Equipment Issue Date

Field Maintenance And Edit

Action:   
Field Name: XEQUIP-ISSUE-DATE

Field Location

Pointer: 36  
Storage Length: 006  
Displacement: 050

Field Options

Propagate:   
Rounding:   
Header Switch: 306121  
 RDBMS Field

Field Properties

Data Type: Century/Complement D  
Field Type:   
Template: Date  
Lengths: Display: 10 Entry:   
Module: Payroll/HRMS  
Structure: Seg has ID and Code  
Seg/Table ID: L1 Table Separator:   
Codeset:   
Edit Routine:

---

## NOTES

---

## User-defined field definitions, continued

### **XEQUIP-SERIAL-NBR**

This field is an alphanumeric field that contains ten characters.

### **XEQUIP-ISSUE-DATE**

This field is a century complement date field that contains six digits and is retained on the file in CYYMDD format. It is displayed as MM-DD-CCYY in the US and Canada, DD-MM-CCYY elsewhere.

# Equipment Return Date

The screenshot shows a 'Field Maintenance And Edit' window for the field 'EQUIP-RETURN-DATE'. The window is split into two main sections: 'Field Location' and 'Field Properties'.

**Field Location:**

- Action: [ ]
- Field Name: **EQUIP-RETURN-DATE**
- Field Location: [ ]
- Pointer: **36**
- Storage Length: **006**
- Displacement: **056**

**Field Options:**

- Propagate: [ ]
- Rounding: [ ]
- Header Switch: **406121**
- RDBMS Field

**Field Properties:**

- Data Type: **Century/Complement D**
- Field Type: [ ]
- Template: **Date**
- Lengths: Display: **10** Entry: [ ]
- Module: **Payroll/HRMS**
- Structure: **Seg has ID and Code**
- Seg/Table ID: **L1** Table Separator: [ ]
- Codeset: [ ]
- Edit Routine: [ ]

---

## NOTES

---

## User-defined field definitions, continued

### **XEQUIP-RETURN-DATE**

This field is a century complement date field that contains six digits and is retained on the file in CYYMDD format. It is displayed as MM-DD-CCYY in the US and Canada, DD-MM-CCYY elsewhere.

### **Update FILECL**

The Client File (FILECL) must be updated to include the new field definitions before Form Builder can use them. The EDIT program does not automatically update the Client File. To update the Client File:

Make the following selections from the Navigator:

<b>Component:</b>		User Tools
<b>Process:</b>		User Tools
<b>Task:</b>		Refresh Client Data

**Result:** ---Sign-on Completed--- is displayed on the form.



## User-defined segment layout—continuation segment

### Large segment layouts

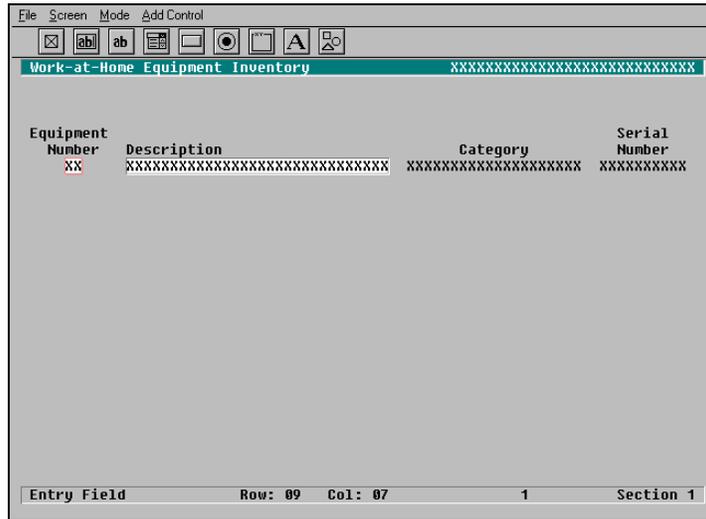
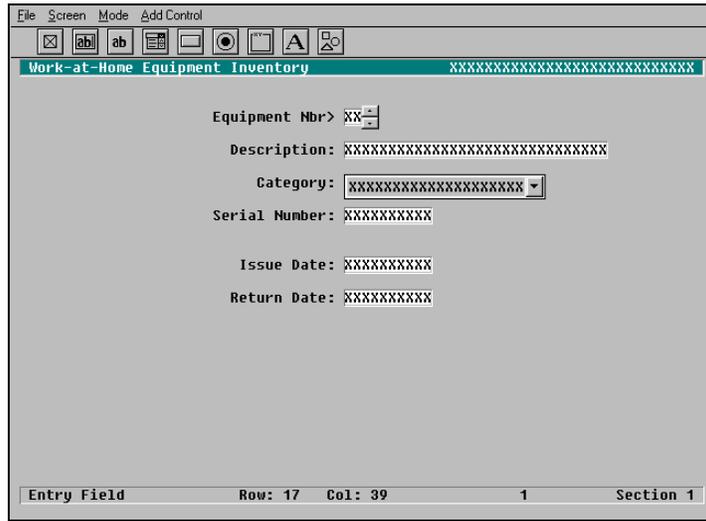
As you recall, each user-defined segment has a fixed length (Company level C segment 84 positions, Employee level L segment 71 positions). If the user-defined segment you are creating requires more than the fixed length, you may use additional segments to continue the segment definition.

- When you use more than one segment code, you must follow these rules:
- Define field definitions for the key fields from the first segment on the second and subsequent segments, using the exact same field formats but different field names.
- Additional coding will be required to delete the additional segments since the system will delete only one segment for you. See the Spouse/Dependent Information form (10-SCR) for an example.
- RDB users using Solution View to create new segments would define each segment one at a time.

# Create Data Entry Form

```

SECURITY.@ Work-at-Home Inventory
P100-START-SCREEN.
KEY-REQUIRED.
UPDATE-EMPLOYEE.
NEW-SCREEN-STYLE.
SCREEN-SECTION '0'.
IF INQUIRY-MODE OR SELECTION-MODE
    GO TO P990-INQUIRY-SCREEN.
P200-ENTRY-SCREEN.
SCREEN-SECTION '1'.
P300-VERIFY.
SET-FOR-MESSAGES.
IF ERRORS-EXIST RETURN.
IF RECORD-NOT-UPDATED
    RETURN.
CALL 'XXXXXX'. @ EDIT ROUTINES
RETURN.
P990-INQUIRY-SCREEN.
SCREEN-SECTION '8'.
RETURN.
P999-PROMPTS.
SCREEN-SECTION '9'.
    
```



## NOTES

## User-defined entry form

### Data entry form

The fourth step in creating user-defined segment fields is to create the data entry form for the fields to be updated. This program will be created using the form design application, the Edit Utility (EDIT), and the verbs and techniques discussed in Section 3.

Remember, even though you may be creating a single-occurrence segment, the segment physically resides in a multiple-occurrence segment (Company C segment, Employee L segment). Therefore the following rules apply:

- All key fields must precede the non-key fields.
- If more than one segment is included on the form, fields from different segments must be placed on separate data entry lines or in separate sections.

### Context-sensitive menus

An optional step in creating user-defined segment fields is to create the context-sensitive menus for the entry form. These are created using the Edit Utility (EDIT) as discussed in Section 3.

# Data Entry Form

## Entry and Inquiry/Select Mode

### Entry Mode

Work-at-Home Equipment Inventory AUSTIN, STEVEN

Equipment Nbr: 1

Description: Compaq LTE 5000

Category: Computer

Serial Number: JJD78398DJ

Issue Date: 01-01-1998

Return Date:

Save This Form  
Select An Employee...  
Show selection  
Company Personal Property  
Company Automobile Information

### Inquiry/Select Mode

Equipment Number	Description	Category	Serial Number
01	Compaq LTE 5000	Computer	JJD78398DJ
02	HP Laserjet	Printer	D9273654A
03	X2 Technology	Modem	747H37

---

## NOTES

---

## Verify user-defined definitions

### Verify form and fields

After creating the data entry form program, it is necessary to verify that the program functions properly. This verification includes:

- Valid values can be entered into each field:
  - Date fields require data in YYMMDD or MM-DD-YY format for the US and Canada, DD-MM-YY elsewhere.
  - Numeric fields accept the proper number of integers and decimals. Results are displayed on the form properly.
  - Name fields require data in Last, First format.
  - Required fields must be entered, otherwise an error occurs.
  - Option list fields are edited against an option list.
  - Option list description fields display the code descriptions properly.
- Multiple-occurrence segment forms require the following navigation functions:
  - Inquiry/Select mode—(Actions|Select or right click |Show selection)
  - Display Next Page—(Actions|Next Page)
  - Display Previous Page—(Actions|Previous Page)
  - First Entry—(Actions|First Entry)

# Segment Layout Form (LSEGS)

```
0....+...10....+...20....+...30....+...40....+...50....+...60....+...70
LL101Compaq LTE 5000                00001JJ0783980J201L31
LL102HP Laserjet                    00002D9273654A 201L31
LL103X2 Technology                   00003747H37 201L31
L0100204AUSTIN, JULIANNE            382-67-4453 F234A05312972-0789F
L02002
L03002
L04001AUSTIN, JOHN L.                312339-8324
L050011321 VAN BUREN APT 392         CHICAGO, IL 60604
L06001STRUTHERS, DR. LOUIS           312843-0021
L070018 PINE GROVE                   CHICAGO, IL 60603
LPR200K18101 Hives                  1101BA-101-AUS 004010YWARH
LRD211J2907211J2903211J2701
LVG215K31060421511MAIN PLANT0540   0007215K28
LWA200203J07001Photograph           00000000010
LWA300203J07001Job Evaluation        00000000016
LZ7215A289 121 03 00015000215B0402
LZ8215B04171010PASS 81000PASS 82947LOW 93000PASS
LZA221A1525684-48 20 312971992331245418651121PDE1
LZB218H07D0200 2 Y021
```

---

## NOTES

---

## Verify user-defined definitions, continued

### Verify segment layout

The final verification step is to match your original hard copy segment layout to the physical layout of the segment. This is achieved through the use of the Segment Layout form (LSEGS).

The Segment Layout form (LSEGS) allows you to view all Human Resource and user-defined information, by segment, for an employee. This form can be used to check the layout of a particular segment code and its information.

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Employee Database Utilities
- Task:**  List Employee L Segments

**Result:** The Employee Selection Form displays.

Use the following key fields to display the L segment layout information.

- Number Field** Type an employee number. OR
- Name Field** Type an employee name.
- Choose OK or press Enter.

**Result:** The Segment Layout form (LSEGS) is displayed with a list of all the Human Resource and user-defined L segments for the employee number in the Key field.

## Create a FIND– Verb

```
MAKE A 'FIND CURRENT OCCURRENCE' VERB                                MKVERB

This screen will create a verb called FIND-CURRENT-Lxx-SEG
where xx is a specified segment code. The verb presumes
the first key field in the segment is a date in the
century-complement form.

Complete the text boxes below, then hit enter: X

If needed, alter Org Level: 999999

Enter SEGMENT-CODE: N9

Enter MODULE-CODE: PP

e.g. PR, BA, HR, SA, etc.
```

```
---Complete---
```

---

### NOTES

---

## Query-related programs

### Create a FIND– verb

The Create A FIND Verb program (MKVERB) enables you to create a FIND– verb. A FIND- verb locates the most recent occurrence of data in an L segment code (Pointer 36) when the first key field is a date in complement form.

When you use MKVERB to create a verb, The Solution Series assigns it a name using this general format:

FIND-CURRENT-Lxx-SEG, where xx is the segment code.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Fields and Verbs
<b>Task:</b>		Create a FIND Verb

**Result:** The Make a ‘Find Current Occurrence’ Verb form is displayed.

Use the following form fields to create a FIND Verb

<b>Segment Code</b>	Type the segment code for the segment, for example, L1.
<b>Module Code</b>	Type the segment’s module code.
	Choose OK or press ENTER.

**Result:** The ----COMPLETE---- message appears and the FIND-CURRENT-Lxx-SEG verb is created.

# Create the Segment Cross-Reference

Solution View Cross Reference Information

Enter the following cross reference elements

Segment: Company HED's

Non-key Data Name: HED - NAME

Find Current Occurrence Verb:

Read Table Verb:

All Key Fields For Segment: COMPANY - HED - NUMBER

Solution View is a Register

Save This Form  
Select An Employee...  
Show selection  
Solution View Table To Segment Cross Reference Information

---

## NOTES

---

## Query-related programs, continued

### Segment cross-reference

The Cross Reference Data Entry form (D-XREF) provides fields into which user-defined segment information must be keyed if the data is to be accessed by the Solution View (WRITER) program.

### Procedure to access D-XREF

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Fields and Verbs
<b>Task:</b>		Segment Cross Reference

**Result:** The Cross Reference Data Entry form (D-XREF) is displayed.

### Field description

There are four entry areas on the form:

- A NON-KEY DATA NAME—Identifies a non-key field for the user segment.
- FIND-CURRENT-OCCURRENCE verb—Identifies a unique verb for the multiple-occurrence segment.
- READ TABLE verb—Identifies a unique READ verb when creating a table cross-reference.
- ALL KEY FIELDS FOR SEGMENT—Identify all fields that have been defined as keys to the segment in key order.

# Create Cross-Reference Records

Solution View Cross Reference Information

Enter the following cross reference elements

Segment: Company HED's

Non-key Data Name:

Find Current Occurrence Verb:

Read Table Verb:

All Key Fields For Segment:

Solution View is a Registered

Save This Form  
Select An Employee...  
Show selection  
Solution View Table To Segment Cross Reference Information

---

## NOTES

---

## Query-related programs, continued

### Using D-XREF

To create the cross-reference records:

**Selection:**

**A Non-Key Data  
Name**

**Find-Current-  
Occurrence Verb**

**All Key Fields for  
Segment**

**Step:**

1. Type the name of a non-key field to the segment.
2. Type the name of the FIND-CURRENT-Lxx-SEG verb.
3. Type the name of all key fields for the segment.
4. Choose OK or press Enter.

**Result:** The form is returned with no form data or an error message.

## Section Summary

- **User-defined segments overview**
- **Data requirements analysis**
- **User-defined segment layout**

---

### NOTES

---

## Section summary

In this section, you learned the steps in creating user-defined segments. Specifically, you learned:

### **User-defined segments overview**

---

---

---

### **Data requirements analysis**

---

---

---

### **User-defined segment layout**

---

---

---

## **Section Summary, continued**

- **User-defined field definitions**
- **User-defined entry form**
- **Verify user-defined definitions**
- **Create a FIND– verb**
- **Create Solution View (WRITER) cross-reference records**

---

### **NOTES**

---

## Section summary, continued

### User-defined field definitions

---

---

---

### User-defined entry form

---

---

---

### Verify User-defined definitions

---

---

---

### Create a FIND- verb

---

---

---

### Create Solution View (WRITER) cross-reference records

---

---

---



## Section 5 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

Your company has decided to computerize the ordering of office supplies. The office supplies will be ordered by each employee using a data entry form. Later a report will be created for each employee, to be used as the order form for the supplies.

1. Create a form that displays the following fields and allows them to be updated.

Field	Nbr of Characters (Field Length)
ORDER DATE (CYYMDD)	6
ORDER NUMBER	6 (Alphanumeric)
BILL TO DEPARTMENT	30 (Alphanumeric)
OFFICE SUPPLY CODE	3 (Alphanumeric)
OFFICE SUPPLY DESCRIPTION	20 (Alphanumeric)
AMOUNT ORDERED	4
ORDER PRICE	9

### Key fields

- ORDER DATE
- ORDER NUMBER

### Title

Office Supply Order Form

### Option list validation

OFFICE SUPPLY CODE is to be validated against the Option List SCZZ created in the Section 4 Exercise.

### Employee name display

Display the employee name (in inquiry) at the top of the form.

2. Write a form program that will display the form you designed and create context-sensitive menu records for navigation to related forms.
3. Create a FIND- verb and a Solution View Cross-reference record for your new segment, if appropriate.

---

**NOTES**

---

## Section 6: Customization Basics Plus

---

### Table of Contents

Overview.....	175
Program processing.....	177
Form messages.....	179
Creating report parameter entry forms (RGMSTR) .....	193
Section summary.....	195
Section 6 exercise .....	197

## Objectives

- **Identify the operation of building a form and processing form input**
- **Create error messages and documentation**

---

NOTES

---

## Overview

### **Purpose**

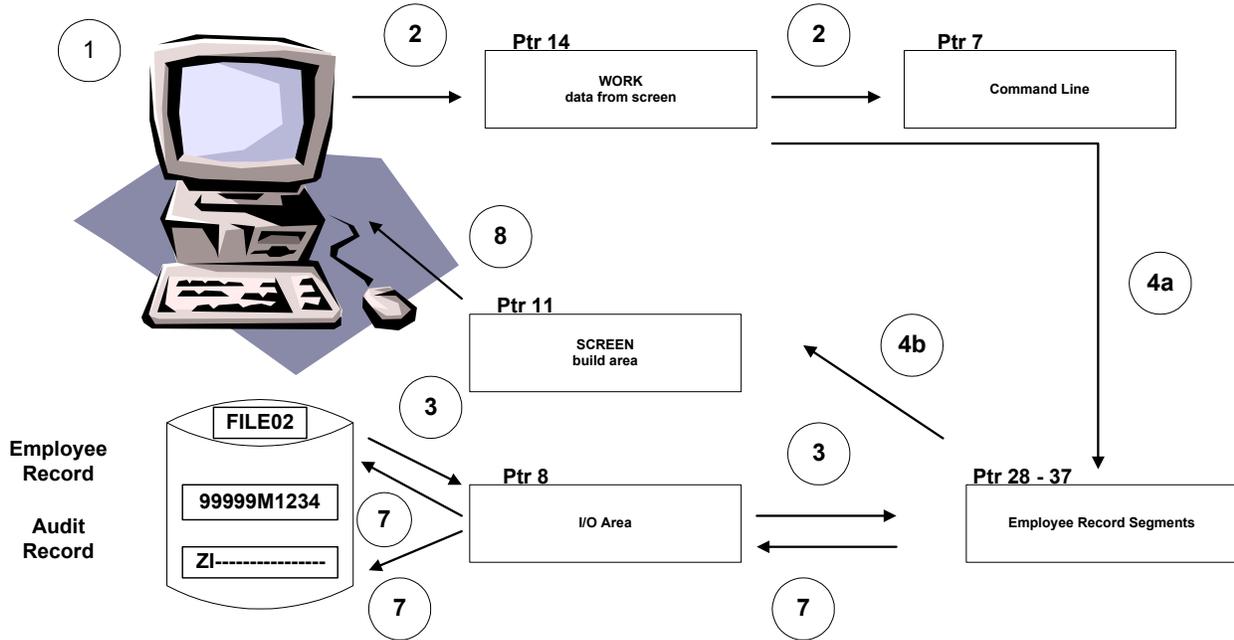
In this section, you will learn special form techniques.

### **Objectives**

When you complete this section, you will be able to:

- Identify the operation of building a form and processing form input.
- Create error messages and documentation.

# PROGRAM PROCESSING



## NOTES

## Program processing

1. User requests form by pressing Enter or clicking OK.
2. System moves the data entered on the form, including the command line, to WORK (Pointer 14).  
System saves the command line in Pointer 7.  
System transfers control to the requested online form program.
3. Form program gets employee record from FILE02 through Pointer 8 to the employee memory area and sets segment pointers 28–37.
4. For each entry field on the form:
  - a) Form section verbs take the data from WORK, if not blank, and updates the record fields:
    - Saves the key values.
    - Locates the correct segment based on the key values.
    - Inserts the segment if it does not exist.
    - Updates the RECORD-UPDATED field to indicate update.
    - Verifies Option List values, inserts leading zeros.
    - CALLs field edit routines.
  - b) Form sections move data from record fields to SCREEN (Pointer 11).
5. If updates were entered and no system errors were found, the online program's relational edits are processed.
6. Control is RETURNed to CYB90x.
7. If no errors exist and RECORD-UPDATED='Y', the FILE02 record is updated and an Audit record (ZI/ZZ) is created.
8. The (re)built form is sent to the terminal from SCREEN (Pointer 11).

## Form Messages

- **Reject**
- **Warning**
- **Message**

---

NOTES

---

## Form messages

### **Description**

Cyborg Scripting Language (CSL) enables you to perform simple and complex relational editing to verify data on input. If a user enters incorrect or illogical data, the system issues one of these three kinds of error messages:

### **Reject (R)**

Indicates errors that the user must correct before the system will process the form. The user can choose to quit the form without updating by pressing CANCEL or by entering a Q in the action field in the command dialog box.

### **Warning (W)**

Indicates errors that stop processing. The user can choose to correct the error or to accept the data by pressing 'Yes' on the Override Warning dialog OR by entering an A in the action field in the command dialog box.

### **Message (M)**

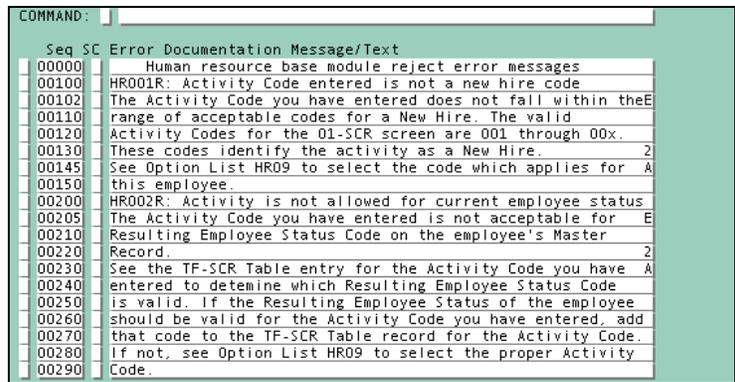
Provides information only. These messages do not require the user to take any action.

This topic describes:

- Creating and documenting messages.
- Displaying messages.

# Creating Error Messages Using EDIT

Record Group Name Format:	ERRxy	
Where:	ERR	Is a literal value
	xx	Specifies the module code
	y	Identifies the message type: R, W, or M



## NOTES

## Form messages, continued

### Creating messages

You may use existing Cyborg-delivered messages, add to Cyborg-delivered message groups, or create your own message groups. The Edit Utility form (EDIT) is used to access the record group with an Object of Error Messages (P/R) and an error record group of ERRxxy.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		Programming Utilities
<b>Task:</b>		Edit Control Repository Objects

**Result:** The Edit Utility form (EDIT) is displayed.

Use the following Edit Utility form (EDIT) fields to create an option list

<b>Object</b>	Select Error Messages (Object=P/R).
<b>Object Key</b>	Type the error message record group to be edited in ERRxxy format. Choose OK or press Enter.

**Result:** The Option List Edit form is displayed.

### EDIT columns

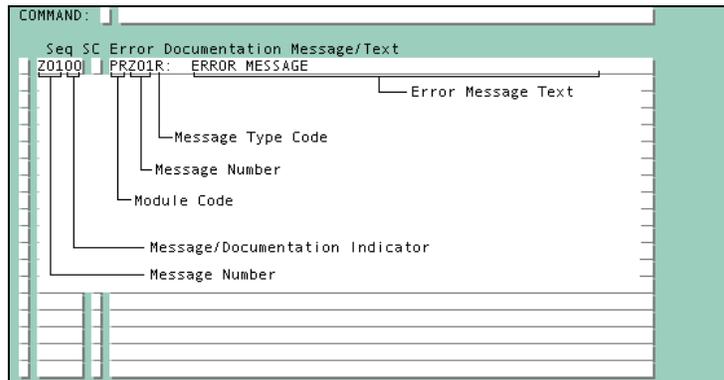
The Edit Utility form (EDIT) for error messages is in a different format than the source code. The column definitions are:

- Line Command  
Specifies whether you are adding, changing or deleting the line.
- Seq  
The Sequence Number is used to identify the error message and associated error documentation.
- SC  
The Sequence Code is used by the system; no entry should be made here.
- Error Documentation Message/Text  
Used to enter the error message and error documentation.

### Alternate language messages

Alternate language messages may be created with EDIT by using the Alt Lang Error Msgs object (Object = P/Q) instead of Error Messages (Object = P/R).

# Error Message



Error Message Format:            xxnnny

Where:            xx            Specifies the module code  
                  nnn          Identifies the message's Edit sequence number, which must end in 00:  
                  y            Specifies the message type: R, W, or M.

---

## NOTES

---

## Form messages, continued

### Error messages

Error messages are displayed by special CSL verbs that use the data in the Seq (Sequence Number), and Error Documentation Message/Text columns to uniquely identify each error message.

### Sequence number convention

The sequence number conventions are:

- The first 3 positions represent the message number.
- The last 2 positions represent the Message/Documentation Indicator. If the last 2 positions = 00, the Error Documentation Message/Text column contains the message, otherwise it contains the message documentation.

### Message conventions

The Error Documentation Message/Text error message conventions are:

- The first 6 positions contain the Message ID, which is made up of a Module Code, Message Number, and Message Type Code as detailed above. The Message IDs Message Number must be the same as the first 3 positions of the SEQ field.
- The 7th through 8th positions must contain a colon (:) followed by one space.
- The 9th through 60th positions contain the message text. Message text should be helpful to your users. Often a short, one-line message can convey enough information to enable users to correct an error without having to request more documentation.

Additions to the Cyborg-delivered message group are assigned sequence numbers M00 through Z99.

# Error Message Documentation

The screenshot shows a window titled "COMMAND:" containing a table of error messages. The table has four columns: "Seq", "SC", "Error Documentation Message/Text", and a final column with single characters. Annotations with arrows point to specific parts of the table: "Message Number" points to the "Seq" column, "Message/Documentation Indicator" points to the "SC" column, "Message Documentation" points to the "Error Documentation Message/Text" column, and "Headings/Carriage Control" points to the final column.

Seq	SC	Error Documentation Message/Text	
00102		Message Explanation...	E
00104		Message Explanation continued...	2
00106		Recommended Action...	A
00108		Recommended Action continued...	

---

## NOTES

---

## Form messages, continued

### Documenting messages

Most messages have accompanying documentation that explains the message more fully and recommends the action the user should take. This documentation resides below the message text on lines with sequence numbers that share the same first three digits as the message, but end with numbers from 01 to 99.

### Sequence number convention

The Sequence Number conventions for message documentation are:

- The first 3 positions represent the Message Number.
- The last 2 positions represent the Message/Document Indicator that must have a value from 01 to 99 for the message documentation.

### Message documentation conventions

The Error Documentation Message/Text error documentation conventions are:

- The first 59 positions may contain the error message documentation text.
- The 60th position is reserved for Headings/Carriage Control codes.
  - Heading Codes—E indicates the start of the explanation; A indicates the start of the recommended action.
  - Carriage Control—blank tells the system to single space; a value of 2 tells the system to double space after the associated line.

# Error Message Processing

```
SECURITY.    @ Sample Program
P100-START-SCREEN.
  KEY-REQUIRED.
  UPDATE-EMPLOYEE.
  NEW-SCREEN-STYLE.
  SCREEN-SECTION '0'.
  IF INQUIRY-MODE OR SELECTION-MODE
    GO TO P990-INQUIRY-SCREEN.
P200-ENTRY-SCREEN.
  SCREEN-SECTION '1'.
P300-VERIFY.
  SET-FOR-MESSAGES.
  IF ERRORS-EXIST RETURN.
  IF RECORD-NOT-UPDATED
    RETURN.
  CALL 'XXXXXX'. @ EDIT ROUTINES
  RETURN.
P990-INQUIRY-SCREEN.
  SCREEN-SECTION '8'.
  RETURN.
```

---

## NOTES

---

## Form messages, continued

### Displaying messages

To display a message on the user's terminal you must add coding to your program to:

- Set the placement of the message
- Check for system errors
- Check for user input into the form
- Display the message

### Message placement

The SET-FOR-MESSAGES verb is used to determine where error messages should begin on the form and positions the FORM pointer 11 past the last line.

### Check for system errors

The second step in coding for error messages is to verify that no system level errors have been found. The ERRORS-EXIST Boolean expression checks the value of the SCREEN-ERROR field to be not equal to an F. This field is initially set to F before the form is painted. If an error is found on the form, the value is changed.

### Check for user input

The RECORD-NOT-UPDATED Boolean expression checks the value of the RECORD-UPDATED field for a blank. This field is initially set to a blank when a form is painted. If any data is detected by an ENTRY verb, it is set to Y.

# Error Message Subroutine

## Reject

Format

`PRINT-REJECT 'xx999'.`

## Warning

Format

`PRINT-WARNING 'xx999'.`

## Message

Format

`PRINT-MESSAGE 'xx999'.`

## Form Message—Error

```
Form Message - Error: SECURITY. @ Error Subroutine
IF FIELD-4 EQUALS '00' OR '01' AND
  FIELD-W IS GREATER THAN :1000.00
  PRINT-REJECT 'PRZ01' RETURN.
IF FIELD-X IS LESS THAN :10
  PRINT-WARNING 'PRZ01' RETURN.
```

---

## NOTES

---

## Form messages, continued

### Display message

The final step in coding for error messages is to create a subroutine program that will test for and display the message. Message statements must include one of the following verbs, depending on the type of message you want to display:

- PRINT-REJECT  
To access and display reject and file error messages.
- PRINT-WARNING  
To access and display warning messages.
- PRINT-MESSAGE  
To access and display memo messages.

The message verb identifies the kind of message you want displayed.

The message ID specifies the message you want displayed. Enclose this literal value in single quotation marks ('...'). You do not have to include the message type (M, R, W) in your statement.

*HINT: Because the number of form lines is limited, it may not be possible to display all messages at once. Therefore, it is recommended that you display messages in groups by severity order and require correction of the errors in a group before proceeding to the next group, for example, Rejects, then Warnings, then Messages.*

# Field Error Subroutine

```

Example 1: Salary maximum error
SECURITY. @ Salary field edit subroutine
IF W8-08-240 GREATER THAN '09999999'
MOVE '@' TO SCREEN-ERROR
MOVE '@' TO W8-01-480
MOVE '@PRRZ01' TO W8-07-330.
RETURN.
    
```

Result: PRZ01R: Salary cannot be greater than 99,999.99.

Field Message Format: @xxRnnn

Where: @ = at-sign

      xx = Specifies the module code

      y = Specifies the message type: R

      nnn = Identifies the message's number

---

## NOTES

---

## Form messages, continued

### Field error subroutine

As you recall, when defining a field you may specify a field edit routine that is used to provide additional editing.

- The subroutine is automatically called when new data is entered into the form in that field.
- System edits are performed before the field-edit routine is executed.

### Work fields

Several Pointer 8 work fields are used to test and display an error message. These work areas include:

- W8-30-240  
Contains the data from the form field after it has been converted to its storage format.
- W8-30-270  
Contains the data as it was typed in the form field.
- W8-30-300  
Data as originally displayed on the form before the change. This field is in edited format.
- W8-07-330  
Error Message Number. Place the error message number into this field to identify which error message to display. The format of this field is @MMR999. Only reject messages are valid.
- W8-01-480  
Error Message Flag. Place an at-sign (@) into this work area to signal a field reject error to the system.
- SCREEN-ERROR (W7-01-093)  
Form Error Flag. Place an at-sign (@) into this work area to signal a form reject error to the system.

# Creating Batch Report (RGMSTR) Parameter Entry Forms

Report Group Activities

ADRIEN	FSA reports
AT0001	Applicant Tracking Test Reports 1
AT0002	Applicant Tracking Test Reports 2
AT0003	Applicant Tracking Test Reports 3
BA0001	Benefits Administration Test Reports 1
BA0002	Benefits Administration Test Reports 2
BA0003	Benefits Administration Test Reports 3
BA0004	Benefits Administration Test Reports 4
BA0005	Benefits Administration Test Reports 5
BA0006	Benefits Administration Test Reports 6
BATABL	Benefits Admin Table Reports 1
ER0001	Employee/Labor Relations Test Reports
HR0001	HRMS Base Test Reports 1
HR0002	HRMS Base Test Reports 2
HR0003	HRMS Base Test Reports 3
HR0004	HRMS Base Test Reports 4
HRTABL	HRMS Base Table Reports
HS0001	Health & Safety Test Reports
JRE998	Testing Parameter Screens in Build

Buttons: Add, Edit, Delete, Reports, Companies, Org Unit, Parameters

Parameter Selection For Benefits Administration Test Reports 1 BA0001

2E-RPT	Adp Earnings Report	<input type="checkbox"/>	Set Parameters
40-RPT	Calc Credited Serv Granted	<input type="checkbox"/>	Set Parameters
41-RPT	Calc Vesting Earned	<input type="checkbox"/>	Set Parameters
42-RPT	Vested Employees By Percent Vested	<input type="checkbox"/>	Set Parameters
48-RPT	Retiree Payment Stop Date Passed	<input type="checkbox"/>	Set Parameters
49-RPT	Retiree Deceased	<input type="checkbox"/>	Set Parameters
4A-RPT	Welfare Benefits Register	<input type="checkbox"/>	Set Parameters
4B-RPT	Plan Participation Register	<input type="checkbox"/>	Set Parameters
4C-RPT	Plan Beneficiary Listing	<input type="checkbox"/>	Set Parameters
4D-RPT	Plan Dependent Listing	<input type="checkbox"/>	Set Parameters

Report Parameters For Adp Earnings Report 2E-RPT

Report Group - Benefits Administration Test Reports 1 BA0001

ADP Date: 12-31-1989

Plan ID: 500

## NOTES

## Creating report parameter entry forms (RGMSTR)

### Reporting parameter selection

Report run-time parameters are entered using an online parameter entry form. To access the parameter entry form:

Make the following selections from the Navigator:

<b>Component:</b>		Reporting
<b>Process:</b>		Report Scheduling
<b>Task:</b>		Schedule Report Groups

**Result:** The Report Group Activities form displays.

- Select a report group on the Schedule Report Groups form (RGMSTR) and push the parameters button. The Parameter Selection form (RPARMS) displays the names of all the reports in the report group with those needing online parameters having a Set Parameters button to the right of them.
- Select the Set Parameters button to display a Report Parameters entry form (RPTARG) for that report.
- Enter the parameters for the report. The parameters will be stored in Pointer 6 with all entered dates in century complement format.

### Parameter entry form development

The parameter entry form (RPTARG) is a combination of a standard form program (RPTARG) containing the header report title and report group name fields and a CALLED parameters entry form program containing the parameter entry fields for a particular report. There are several standard parameter entry forms delivered with the system. They are named GRPRSx. Additional parameter entry forms may be developed for new reports using the naming convention of R-xxxx where xxxx is the first four positions of the report name that is, R-X1-R).

The steps in developing an R-xxxx Parameter Entry form are:

- Create parameter entry field definitions pointing to W6-36-036.
- Create an R-xxxx form using Section 1 to contain only the parameter entry fields defined in Pointer 6.
- Include the Parameter Entry form name on line 00002 in the Batch Report extract program as: 00002 @PARMS=R-xxxx

☞ *Further information on creating Batch Reports is covered in the Cyborg Scripting Language Report Customization documentation.*

## Section Summary

- **Form processing**
- **Form messages**

---

### NOTES

---

## Section summary

In this section, you learned several techniques to manipulate forms. Specifically, you learned:

### Form Processing

---

---

---

---

### Form Messages

---

---

---

---

## Section 6 Exercise

---

**NOTES**

---

## Section 6 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. Add code to your Office Supply Order form for the program to perform the following functions:
  - Use a field error subroutine to cause a field error reject message to appear when an Order Number of all zeros or spaces is entered. Use reject message group ERRSCR, message number Z00. Include explanation and recommended action documentation.
  - Include code in your program to cause a warning message to appear when more than 100 Post-it Pads are ordered. Use warning message group ERRSCW, message number Z00. Include explanation and recommended action documentation.

---

**NOTES**

---

## Section 7: Online Query Programming

---

### Table of Contents

Overview.....	201
Query programming.....	203
Maintaining alternate keys.....	207
Section summary.....	221
Section 7 exercise .....	223

## Objectives

- **Identify query verbs**
- **Create a query program**
- **Create user-defined alternate keys**

---

## NOTES

---

## Overview

### **Purpose**

In this section, you will learn how to create a query program and use query-related utilities.

### **Objectives**

When you complete this section, you will be able to do the following:

- Identify query verbs
- Create a query program
- Create user-defined alternate keys

# QUERY Related Verbs

READ-COMPANY.

READ-EMPLOYEE.

READ-TAXES.

NEXT-LINE.

PRINT { literal  
field-name-1 } { literal  
field-name-2.. }

INQUIRY field-name-1 { literal...  
field-name-2. }

FIND non-key-field-1 { FROM HERE  
STARTING WITH { literal  
field-name-1 } }

PROCESS non-key-field-1 { FROM HERE  
STARTING WITH literal  
field-name-1  
ENDING WITH literal  
field-name-2 }  
imperative statement ...  
END-PROCESS

---

## NOTES

---

## Query programming

### Query programs

QUERY programs allow you to view data online for the selected Master Records. The QUERY program Command Line lets you enter a range of values, instead of just one value. For example, all employees between 0001 and 9999, instead of just one employee number.

### Query verbs

QUERY programming uses several of the verbs with which you are already familiar. These verbs include:

- The READ– verbs must be used to access records. UPDATE is not used in Query programs.
- The INQUIRY and PRINT verbs are used to build the output line for each employee record.
- The FIND– and PROCESS/END–PROCESS verbs are used to point to a particular occurrence of a multiple-occurrence segment.
- The NEXT–LINE verb is used just as it is in online form programming.

# QUERY-Only Verbs

**Format:**

QUERY-ONLY.

QUERY-FIRST-PASS.

TIME-TO-PRINT-TITLE.

QUERY-HEADERS.

**Query example using FIND:**

```
READ-EMPLOYEE.  
IF TIME-TO-PRINT-TITLE  
  PRINT 'QUERY TITLE' NEXT-LINE NEXT-LINE.  
QUERY-HEADERS.  
FIND AMOUNT-ONE STARTING WITH '500'.  
MATCH-SEGMENT-TYPE.  
IF FOUND  
  INQUIRY EMPLOYEE-NUMBER EMPLOYEE-NAME-20  
HED-AMOUNT-MTD  
  HED-AMOUNT-QTD HED-AMOUNT-YTD  
NEXT-LINE.
```

**Query example using PROCESS/END-PROCESS:**

```
READ-EMPLOYEE.  
IF TIME-TO-PRINT-TITLE  
  PRINT 'QUERY TITLE' NEXT-LINE NEXT-LINE.  
QUERY-HEADERS.  
PROCESS AMOUNT-ONE STARTING WITH '500'.  
  INQUIRY EMPLOYEE-NUMBER EMPLOYEE-NAME-20  
HED-AMOUNT-MTD  
  HED-AMOUNT-QTD HED-AMOUNT-YTD.  
NEXT-LINE.  
END-PROCESS.
```

---

## NOTES

---

## Query programming, continued

### Query verbs

In addition to the verbs you have learned already, there are several special verbs used only with Query programs. These verbs include:

#### **QUERY-ONLY**

Instructs the system to inhibit execution of the program to only the QUERY Facility. This statement should be the first command coded in a Query program.

#### **QUERY-FIRST-PASS**

Used with a conditional IF statement to determine the timing for initialization or any preprocessing tasks.

#### **TIME-TO-PRINT-TITLE**

Used with a conditional IF statement to determine the timing for printing the Query Title at the top of each form.

#### **QUERY-HEADERS**

Instructs the program to use default heading information for the fields as defined on the Field Name Table. If used, this verb must be specified before any fields are displayed on the form.

The headings are not painted until the INQUIRY verb is used.

# Maintaining Alternate Keys

## QUERY Primary Keys

00	Employee Number
HL	History/Labor
MN	Documentation
P	Program Name
TX	Tax Records
ZZ	Audit Record

## QUERY Alternate Keys

01	Social Security Number
02	Employee Name
??	Your User-defined Keys

---

## NOTES

---

## Maintaining alternate keys

### Query keys

Keys have been defined by Cyborg and are available for your use with query. Two groups of QUERY keys are:

- Primary
- Alternate

### Primary keys

Primary keys are used to direct your query program to the type of record for processing in standard sequence, for example, employee data versus Audit Record data. The primary keys are available for immediate use and require no additional maintenance.

### Alternate keys

Alternate key records are stored on the system control repository (FILE01) to allow access to your data in a sequence other than the standard record key order.

- When a query is executed using an alternate key, the system must:
  - Read the alternate key record from the system control repository (FILE01).
  - Read the employee Record from the Employee Database (FILE02) using the Control 1-2 and employee number from the alternate key record.
- Alternate keys are not maintained by the system and must be rebuilt periodically.
- User-defined alternate keys may be created and maintained using Cyborg Scripting Language.



## Maintaining alternate keys, continued

### User-defined alternate keys

You may create user-defined alternate keys by modifying the Build Alternate Keys program (KEY-00), which is used to build the alternate key index on FILE01.

- Any field on the Employee Database may be used as an alternate key value.
- The alternate key index allows for 100 duplicate occurrences of the same alternate key value.

### Modify KEY-00

Add new coding to the Build Alternate Keys program (KEY-00) for each alternate key you will be defining. The coding instructions are included in KEY-00 and are as follows:

- Remove the leading @ (at-signs) and add sequence lines 00400 through 90000 with new sequence numbers 00480 through 89990.
- Replace the XX in sequence line 00440 with a two-character identifier for the alternate key.
- Replace the XXXXXXXXXXXXX in sequence line 00460 with the first field to be used as an alternate key value.
- Repeat sequence line 00460 for each additional field that will be used as a composite alternate key value. The total alternate key value may not exceed 19 positions.

*Note: It may be necessary to position the segment pointer using FIND at the appropriate segment occurrence for multiple-occurrence segment fields.*

*Note: These program changes do not take effect until the Build Alternate Keys program (KEY-00) is RELOADED.*

# Modifying Query for User-Defined Alternate Keys

**Unmodified** 05400 @ TO RESTRICT QUERY TO ONLY THE CURRENT (TOP LINE)  
05500 @ CONTROL 1-2, REMOVE THE AT-SIGNS IN THE NEXT 3 LINES  
05600 @ AND RELOAD QUERY.  
05700 @ IF W7-06-240 NOT EQUAL TO W6-06-026  
05800 @ CALCULATE QUERY-FROM-DUP + :1 = QUERY-FROM-DUP  
05900 @ GO TO P150-READ.  
56000 :  
56100 :  
37100 P980-MENU.  
37200 SPACE-OVER :18. PRINT '00 EMPLOYEE-NUMBER'. NEXT-LINE.  
37300 SPACE-OVER :18. PRINT '01 SOCIAL SECURITY NUMBER'.  
37400 NEXT-LINE. SPACE-OVER :18. PRINT '02 LAST NAME, FIRST'.  
37500 NEXT-LINE. SPACE-OVER :18.  
37600 PRINT 'SEE DOCUMENTATION FOR OTHERS'. NEXT-LINE.  
37700 @@@@ADD USER ALTERNATE KEY DESCRIPTIONS AFTER THIS LINE@@@@

05400 @ TO RESTRICT QUERY TO ONLY THE CURRENT (TOP LINE)  
05500 @ CONTROL 1-2, REMOVE THE AT-SIGNS IN THE NEXT 3 LINES  
05600 @ AND RELOAD QUERY.  
05700 IF W7-06-240 NOT EQUAL TO W6-06-026  
05800 CALCULATE QUERY-FROM-DUP + :1 = QUERY-FROM-DUP  
05900 GO TO P150-READ.  
06000 :  
**Modified** 06100 :  
37100 P980-MENU.  
37200 SPACE-OVER :18. PRINT '00 EMPLOYEE-NUMBER'. NEXT-LINE.  
37300 SPACE-OVER :18. PRINT '01 SOCIAL SECURITY NUMBER'.  
37400 NEXT-LINE. SPACE-OVER :18. PRINT '02 LAST NAME, FIRST'.  
37500 NEXT-LINE. SPACE-OVER :18.  
37600 PRINT 'SEE DOCUMENTATION FOR OTHERS'. NEXT-LINE.  
37700 @@@@ADD USER ALTERNATE KEY DESCRIPTIONS AFTER THIS LINE@@@@  
37710 SPACE-OVER :18. PRINT 'CC CITIZENSHIP CODE'. NEXT-LINE.

---

## NOTES

---

## Maintaining alternate keys, continued

### Modify QUERY

Add new coding to the QUERY program for each alternate key you will be defining. This additional coding will:

- Display the query alternate key two character identifier and description on the QUERY form.
- Optionally, limit Control 1-2 access using alternate keys.

To display the Query alternate key identifier and description:

- Edit the code in paragraph P980-MENU to include the two-character Query alternate key identifier and description.

To restrict Control 1-2 access using query alternate keys to the Control 1-2 currently in the command line:

- Remove the at-signs (@) from the indicated code in paragraph P150-READ.

*Note: These program changes will not take effect until QUERY is RELOADED.*

## The Solution Series Alternate Key Utilities

- **Delete alternate keys (KEYDEL)**
- **Build alternate keys (KEY-00)**

---

### NOTES

---

## Maintaining alternate keys, continued

### Alternate key utilities

Since alternate keys represent an "as of" situation, new master records do not yet have alternate key records. For this reason, alternate key records must be periodically purged and rebuilt. Cyborg suggests that you set up a schedule for maintaining these records.

If new master records (employees) have been added, but the alternate key records have not been updated, the only way to view the new records in QUERY is to use one of the primary keys.

The Solution Series utility programs required to maintain your alternate keys are:

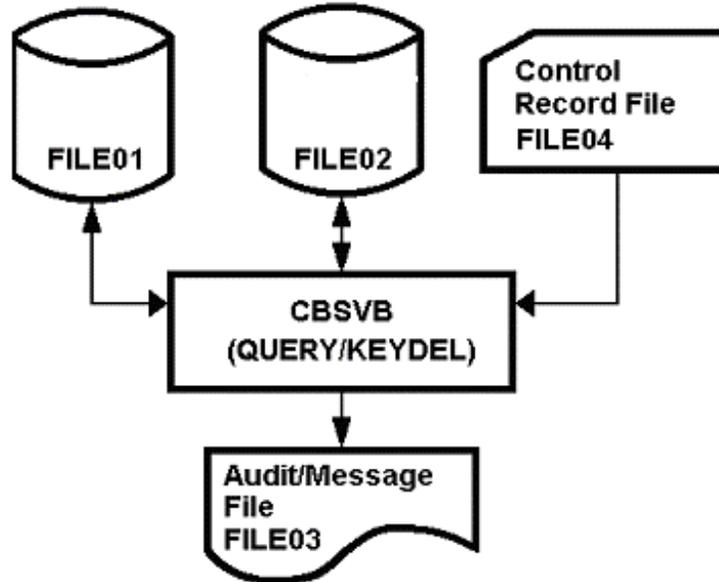
- Delete alternate keys (KEYDEL)
- Build alternate keys (KEY-00)

### Maintaining alternate keys in batch

Because you are dealing with a large number of records, it is recommended that you execute the Build Alternate Keys (KEY-00) and Delete Alternate Keys (KEYDEL) programs in batch mode.

To execute alternate key maintenance in batch mode, you must use a query control record in your JCL.

# Alternate Key Deletion (KEYDEL)



**Input Files:** FILE01 System Control Repository File  
 FILE02 Employee Database  
 FILE04 Control Record File

**Execute:** CBSVB

**Output files:** FILE03 Audit/Message File

## QUERY Control Record:

### Social Security Number Delete:

```

      1  1  2  2  3  3  4  4  5  5  6  6  7  7  8
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0
P QUERY LINE1  999999QUERY                                KEYDEL010          *
P QUERY LINE2                                9999999999
    
```

### Employee Name Delete:

```

      1  1  2  2  3  3  4  4  5  5  6  6  7  7  8
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0
P QUERY LINE1  999999QUERY                                KEYDEL02A          *
P QUERY LINE2                                ZZZZZZZZZZ
    
```

---

## NOTES

---

## Maintaining Alternate Keys, continued

### **KEYDEL**

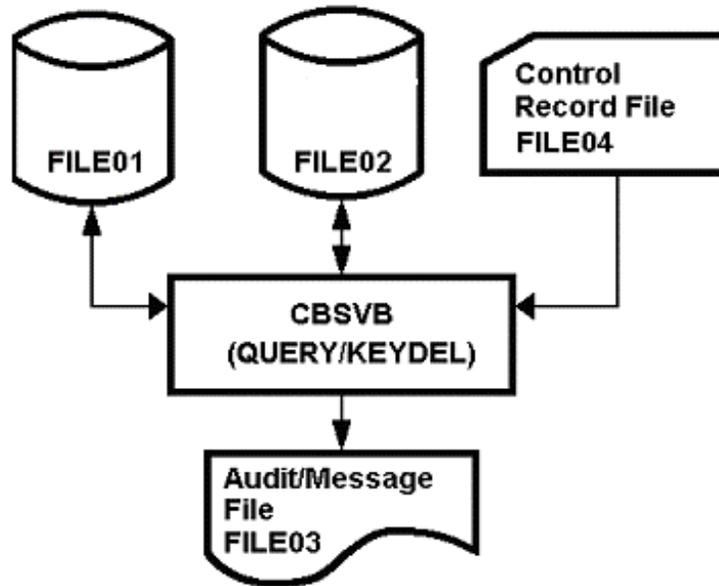
The Delete Alternate Keys program (KEYDEL) deletes alternate key records from the system control repository by alternate key type and Control 1-2. Once the records are deleted, new alternate key records can be (re)built.

### **QUERY Control Record for KEYDEL**

The Delete Alternate Keys program is executed utilizing the QUERY facility and requires a two-line control record:

- Control records for each alternate key type must be added to the KEYDEL job.
- Each QUERY control record must have the appropriate FROM and TO values to match the alternate key being deleted.

# Alternate Key Build (KEY-00)



Input Files: FILE01 System Control Repository File  
 FILE02 Employee Database  
 FILE04 Control Record File

Execute: CBSVB

Output files: FILE03 Audit/Message File

## QUERY Control Record:

### Rebuild for Control 1-2 999999:

```

      1 1 2 2 3 3 4 4 5 5 6 6 7 7 8
    ....5....0....5....0....5....0....5....0....5....0....5....0....5....0
P QUERY LINE1 999999QUERY KEY-000000 *
P QUERY LINE2 9999999999
  
```

### Rebuild for Control 1-2 991111:

```

      1 1 2 2 3 3 4 4 5 5 6 6 7 7 8
    ....5....0....5....0....5....0....5....0....5....0....5....0....5....0
P QUERY LINE1 991111QUERY KEY-000000 *
P QUERY LINE2 9999999999
  
```

---

## NOTES

---

## Maintaining alternate keys, continued

### **KEY-00**

The Build Alternate Keys program (KEY-00) (re)builds alternate key records that supply data for your Query programs. As delivered, the build alternate keys program creates alternate key records by Control 1-2 for each Employee Database record (FILE02) on file.

### **QUERY control record for KEY-00**

The Build Alternate Keys program is executed utilizing the QUERY facility and requires a two-line control record:

- Control records for each Control 1-2 value must be added to the KEY-00 job.
- The primary KEY of 00 must be used with the appropriate FROM and TO values to include ALL employees.

# Using Alternate Keys with QUERY

```
Query XINTRO Key CC From  To ZZ
00 Employee Number
01 Employee Soc Sec Nbr
02 Employee Name
   See Documentation for Others
```

---

## NOTES

---

## Maintaining alternate keys, continued

### Using alternate keys

Alternate keys are available to be used with the QUERY program as soon as they are rebuilt using the Delete Alternate Keys (KEYDEL) and Build Alternate Keys (KEY-00) programs.

Make the following selections from the Navigator:

**Component:**  User Tools  
**Process:** User Tools  
**Task:**  Run a Query

**Result:** The Query form is displayed.

To execute a QUERY program using an alternate key:

- Identify the alternate key by typing the two-character identifier in the QUERY KEY field.
- Designate beginning and ending ranges by typing a beginning value in the QUERY FROM field and an ending value in the QUERY TO field.

## Section Summary

- **Query-related programs**
- **QUERY programming**
- **Maintaining alternate keys**

---

### NOTES

---

## Section summary

In this section, you learned several techniques for programming and using Query programs. Specifically, you learned:

### Query-related programs

---

---

---

### QUERY programming

---

---

---

### Maintaining alternate keys

---

---

---

# Section 7 Exercise

---

**NOTES**

---

## Section 7 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. Using the Office Supply form data, do the following:
  - Create a Query program to list the following information:
    - Title: Office Supply Query
    - Fields: Employee Name (length of 10 positions)
    - Order Date
    - Order Number
    - Bill to Department (length of 20 positions)
    - Office Supply Code
    - Amount Ordered
    - Price
  - Run the Office Supply Query program.

---

**NOTES**

---

## Appendix A: Exercise Answers

---

### Table of Contents

Section 2 exercise 1 .....	226
Section 2 exercise 2 .....	229
Section 3 exercise .....	233
Section 4 exercise .....	234
Section 5 exercise .....	237
Section 6 exercise .....	245
Section 7 exercise .....	247

## Section 2 exercise 1

### Purpose

Design a form that will display the following fields and allow them to be updated.

1. Use the title **Personal and Identification Information**

The screenshot shows a software window titled "Personal and Identification Information". The window has a menu bar with "File", "Screen", "Mode", and "Add Control". Below the menu bar is a toolbar with icons for undo, redo, and other actions. The form content includes:

- Record Date: XXXXXXXXXXXX
- Marital Status: XXXXXXXXXXXXXXXXXXXX (dropdown)
- Dependents: XX (dropdown)
- Citizenship: XXXXXXXXXXXXXXXXXXXX (dropdown)
- Prior Name: XXXXXXXXXXXXXXXXXXXX
- Identification section containing:
  - Verified: XXXXXXXXXXXXXXXXXXXX (dropdown)
  - Provided: XXXXXXXXXXXXXXXXXXXX (dropdown)

At the bottom of the window, a status bar shows "List Box", "Row: 13", "Col: 56", "1", and "Section 1".

2. Display the employee name in the form heading.

### Form Header - Title

The screenshot shows a "Text" dialog box with the following configuration:

- Text: Personal and Identificat
- When Shown:
  - Always
  - Character Mode
  - Graphical Mode
- Section: 0

Buttons: OK, Delete, Cancel

The screenshot shows a "Display Box" dialog box with the following configuration:

- Field Name: EMPLOYEE-NAME
- Label: (empty)
- Exit Routines:
  - Before: (empty)
  - After: (empty)
- When Shown:
  - Always
  - Character Mode
  - Graphical Mode
- Label Location:
  - No Label
  - Left, Left Justified
  - Left, Right Justified
  - Above, Left Justified
  - Above, Centered on 2 Lines
- Section: 0

Buttons: OK, Delete, Cancel

## Section 2 exercise 1, continued

3. Display the following fields and allow them to be updated.

### RECORD-DATE

Text Box

Field Name: RECORD-DATE  Spin Button?

Label: Record Date

Section: 1

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

When Shown

- Always
- Character Mode
- Graphical Mode

Current Value

- Don't Show
- Show Inside Box
- Show Below Box

Exit Routines

Before:

After:

OK Delete Cancel

### MARITAL-CODE

List Box

Field Name: MARITAL-CODE  Big Codeset

Label: Marital Status

Section: 1

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

When Shown

- Always
- Character Mode
- Graphical Mode

Exit Routines

Before:

After:

OK Delete Cancel

### TOTAL-DEPENDENTS

Text Box

Field Name: TOTAL-DEPENDENTS  Spin Button?

Label: Dependents

Section: 1

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

When Shown

- Always
- Character Mode
- Graphical Mode

Current Value

- Don't Show
- Show Inside Box
- Show Below Box

Exit Routines

Before:

After:

OK Delete Cancel

### CITIZENSHIP-CODE

List Box

Field Name: CITIZENSHIP-CODE  Big Codeset

Label: Citizenship

Section: 1

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

When Shown

- Always
- Character Mode
- Graphical Mode

Exit Routines

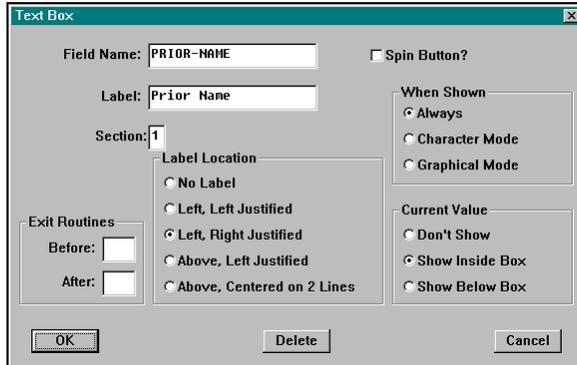
Before:

After:

OK Delete Cancel

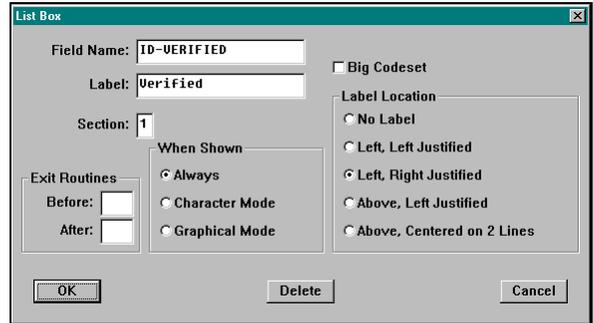
## Section 2 exercise 1, continued

**PRIOR-NAME**



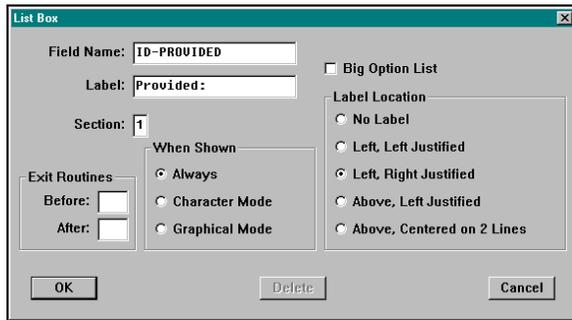
Text Box dialog for PRIOR-NAME. Field Name: PRIOR-NAME, Label: Prior Name, Section: 1. Includes options for Label Location, When Shown, Exit Routines, and Current Value.

**ID-VERIFIED**



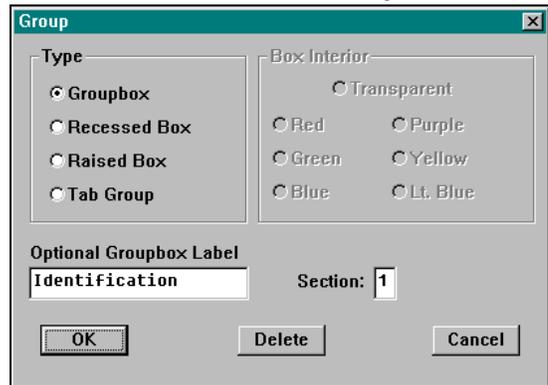
List Box dialog for ID-VERIFIED. Field Name: ID-VERIFIED, Label: Verified, Section: 1. Includes options for Big Codeset, Label Location, When Shown, and Exit Routines.

**ID-PROVIDED**



List Box dialog for ID-PROVIDED. Field Name: ID-PROVIDED, Label: Provided:, Section: 1. Includes options for Big Option List, Label Location, When Shown, and Exit Routines.

**Identification Group Box**



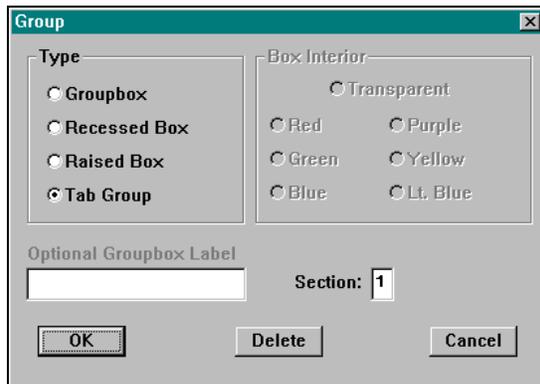
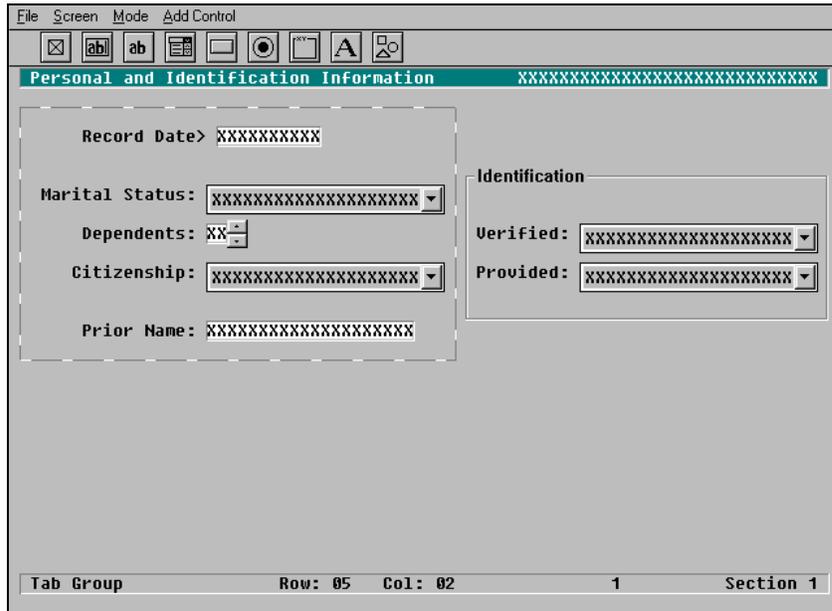
Group dialog for Identification Group Box. Type: Groupbox, Box Interior: Transparent. Includes an Optional Groupbox Label: Identification, Section: 1.

## Section 2 exercise 2

### Purpose

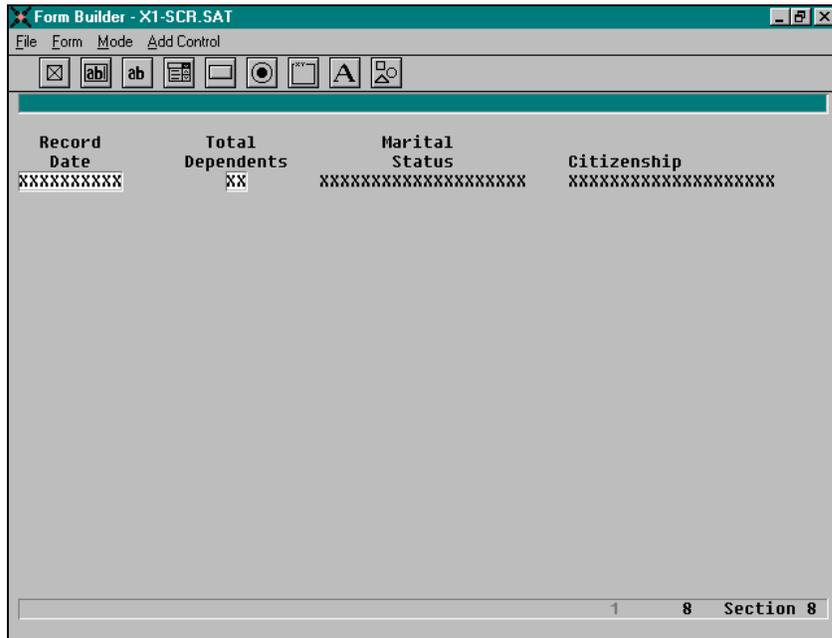
This exercise gives you practice using the information you have learned in the section. Modify the form you created in Section 2, Exercise 1 to include the following:

1. **Add a tab group to the form (Graphical Mode only)**



## Section 2 exercise 2, continued

2. Create a Select/Inquiry section of the form using Form Section 8.



## Section 2 exercise 2, continued

### Section 2 exercise 2 fields

#### RECORD-DATE

**Text Box**

Field Name:   Spin Button?

Label:

Section:

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

Exit Routines

Before:

After:

When Shown

- Always
- Character Mode
- Graphical Mode

Current Value

- Don't Show
- Show Inside Box
- Show Below Box

#### MARITAL-STATUS

**Display Box**

Field Name:

Label:

Exit Routines

Before:

After:

When Shown

- Always
- Character Mode
- Graphical Mode

Label Location

- No Label
- Left, Left Justified
- Left, Right Justified
- Above, Left Justified
- Above, Centered on 2 Lines

Section:

## Section 2 exercise 2, continued

### Section 2 exercise 2 fields

#### TOTAL-DEPENDENTS

The 'Text Box' dialog for 'TOTAL-DEPENDENTS' includes the following settings:

- Field Name: TOTAL-DEPENDENTS
- Label: Total Dependents
- Section: 8
- Spin Button? (unchecked)
- When Shown: Always (selected), Character Mode, Graphical Mode
- Label Location: Above, Centered on 2 Lines (selected), Left, Left Justified, Left, Right Justified, Above, Left Justified, No Label
- Current Value: Show Inside Box (selected), Don't Show, Show Below Box
- Exit Routines: Before, After (empty text boxes)
- Buttons: OK, Delete, Cancel

#### CITIZEN-COUNTRY

The 'Display Box' dialog for 'CITIZEN-COUNTRY' includes the following settings:

- Field Name: CITIZEN-COUNTRY
- Label: Citizenship
- Section: 8
- When Shown: Always (selected), Character Mode, Graphical Mode
- Label Location: Above, Left Justified (selected), Left, Left Justified, Left, Right Justified, Above, Centered on 2 Lines, No Label
- Exit Routines: Before, After (empty text boxes)
- Buttons: OK, Delete, Cancel

## Section 3 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. **Create a form program that displays the form you designed using Form Builder.**

```
P100-START-SCREEN.
  KEY-REQUIRED.
  UPDATE-EMPLOYEE.
  NEW-SCREEN-STYLE.
  SCREEN-SECTION '0'.
  IF INQUIRY-MODE OR SELECTION-MODE
    GO TO P990-INQUIRY-SCREEN.
P200-ENTRY-SCREEN.
  SCREEN-SECTION '1'.
P300-VERIFY.
  SET-FOR-MESSAGES.
  IF ERRORS-EXIST RETURN.
  IF RECORD-NOT-UPDATED
    RETURN.
  CALL 'XXXXXX'. @ EDIT ROUTINES
  RETURN.
P990-INQUIRY-SCREEN.
  SCREEN-SECTION '8'.
  RETURN.
```

2. **Create context-sensitive menu records for your form. Include the following forms:**

- 03-SCR Telephone Information
- 24-SCR Automobile Information
- 16-SCR Emergency Contact/Physician

Screen/Sq	ScrNam	GrLnky	Menu	Screen Title
X1-SCR 01	03-SCR		Y	Telephone Information
X1-SCR 02	24-SCR		Y	Automobile Information
X1-SCR 03	16-SCR		Y	Emergency Contact/Physician
YE0SCR 01	YE1SCR		Y	T4 Specifications
YE0SCR 02	YE2SCR		Y	T4 EI Ratings
YE0SCR 03	YE4SCR		Y	T4 CPP and EI Rates
YE0SCR 04	YE5SCR		Y	T4 Accounting Contact
YE0SCR 05	YE6SCR		Y	T4 Transmitter Summary
YE0SCR 06	YE7SCR		Y	T4 Transmitter Name
YE0SCR 07	YE8SCR		Y	T4 Transmitter Address
YE0SCR 08	YE9SCR		Y	T4 Transmitter Location
YE0SCR 09	YE0SCR	G	Y	T4 Transmitter Contact
YE10CR 01	YE10CR	G	Y	T4A Specifications
YE10CR 02	YE12CR		Y	T4A Business Number Box 61
YE10CR 03	YE13CR		Y	T4A Media Business Number
YE10CR 04	YE14CR		Y	T4A Accounting Contact
YE10CR 05	YE15CR		Y	T4A Transmitter Summary
YE10CR 06	YE16CR		Y	T4A Transmitter Name
YE10CR 07	YE17CR		Y	T4A Transmitter Address

3. **Generate the form program using the Update Form Appearance Table program (PUTSAT).**
4. **Run and test the form program.**





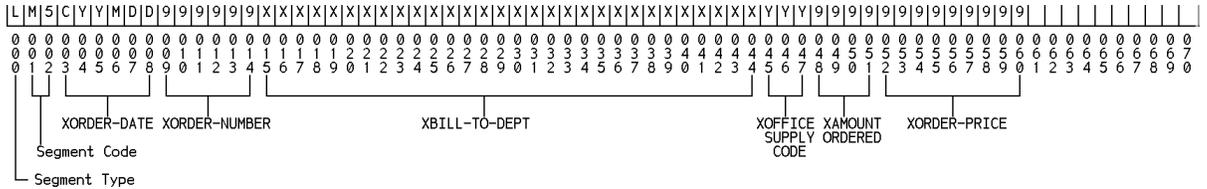


## Section 5 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

Your company has decided to computerize the ordering of office supplies. The office supplies will be ordered by each employee using a data entry form. Later a report will be created for each employee, to be used as the order form for the supplies.



1. **Create a form that displays the following fields and allows them to be updated.**

Field	Nbr of Characters (Field Length)
ORDER DATE (CYYMDD)	6
ORDER NUMBER	6
BILL TO DEPARTMENT	30
OFFICE SUPPLY CODE	3
OFFICE SUPPLY DESCRIPTION	20
AMOUNT ORDERED	4
ORDER PRICE	9

## Section 5 exercise, continued

### Step 1 fields

#### XORDER-DATE

Field Maintenance And Edit	
Action: <input type="text"/>	Field Properties
Field Name: XORDER-DATE	Data Type: Century/Complement D
Field Location	Field Type: Key-Part of Key
Pointer: 36	Template: Date
Storage Length: 006	Lengths: Display: 10 Entry: <input type="checkbox"/>
Displacement: 003	Module: Payroll/HRMS
Field Options	Structure: Seg has ID and Code
Propagate: <input type="text"/>	Seg/Table ID: N9 Table Separator: <input type="text"/>
Rounding: <input type="text"/>	Codeset: <input type="text"/>
Header Switch: 206310	Edit Routine: <input type="text"/>
<input checked="" type="checkbox"/> RDBMS Field	

#### XORDER-NUMBER

Field Maintenance And Edit	
Action: <input type="text"/>	Field Properties
Field Name: XORDER-NUMBER	Data Type: Alphanumeric
Field Location	Field Type: Key-Ending/Only Key
Pointer: 36	Template: <input type="text"/>
Storage Length: 006	Lengths: Display: 08 Entry: <input type="checkbox"/>
Displacement: 009	Module: Payroll/HRMS
Field Options	Structure: Seg has ID and Code
Propagate: <input type="text"/>	Seg/Table ID: N9 Table Separator: <input type="text"/>
Rounding: <input type="text"/>	Codeset: <input type="text"/>
Header Switch: 206210	Edit Routine: <input type="text"/>
<input checked="" type="checkbox"/> RDBMS Field	

## Section 5 exercise, continued

### Step 1 fields, continued

#### XBILL-TO-DEPT

Field Maintenance And Edit	
Action:	
Field Name:	XBILL-TO-DEPT
Field Location	
Pointer:	36
Storage Length:	030
Displacement:	015
Field Options	
Propagate:	
Rounding:	
Header Switch:	100110
<input checked="" type="checkbox"/> RDBMS Field	
Field Properties	
Data Type:	Alphanumeric
Field Type:	
Template:	
Lengths:	Display: <input type="checkbox"/> Entry: <input type="checkbox"/>
Module:	Payroll/HRMS
Structure:	Seg has ID and Code
Seg/Table ID:	N9
Table Separator:	
Codeset:	
Edit Routine:	

#### XOFFICE-SUPPLY-CODE

Field Maintenance And Edit	
Action:	
Field Name:	XOFFICE-SUPPLY-CODE
Field Location	
Pointer:	36
Storage Length:	003
Displacement:	045
Field Options	
Propagate:	
Rounding:	
Header Switch:	307154
<input checked="" type="checkbox"/> RDBMS Field	
Field Properties	
Data Type:	Alphanumeric
Field Type:	
Template:	
Lengths:	Display: <input type="checkbox"/> Entry: <input type="checkbox"/>
Module:	Payroll/HRMS
Structure:	Seg has ID and Code
Seg/Table ID:	N9
Table Separator:	
Codeset:	SCZZ
Edit Routine:	

## Section 5 exercise, continued

### Step 1 fields, continued OFFICE-SUPPLY-DESC

Field Maintenance And Edit	
Action: <input type="checkbox"/>	
Field Name: <input type="text" value="XOFFICE-SUPPLY-DESC"/>	
Field Location	
Pointer: <input type="text" value="36"/>	
Storage Length: <input type="text" value="003"/>	
Displacement: <input type="text" value="045"/>	
Field Options	
Propagate: <input type="text"/>	
Rounding: <input type="text"/>	
Header Switch: <input type="text" value="100210"/>	
<input type="checkbox"/> RDBMS Field	
Field Properties	
Data Type:	<input type="text" value="Alphanumeric"/>
Field Type:	<input type="text" value="Codeset Description"/>
Template:	<input type="text"/>
Lengths:	Display: <input type="text" value="20"/> Entry: <input type="text"/>
Module:	<input type="text" value="Payroll/HRMS"/>
Structure:	<input type="text" value="Seg has ID and Code"/>
Seg/Table ID:	<input type="text" value="N9"/> Table Separator: <input type="text"/>
Codeset:	<input type="text" value="SCZZ"/>
Edit Routine:	<input type="text"/>

### XAMOUNT-ORDERED

Field Maintenance And Edit	
Action: <input type="checkbox"/>	
Field Name: <input type="text" value="XAMOUNT-ORDERED"/>	
Field Location	
Pointer: <input type="text" value="36"/>	
Storage Length: <input type="text" value="004"/>	
Displacement: <input type="text" value="048"/>	
Field Options	
Propagate: <input type="text"/>	
Rounding: <input type="text"/>	
Header Switch: <input type="text" value="107121"/>	
<input checked="" type="checkbox"/> RDBMS Field	
Field Properties	
Data Type:	<input type="text" value="Numeric 0 Decimals"/>
Field Type:	<input type="text"/>
Template:	<input type="text" value="-----"/>
Lengths:	Display: <input type="text" value="05"/> Entry: <input type="text"/>
Module:	<input type="text" value="Payroll/HRMS"/>
Structure:	<input type="text" value="Seg has ID and Code"/>
Seg/Table ID:	<input type="text" value="N9"/> Table Separator: <input type="text"/>
Codeset:	<input type="text"/>
Edit Routine:	<input type="text"/>

### XORDER-PRICE

Field Maintenance And Edit	
Action: <input type="checkbox"/>	
Field Name: <input type="text" value="XORDER-PRICE"/>	
Field Location	
Pointer: <input type="text" value="36"/>	
Storage Length: <input type="text" value="009"/>	
Displacement: <input type="text" value="052"/>	
Field Options	
Propagate: <input type="text"/>	
Rounding: <input type="text"/>	
Header Switch: <input type="text" value="100110"/>	
<input checked="" type="checkbox"/> RDBMS Field	
Field Properties	
Data Type:	<input type="text" value="Numeric 2 Decimals"/>
Field Type:	<input type="text"/>
Template:	<input type="text" value="-----99"/>
Lengths:	Display: <input type="text" value="12"/> Entry: <input type="text"/>
Module:	<input type="text" value="Payroll/HRMS"/>
Structure:	<input type="text" value="Seg has ID and Code"/>
Seg/Table ID:	<input type="text" value="N9"/> Table Separator: <input type="text"/>
Codeset:	<input type="text"/>
Edit Routine:	<input type="text"/>

## Section 5 exercise, continued

### Entry Form Design

Office Supply Order Screen

Order Date> XXXXXXXXXXX

Order Number> XXXXXXXX

Bill to Department: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Supply Category: XXXXXXXXXXXXXXXXXXXXXXX

Quantity: XXXXX

Price: XXXXXXXXXXXXX

1 Section 1

### Select/Inquiry Form Design

Form Builder - XORDER.SAT

Office Supply Order Screen

Order Date	Order Number	Quantity	Price	Supply Category
XXXXXXXX	XXXXXXXX	XXXXX	XXXXXXXXXX	XXX

1 8 Section 8

## Section 5 exercise, continued

2. Write a form program that will display the form you designed and create context-sensitive menu records for navigation to related forms.

```

P100-START-SCREEN.
  KEY-REQUIRED.
  UPDATE-EMPLOYEE.
  NEW-SCREEN-STYLE.
  SCREEN-SECTION '0'.
  IF INQUIRY-MODE OR SELECTION-MODE
    GO TO P990-INQUIRY-SCREEN.
P200-ENTRY-SCREEN.
  SCREEN-SECTION '1'.
P300-VERIFY.
  SET-FOR-MESSAGES.
  IF ERRORS-EXIST RETURN.
  IF RECORD-NOT-UPDATED
    RETURN.
  CALL 'XXXXXX'. @ EDIT ROUTINES
  RETURN.
P990-INQUIRY-SCREEN.
  SCREEN-SECTION '8'.
  RETURN.
    
```

COMMAND:

Screen/Sq	ScrNam	GrLnky	Menu	Screen Title
XORDER 01	FF-SCR		Y	Employee Name/Address Changes
YE0SCR 01	YE1SCR		Y	T4 Specifications
YE0SCR 02	YE2SCR		Y	T4 EI Ratings
YE0SCR 03	YE4SCR		Y	T4 CPP and EI Rates
YE0SCR 04	YE5SCR		Y	T4 Accounting Contact
YE0SCR 05	YE6SCR		Y	T4 Transmitter Summary
YE0SCR 06	YE7SCR		Y	T4 Transmitter Name
YE0SCR 07	YE8SCR		Y	T4 Transmitter Address
YE0SCR 08	YE9SCR		Y	T4 Transmitter Location
YE0SCR 09	YE0SCR	G	Y	T4 Transmitter Contact
YE10CR 01	YE10CR	G	Y	T4A Specifications
YE10CR 02	YE12CR		Y	T4A Business Number Box 61
YE10CR 03	YE13CR		Y	T4A Media Business Number
YE10CR 04	YE14CR		Y	T4A Accounting Contact
YE10CR 05	YE15CR		Y	T4A Transmitter Summary
YE10CR 06	YE16CR		Y	T4A Transmitter Name
YE10CR 07	YE17CR		Y	T4A Transmitter Address
YE10CR 08	YE18CR		Y	T4A Transmitter Location
YE10CR 09	YE19CR		Y	T4A Transmitter Tech.Contact

## Section 5 exercise, continued

3. Create a FIND- verb and a Solution View Cross-reference record.

MAKE A 'FIND CURRENT OCCURRENCE' VERB MKVERB

This screen will create a verb called FIND-CURRENT-Lxx-SEG where xx is a specified segment code. The verb presumes the first key field in the segment is a date in the century-complement form.

Complete the text boxes below, then hit enter:

If needed, alter Org Level:

Enter SEGMENT-CODE:

Enter MODULE-CODE:

e.g. PR, BA, HR, SA, etc.

Solution View Cross Reference Information

Enter the following cross reference elements

Segment: User Defined LN9 Seg

Non-key Data Name:

Find Current Occurrence Verb:

Read Table Verb:

All Key Fields For Segment:

<input type="text" value="XORDER-DATE"/>	<input type="text" value="XORDER-NUMBER"/>
<input type="text"/>	<input type="text"/>

Solution View is a Registered Trademark of Cyborg Systems, Inc.

## Section 5 exercise, continued

### Entry and Select/Inquiry Form

Office Supply Order Screen AUSTIN, STEVEN

Order Date> 01-01-2000  
Order Number> 000001

Bill to Department: Accounting

Supply Category: Legal Pad of Paper

Quantity: 10  
Price: .75

Save This Form  
Cancel This Form  
Select An Employee...  
Show Selection  
Employee Name/Address Changes

Order Date	Order Number	Quantity	Price	Supply Category
01-01-2000	000001	10	.75	Legal Pad of Paper
01-01-2000	000002	5	1.50	Post-It Notes (3x5)
01-01-2000	000003	1	3.00	#2 Pencils (Bx 12)

*Note: The form displays the prompts when the user performs a right-click.*

## Section 6 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section.

1. **Add code to your Office Supply Order form to perform the following functions:**
  - Cause a field error reject message to appear when an Order Number of all zeros is entered. Use reject message group ERRSCR, message number Z00. Include explanation and recommended action documentation.

XORNBR

```

0000 SECURITY. @ Order Number field edit subroutine
00100 IM-A-SUBROUTINE.
00200 IF W8-06-240 EQUALS '000000'
00300   MOVE '@' TO SCREEN-ERROR
00400   MOVE '@' TO W8-01-480
00500   MOVE '@SCRZ00' TO W8-07-330.
00600 RETURN.

```

## Section 6 exercise, continued

- Include code in your form program to cause a warning message to appear when more than 100 Post-it Pads are ordered. Use warning message group ERRSCW, message number Z00. Include explanation and recommended action documentation.

```
XORERR

00000 SECURITY. @ Error Subroutine
00100 IM-A-SUBROUTINE.
00200 IF XOFFICE-SUPPLY-CODE EQUALS '034' AND
00300 XAMOUNT-ORDERED GREATER THAN :100
00400 PRINT-WARNING 'SCZ00'.
```

```
XORDER

P100-START-SCREEN.
  KEY-REQUIRED.
  UPDATE-EMPLOYEE.
  NEW-SCREEN-STYLE.
  SCREEN-SECTION '0'.
  IF INQUIRY-MODE OR SELECTION-MODE
    GO TO P990-INQUIRY-SCREEN.
P200-ENTRY-SCREEN.
  SCREEN-SECTION '1'.
P300-VERIFY.
  SET-FOR-MESSAGES.
  IF ERRORS-EXIST RETURN.
  IF RECORD-NOT-UPDATED
    RETURN.
  CALL 'XXXXXX'. @ EDIT ROUTINES
  RETURN.
P990-INQUIRY-SCREEN.
  SCREEN-SECTION '8'.
  RETURN.
```

## Section 7 exercise

### Purpose

This exercise gives you practice using the information you have learned in the section. Using the Office Supply form data, do the following:

1. **Create a query program to list the following information:**

### Title

Office Supply Query

### Fields

- Employee Name (length of 10 positions)
- Order Date
- Order Number
- Bill to Department (length of 20 positions)
- Office Supply Code
- Amount Ordered
- Price

```

XORDQ1 - First Occurrences of the Office Supply Order Form

P100-START.
  READ-EMPLOYEE.
  IF TIME-TO-PRINT-TITLE
    PRINT 'Office Supply Query'  NEXT-LINE NEXT-LINE.
  QUERY-HEADERS.
  FIND XORDER-PRICE.
  IF FOUND
    INQUIRY EMPLOYEE-NAME-10 XORDER-DATE XORDER-NUMBER
              XBILL-TO-DEPT-20 XOFFICE-SUPPLY-CODE
              XAMOUNT-ORDERED XORDER-PRICE
    NEXT-LINE.

```

```

XORDQ1 - Multiple Occurrences of the Office Supply Order
Form

P100-START
  READ-EMPLOYEE.
  IF TIME-TO-PRINT-TITLE
    PRINT 'Office Supply Query' NEXT-LINE NEXT-LINE.
  QUERY-HEADERS.
  PROCESS XORDER-PRICE.
    INQUIRY EMPLOYEE-NAME-10 XORDER-DATE XORDER-NUMBER
              XBILL-TO-DEPT-20 XOFFICE-SUPPLY-CODE
              XAMOUNT-ORDERED XORDER-PRICE.
  NEXT-LINE.
END-PROCESS.

```

2. **Run the Office Supply Query program.**

---

**NOTES**

---

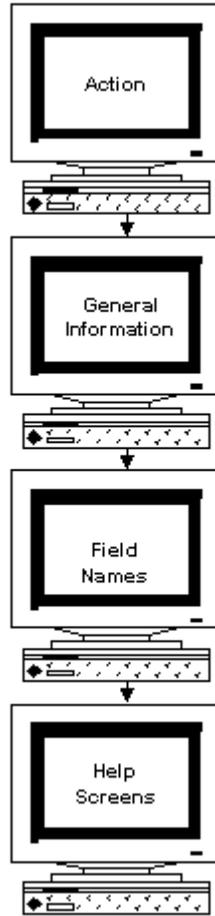
## Appendix B: Solution View

---

### Table of Contents

The new fields definition form series .....	251
Selecting the action .....	253
Titling and describing the new form.....	255
Defining the fields .....	259
Accessing the new form.....	263
New form sample .....	265

# Form Writer Form Series



## The new fields definition form series

### New Fields Definition Writer forms

New Fields Definition Writer (NEWSCR) is an end-user tool that enables you to create new fields and a data entry form to maintain the fields. The facility creates the program and field definitions from entries you make. One form requires general information about the new form, such as its name and title. Another form asks for the field list for the new form.

You can use New Fields Definition Writer (NEWSCR) to create both employee and company forms.

New Fields Definition Writer (NEWSCR) forms may only be created by persons having unlimited Control File access on the Cyborg system. The fields you create through Solution View are automatically added to Field Name Tables.

*Note: You cannot create Table forms with this facility.*

These are the forms, required and optional, that are available in Solution View to create your Form program.

- Action form  
allows you to name the form program, give the form a title, and select the type of action to be performed
- General Information form  
used to specify the form title, description and prompts
- New Field Name Entries form  
used to define the fields to be displayed on the new form
- Help form  
displays to signify that the form program has compiled successfully and prompts you to use the new form

Now that you have previewed the form series that you will use to create the new fields and form, let us look at each task in detail, see what tools are provided on each form, and how to use them.

# Action Form

Solution View Tool Kit      Option:

Name of Program: XAWARD      Title: Service Awards

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

## Selecting the action

### Action form

The first New Fields Definition Writer (NEWSCR) form, the Action form, is used to identify the Writer program and takes you directly to that form series.

### Accessing the form

Make the following selections from the Navigator:

<b>Component:</b>		User Tools
<b>Process:</b>		Application Tools
<b>Task:</b>		Solution View

**Result:** The Action form is displayed.

### Using the action form

To add a New Fields Description program:

<u>Selection:</u>	<u>Step:</u>
<b>Name of Program</b>	1. Type the one- to six-character name of the program, beginning with X, for example, XAWARD.
<b>Title</b>	2. Type a title for the New Fields Description form, for example, Service Award Form (optional).
<b>Program Type</b>	3. Select New User Fields (N).
<b>Action</b>	4. Select Add (A).
<b>Module</b>	5. Select the module ID to be associated with the report (optional).
<b>Security Code</b>	6. Type the Security Code for the report (optional).
<b>OK</b>	7. Choose OK or press Enter.

**Result:** When the Action form is error-free, the General Information form displays.

# General Information Form

Solution View - New Fields Definition \*\*\*\*\* General Information \*\*\*\*\*

Screen Name	Screen Title	Security Code	Module Code	Employee Data
XAWARD	Service Awards	**	PP	Y

Screen Description

Use As Prompts

---

## Titling and describing the new form

### General information

The General Information form requests basic information about the form you are creating. Some fields on this form are optional.

### Field definitions

The General Information form contains the following fields:

#### Screen name

Displays the name that was entered on the Action form. This field is automatically filled.

#### Screen title

Used to enter a one- to thirty-character title for the form program. This title displays at the top of the entry form.

#### Security code

Optionally displays the Security Code entered on the Action form.

#### Module code

Optionally displays the Module Code entered on the Action form.

#### Employee data

Used to specify either Y (the default) for an employee-level form, or N to create a company level form.

#### Screen description

Optionally used to enter HELP form documentation by typing a one- to four-line explanation of the form's purpose.

#### Screen prompts

Optionally used to create prompt message text that appears on the bottom of the newly created form.

If you wish to include the message text as a navigational prompt, choose the Use As Prompts check box to create a button to the left of the prompt text.

The prompt text must be in this format:

XXXSCR      Form title

XXXSCR      Form title

(where XXX are the first three characters of the form program name).

#### Overriding the segment code

Use this form to override the segment code, if applicable. Type the override segment code in the Additional Key field of the Command Line before executing the form.

# General Information Form

Solution View - New Fields Definition \*\*\*\*\* General Information \*\*\*\*\*

Screen Name	Screen Title	Security Code	Module Code	Employee Data
XAVARD	Service Awards	**	PP	Y

Screen Description

This screen is used to record service awards granted to an employee. Awards may be Letters of Recognition, monetary awards, or gifts.

Use As Prompts

HH-SCR Employee Earnings and Deductions.

## Titling and describing the new form, continued

### Using the general information form

To use the General Information form:

<u>Selection:</u>	<u>Step:</u>
<b>Additional Key Field</b>	1. Optionally, type an override segment code value.
<b>Screen Name</b>	2. Bypass this field.
<b>Screen Title</b>	3. Bypass this field.
<b>Security Code</b>	4. Bypass this field.
<b>Module Code</b>	5. Bypass this field.
<b>Employee Data</b>	6. Retain Y (the default) for an employee-level form, or type N to create a company level form.
<b>Screen Description</b>	7. Type a one- to four-line explanation of this form's purpose (optional).
<b>Use As Prompts</b>	8. Choose Use As Prompts if message text is a navigational prompt.
<b>Screen Prompts</b>	9. Type one to three lines of text that you want displayed at the bottom of the form (optional).
<b>OK</b>	10. Choose OK or press enter

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape a form that has errors, select Restart Process (X) in the Option Field, and execute the form to return to the Action form. Then delete the program and re-enter it correctly.

**Result:** When this form is error-free, the next form in the series displays.

*Note:* If a *SAVE CHANGES?* dialog box appears when you press <Enter> (Step 10), choose *NO* and then press *OK* or <Enter>.



## Defining the fields

### New Field Name Entries form

The third form in the New Fields Definition Writer (NEWSCR) form series is the New Field Name Entries form which allows you to define the fields that you want on the form. You must assign unique field names which may not have imbedded spaces; use hyphens to link words in a field name. Starting your field names with X prevents conflict with delivered CYBORG field names.

### Field descriptions

The New Field Name Entries form contains the following fields:

#### New field name

Used to enter a 1- to 20-character field name. List key fields first.

#### Length

Used to specify the number of characters that the field requires (limit of 30). This field is required for all fields except DT fields (dates) and CD fields (Option List descriptions). You may leave the Length entry field blank for DT and CD fields to assign the default values.

The Date field (DT) default is 06.

The Option List Description (CD) default is 20.

*Note: The total characters in all key fields must not exceed 12 (stored length).*

#### Type

Used to designate the field type, such as date, numeric, code set. Valid types are:

blank	- alphanumeric
CD	- Option List Description
DT	- Date
N0	- Numeric/0 Decimals
N2	- Numeric/2 Decimals
N4	- Numeric/4 Decimals
N6	- Numeric/6 Decimals
N\$	- Numeric/Leading \$

#### Key

Used to enter K to designate a field as a key, making the segment that is created a multiple occurrence segment. List key fields first in order of importance. More than one field can be combined to create a key. Enter G to designate a required non-key field when the form is entered.

#### Option list

Designates the field as an option list field, and allows you to specify the option list that validates this field.



---

## Defining the fields, continued

### Using the new field name entries form

To use the New Field Name Entries form:

<u>Selection:</u>	<u>Step:</u>
<b>New Name Field</b>	1. Type a unique, 1- to 20-character field name. (One way of assuring this is to prefix all new field names with X.) List all Key fields before non-key fields.
<b>Length</b>	2. Type the number of characters that the field requires. Bypass this field to use the system default for a date or Option List field.
<b>Type</b>	3. Type the code (listed on the right side of the form) that defines the field type.
<b>Key</b>	4. Type K if the field is part of the segment key. Type G if this is a required non-key field.
<b>Option List</b>	5. Type the name of the Option List that validates the field. 6. Repeat Steps 1 through 5 for each unique field name that you want on the form.
<b>OK</b>	7. Choose OK or press Enter.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape a form that has errors, select Restart Process (X) in the Option field, and execute the form to return to the Action form. Then delete the program and re-enter it correctly.

**Result:** If this form is error-free, the program source code is written and compiled, and the Help form displays.

# Help Form

Service Awards XAWARD

This screen is used to record service awards granted to an employee. Awards may be Letters of Recognition, monetary awards, or gifts.

Complete the text boxes below, then hit enter:

If needed, alter Org Level:

For inquiry only, enter 'I':

Enter EMPL #:  or SSN:

or NAME:

Enter XAWARD-DATE:CCYYMMDD

## Accessing the new form

### Updating FILECL

After creating new fields to be used by the entry form program, execute the UPDTCL program before using the new form for the first time. FILECL is a delivered indexed file that contains Option Lists, event menu records, field definitions, and some security information.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		System Operations
<b>Task:</b>		Update Client File

**Result:** The FILECL is updated.

### Help form

You may now access the new form through the Help form.

The Help form only displays if your entry form program has been written and compiled successfully. Cyborg Scripting Language (CSL) Code now exists for the form program.

Make the appropriate entries in the Help form and select Execute Screen to run the entry form program.

### Direct form access

As an option from this point on, you may access the new form directly by using the Command Line.

You may delete any existing occurrence by using the Delete This Entry function (ZDELETE).

# New Form Sample

Service Awards AUSTIN, STEVEN

Award Date:   
Award Type:   
Award Amount:

Save This Form  
Select An Employee...  
Show selection  
Employee Earnings and Deductions

Service Awards AUSTIN, STEVEN

Award Date: 10-01-1998  
Award Type: LR Letter of Recog  
Award Amount: .00

Save This Form  
Select An Employee...  
Show selection  
Employee Earnings and Deductions

---

## New form sample

### New fields definition

The form displayed on the left is the result of a Form New Fields Definition program (NEWSCR) that was created using Solution View.

### Verify form and fields

After creating the data entry form program, it may be desirable to verify the features of the new form to gather information for documenting the form's use. This verification includes, but is not limited to:

- All key fields are labeled with a (>) preceding the entry field.
- Valid values can be entered into each field:
  - Date fields accept data as YYMMDD or MM-DD-YY in the US and Canada, DD-MM-YY elsewhere.
  - Numeric fields accept the proper number of integers and decimals. Results display on the form properly.
  - Required fields must be entered, otherwise an error occurs.
  - Option List fields are edited against an Option List, and display the correct literal description.

For multiple occurrence segments, test the following functions from the Actions Menu Bar:

- Next Page  
Move to the next oldest entry if the occurrence is stored by date, or the next highest numbered or lettered occurrence. (Code2 = U)
- Previous page  
Move back up the occurrences until you reach the first entry. (Code2 = B)
- First Entry  
Move to the most recent occurrence or the top of the stacked segment. (Code2 = F)
- Select  
Provides a summary list of all the form's occurrences. (Code1= I)

### Finishing touches

You may want to add your new form to the online menu feature. Procedures for adding forms to the menu, as well as other options, are listed in the menu SCHL00.

---

**NOTES**

---

## Appendix C: Quick Solution—Technical Reference Guide

---

### Table of Contents

System Control Repository object codes .....	268
Operating system codes .....	269
Edit command list .....	270

## System Control Repository object codes

System Control Repository object codes are used to select specific (FILE01) records for processing.

<u>Object Description</u>	<u>Object</u>	<u>Object Description</u>	<u>Object</u>	<u>Object Description</u>	<u>Object</u>
A Records	A	Field Documentation	F/D	Rpt C1-2 Schedule	PD
Alt Lang Doc	P/N	Field Entry	F1	Rpt C1-2 Validation	PC
Alt Lang Error Msgs	P/Q	Field Name Tbl Menu	FTM	Rpt Print Pos Recrds	RT
Alt Lang Menu	P/K	Field Name Tbl Xref	FTX	Rpt Schedule	PE
Alternate Key Records	Q	Field Security	FS	Sample JCL	P/J
Assembler Source	P/W	Field Table (All)	F	Screen Chain	P/C
B Records	B	FILECL Updates	MCL	Screen Items Table	P/S
Codeset (All)	C	GUI Menu	MMN	Security Records	Y
Codeset Doc	C/D	HRMS L Defaults	D	Security Violations	Y/V
Codeset EL Calc	C/M	Hypertext Titles	P/D	Sol View Qry Specs	RQM
Codeset EL Edit	C/R	Locked Records	ZL	Sol View Qry Src	QRY
Codeset Value	C/V	Menu	P/L	Sol View Rpt Specs	RRM
Codeset W/Country cd	C/C	Object Description	Object	Sol View Scr Specs	RSM
Consultant Tables	W	Obj/Act Security	PXS	Sol View Screen Src	SCR
Context Menus	ECM	Object Code	P/X	Sol View Xref Recrds	XRF
Distributed Solution Tables	U	Objects	PX	Sol View Xtr Specs	RXM
Documentation	P/M	P Records (All)	P	Sol View Xtr Src	XTR
Edit Heading	PXH	P Records (No Obj)	P-X	Table	T
EL Rpt Components	RPT	PAYXTR/PAYMRG Select	EXT	Transaction Data	P/T
EL Rpt Source (All)	RS	Pos Mgt Navigation	EPM	User Tables	X
EL Source	P/	PTF History	P/H	Verbs	F/V
EL Source (All)	SRC	R Records	R	Virtual Pgm Name	PI
EL Source-Generated	P/G	RDBMS View Names	ETL		
Error Messages	P/R	Reload Messages	P/E		

## Operating system codes

Operating system codes are used to distinguish unique platform characteristics.

<u>Manufacturer</u>	<u>Code</u>	<u>Environment</u>	<u>Manufacturer</u>	<u>Code</u>	<u>Environment</u>
Bull	HW7	DPS 7 (5)	NCR	NCM	NCR (MEGAPATH) (5)
	HW8	DPS 8 (5)		NCR	NCR (NO MONITOR) (5)
Data General	DG	DATA GENERAL (4)	Prime	NCT	NCR (TRANPRO) (5)
Digital Equipment	VAX	DEC/VAX (5)		PR1	PRIME (VT100) (5)
	VAR	DEC/VAX RDB (5)	PR2	PRIME COBOL 2 (5)	
Hewlett Packard	HP3	H/P 3000 (4)	Unisys	PRM	PRIME (PT200) (5)
	HPS	H/P SPECTRUM (4)		U11	UNIVAC 1100/TIP(4)
IBM	AS4	IBM AS/400 (5)	U1M	UNIVAC 1100/MCB(4)	
	ASR	IBM DB2/400	U1S	UNIVAC 1100/SFS (4)	
	CM2	IBM VM/CMS COBOL 2 (4)	U31	UNIVAC OS3 (4)	
	DL2	IBM DLI COBOL 2 (4)	U30	UNIVAC 9030 (4)	
	DO2	IBM DOS COBOL 2 (4)	BME	BURROUGHS MEDIUM (5)	
	DO2	IBM VSE/ESA COBOL 2 (4)	BLR	BURROUGHS LARGE (6)	
	ID2	IBM IDMS COBOL 2(4)	BLC	BURROUGHS COMS (6)	
	IM2	IBM IMS COBOL 2 (4)	Wang	WNG	WANG (5)
	OS2	IBM OS/MVS COBOL 2(4)	Various - UNIX	MF2	AIX, GPIX, ULTRIX XENIX (5)
	OSR	IBM OS/MVS DB2 (4)	Various - PC	PC	IBM Compatible PC (4)
	S38	IBM System 38 (5)	MFJ	NT SQL Server	
NCR	NCB	NCR (BOSS 3) (5)	MFT	NT Indexed	

## Edit command list

The Edit Utility form (EDIT) contains a secondary command line with two fields. These EDIT command fields allow you to perform various editing functions on the current object.

The first field, referred to as the command field, requires a one-character command code to perform actions within the EDIT program. The second field, referred to as the parameter field, is used to specify parameters related to certain command codes.

*Note: Some codes are valid for specific objects;*

*\* Denotes commands valid for all P records.*

*\*\*Denotes commands valid for OBJECT = P/.*

### **A—AUTO ADD MODE\*\***

The A command automatically adds sequence line numbers to the end of your file, in increments of 100. It places an A (Add) in the first column of each added line, making them ready for entry. Auto Add Command:

(A)( )

### **B—BACKUP 19 LINES\***

The B Command allows you to display the previous page of your file.

Backup 19 Lines Command:

(B)( )

### **C—CHANGE STRING**

The C command works like a global change. You can request that the system display all lines containing the specified word or string together, or one line at a time. You must use literals, that is, the exact characters as they appear in the string.

Change all Command:

(C)(/READ-EMPLOYEE/UPDATE-EMPLOYEE/ )

Change One Command:

(C)(READ-EMPLOYEE/UPDATE-EMPLOYEE/1 )

### **D—DELETE RANGE\***

The D command allows you to delete lines.

Line delete Command:

(D)(00220, 00240, 00530 )

Range delete Command:

(D)(00220/00350 )

**E—END AUTO ADD MODE\*\***

The E command ends the Auto Add Mode and returns the form to manual operation.

End Auto Add Command:

(E)( )

**F—FIND STRING**

The F command will display all lines containing the specified word or string together, or one line at a time. You must use literals, that is, the exact characters as they appear in the string. You can also use an equal sign (=) as a wild card in all but the first position of the search argument.

Find all Command:

(F)(/W8-01-999/ )

Find one Command:

(F)(/CONTROL-1-2/1 )

Find Wildcard Command:

(F)(CONTROL - = CODE/ )

**G—GO TO SEQUENCE NUMBER**

The G command can select a line to appear as the first line on your form.

Go To Command:

(G)(00312 )

**H—HOLD RANGE\*\***

The H command is used to hold a range of lines for the purpose of inserting (I) or Transferring (T).

Hold Range Command:

(H)(00100/00200 )

**I—INSERT HELD RANGE\*\***

The I command is used with the H (HOLD) command to copy text or code within the same file. There must be room between the existing lines to insert the copied lines. The inserted line numbers must be currently unused and not duplicate existing line numbers. The inserted lines are renumbered in increments of 10 unless otherwise stated.

Insert Command:

(I)(00205 )

Insert with increments of 5 lines Command:

(I)(00205/005 )

### **K—KEY CHANGE**

The K command is used to quickly change the Edit Utility form (EDIT) from the file you are editing to a different file within the same record type (OBJECT code). If you are editing a documentation file (P/M) and wish to access another (P/M) file, you should use this quick method. You can also use the K command to create a new file.

Change file command:  
(K)(MYFILE )

### **L—LOGGING ON/OFF\*\***

The L command is used to activate LOGGING. Logging saves the original lines of code, as they were prior to any editing, and places them in FILE01. They are saved under your Operator ID and are remembered for as long as LOGGING stays on.

Turn Logging On/Off Command:  
(L)( )

### **M—MINI DISPLAY ON/OFF**

The M command is used to reduce the size of the form display from a whole form to 1–9 lines.

Mini Display ON 5 line Command:  
(M)(5 )

Mini Display OFF Command:  
(M)( )

### **N—NEXT FORM**

The N command is used to advance to the next page of text or source. Typically, your display scrolls automatically when you press your enter key. You may use the N command to force the display to immediately advance to the next page when you make modifications and press your enter key.

Next Form Command:  
(N)( )

### **O—OBJECT CHANGE**

The O command is used to access a file that has an Object code that is different from the file on your Edit Utility form (EDIT). For example, if you are editing a Code Set file (C/V), and wish to change to a (P/ ) file, you must use this method.

Object Change Command:  
(O)(P/ )

### **P—PARAGRAPH LOCATE\*\***

The P command is used to locate and display a specified paragraph in your program. The paragraph label must begin in the first position of your code line.

Paragraph Locate Command:  
(P)(380)

**Q—QUIT EDIT**

The Q command is used to exit the EDIT program and the Cyborg System simultaneously. You do not have to use the GOODBY program to sign off. (This command is valid for versions 3.x and 4.x only.)

Quit Edit Command:

(Q)( )

**R—RELOAD\*\***

The R command automatically reloads, or compiles, your Cyborg Scripting Language source code (OBJECT = P/). The same rules apply as when performing RELOAD from the command Line.

Reload Command:

(R)( )

**S—RESEQUENCE\*\***

The S command used to resequence the lines of your program in increments of 100. The lines numbered from 00000 through 00010 are reserved and are NOT included in the resequence.

Resequencing Command:

(S)( )

**T—TRANSFER RANGE\*\***

The T command is used with the H (HOLD) command to transfer held ranges within a file (Move) or to another file (Copy). The lines are copied/moved in increments of 10 unless otherwise specified.

To COPY from one file to another, do the following:

Execute the H (HOLD) command, specifying the lines to be copied.

(H)(00310/00330 )

Execute the K (KEY CHANGE) command to access the program to which you wish to copy the lines.

(K)(NEWFILE )

Enter a T in the command field and the beginning sequence line number at which you want to copy the lines (in HOLD) in the parameter field. If you wish the copied lines to have an increment other than 10, type a slash after the beginning sequence line number and then a 3-digit increment.

(T)(00800 )

OR

(T)(00800/050)

To MOVE code within a file complete the following sequence of commands:

Execute the H (HOLD) command, specifying the lines to be copied.

(H)(00310/00330 )

**T—TRANSFER RANGE\*\***

Enter a T in the command field and the beginning sequence line number at which you want to move the lines (in HOLD) in the Parameter field.

(T)(00800 )

OR

(T)(00800/050)

*Note: The T command MOVES code to another place within the program you are currently editing. It automatically deletes the lines from their original location.*

**U—UNDO\*\***

The U command is used with the L command (LOGGING), to reverse or "undo" the previously performed edit. For example, if you deleted lines of code by mistake, UNDO will restore them.

*Note: Undo assumes that the L command (LOGGING) is active and the counter to the right of the EDIT Command Line contains a number. UNDO the last add, change or delete command.*

(U)( )

UNDO all the edited lines that are accounted for in the counter command:

(U)(ALL )

CAUTION: If you "undo" all prior editing, LOGGING automatically turns itself off.

**X—EXECUTE PROGRAM/FORM**

The X command is used to execute a form directly from the Edit Utility form (EDIT) command line, instead of using The Solution Series command line (top line on form). The parameter field entries must correspond with The Solution Series command line entry fields. You must allow for spaces between entries, as necessary.

Execute Program/Form Command:

(X)(EE-SCR 1234567890 )

:	:	:	:
Program Code Key	Additional-Key		
field	field	field	field
(1-6)	(7-8)	(9-18)	(19-33)

## Appendix D: Pointer 7 Fields

---

### Table of Contents

Pointer 7 fields .....	276
Pointer 7 fields in pointer order: .....	277
Pointer 7 fields in alphabetical order: .....	278
Pointer 7 field definitions .....	279

## Pointer 7 fields

### Description

Pointer 7 is a work area used by the CBSV programs that can also be used by other programs. It consists of pre designated fields and constants.

### Naming conventions

Work areas are defined as follows:

- W7-99-000-STAT
  - W—literal of W for work
  - 7—pointer number
  - 99—field length
  - 000—displacement within the pointer (relative to 0)
  - STAT—comments tag (ignored by the system)

**Pointer 7 fields in pointer order:**

W7-02-000	STAT-KEY	W7-01-301	COMM-ACTION
W7-01-003	TRACE-CONTROL	W7-01-302	COMM-CANCEL
W7-01-004	TRACE-READY	W7-01-303	COMM-INQUIRY
W7-01-005	QUOTE	W7-04-304	SESSION-ID
W7-02-006	LINE-ADVANCE	W7-06-316	QUERY-PROGRAM
W7-02-008	KEY01-SIZE	W7-02-322	QUERY-ALT-KEY
W7-24-010	KEY01-AREA	W7-19-324	QUERY-FROM-VALUE
W7-01-034	LOW-VALUE, LOW-VALUES	W7-02-343	QUERY-FROM-DUP
W7-09-035	MORE-LOW-VALUES	W7-19-345	QUERY-TO-VALUE
W7-02-044	KEY02-SIZE	W7-02-364	QUERY-TO-DUP
W7-32-046	KEY02-AREA	W7-01-366	QUERY-SWITCH
W7-14-078	SAVE-SEGMENT-KEY	W7-02-368	PAGE-LINES
W7-01-093	SCREEN-ERROR	W7-02-370	PAGE-LENGTH
W7-02-094	RECORD-LOCKED	W7-06-372	MENU-STOP
W7-01-096	SCREEN-WARNING	W7-13-378	TRACE-HOLD
W7-01-097	RECORD-UPDATED	W7-06-391	CURRENT-DATE-CYYMDD
W7-04-112	TRANS-NUMBER	W7-01-397	TRACE-SWITCH
W7-06-116	CURRENT-DATE	W7-01-398	AUTOHEADERS-SWITCH
W7-06-122	CURRENT-TIME	W7-01-399	HORIZONTAL
W7-48-128	CALL-STACK	W7-03-400	OPERATING-SYS-CODE
W7-01-176	SPACE, SPACES	W7-01-403	MACHINE-SWITCH
W7-61-177	MORE-SPACES	W7-01-428	SESSION-STATUS
W7-01-238	HIGH-VALUE, HIGH-VALUES	W7-01-429	CONTROL-COUNTRY
W7-01-239	MORE-HIGH-VALUE	W7-04-430	BALNCE-NUMBER
W7-06-240	COMPANY-NUMBER	W7-06-434	BACKREF-HOLD
W7-06-246	PROGRAM-FIELD	W7-01-440	PRODUCTION-VERSION
W7-01-252	CODE-1	W7-01-441	COLOR-INDICATOR
W7-01-253	CODE-2	W7-01-442	QM-PHASE
W7-10-254	KEY-FIELD	W7-01-443	DATE-FORMAT
W7-15-264	ADDITIONAL-KEY	W7-01-444	APPL-INDICATOR
W7-04-280	OPERATOR-ID	W7-03-445	MENU-PAGE
W7-01-298	COMM-CHAR	W7-01-448	AUTO-KEY-SWITCH
W7-01-299	COMM-MODE	W7-01-449	CODE-SET-SWITCH
W7-01-300	REPEAT-COUNTER		

## Pointer 7 fields in alphabetical order:

W7-15-264	ADDITIONAL-KEY	W7-01-239	MORE-HIGH-VALUE
W7-01-444	APPL-INDICATOR	W7-09-035	MORE-LOW-VALUES
W7-01-448	AUTO-KEY-SWITCH	W7-61-177	MORE-SPACES
W7-01-398	AUTOHEADERS-SWITCH	W7-03-400	OPERATING-SYS-CODE
W7-06-434	BACKREF-HOLD	W7-04-280	OPERATOR-ID
W7-04-430	BALNCE-NUMBER	W7-02-370	PAGE-LENGTH
W7-48-128	CALL-STACK	W7-02-368	PAGE-LINES
W7-01-252	CODE-1	W7-01-440	PRODUCTION-VERSION
W7-01-253	CODE-2	W7-06-246	PROGRAM-FIELD
W7-01-449	CODE-SET-SWITCH	W7-01-442	QM-PHASE
W7-01-441	COLOR-INDICATOR	W7-02-322	QUERY-ALT-KEY
W7-01-301	COMM-ACTION	W7-02-343	QUERY-FROM-DUP
W7-01-302	COMM-CANCEL	W7-19-324	QUERY-FROM-VALUE
W7-01-298	COMM-CHAR	W7-06-316	QUERY-PROGRAM
W7-01-303	COMM-INQUIRY	W7-01-366	QUERY-SWITCH
W7-01-299	COMM-MODE	W7-02-364	QUERY-TO-DUP
W7-06-240	COMPANY-NUMBER	W7-19-345	QUERY-TO-VALUE
W7-01-429	CONTROL-COUNTRY	W7-01-005	QUOTE
W7-06-116	CURRENT-DATE	W7-02-094	RECORD-LOCKED
W7-06-391	CURRENT-DATE-CYYMDD	W7-01-097	RECORD-UPDATED
W7-06-122	CURRENT-TIME	W7-01-300	REPEAT-COUNTER
W7-01-443	DATE-FORMAT	W7-14-078	SAVE-SEGMENT-KEY
W7-01-238	HIGH-VALUE, HIGH-VALUES	W7-01-093	SCREEN-ERROR
W7-01-399	HORIZONTAL	W7-01-096	SCREEN-WARNING
W7-10-254	KEY-FIELD	W7-04-304	SESSION-ID
W7-24-010	KEY01-AREA	W7-01-428	SESSION-STATUS
W7-02-008	KEY01-SIZE	W7-01-176	SPACE, SPACES
W7-32-046	KEY02-AREA	W7-02-000	STAT-KEY
W7-02-044	KEY02-SIZE	W7-01-003	TRACE-CONTROL
W7-02-006	LINE-ADVANCE	W7-13-378	TRACE-HOLD
W7-01-034	LOW-VALUE, LOW-VALUES	W7-01-004	TRACE-READY
W7-01-403	MACHINE-SWITCH	W7-01-397	TRACE-SWITCH
W7-03-445	MENU-PAGE	W7-04-112	TRANS-NUMBER
W7-06-372	MENU-STOP		

## Pointer 7 field definitions

### **ADDITIONAL-KEY**

W7-15-264

This area contains the Additional Key field value from the Command Line. See also the SCNT11 text file.

### **ADDL-KEY-PROGRAM**

W7-07-272

### **APPL-INDICATOR**

W7-01-444

This area contains a code used to indicate which application is being executed. It is set when you log in. Values are: H = Payroll/HR Solutions, T = Time and Attendance stand-alone.

### **AUTO-KEY-SWITCH**

W7-01-448

This area contains a switch set by the REPEAT verb and reset by CYB90.

### **AUTOHEADERS-SWITCH**

W7-01-398

This area contains a switch used by the AUTO-HEADERS verb.

### **BACKREF-HOLD**

W7-06-434

This area is used by the MENU and DOCPRn programs to hold a back reference menu name.

### **BALNCE-NUMBER**

W7-04-430

This area is used by the BA-SCR program. When a value other than "0000" is present, an Inquiry form is returned.

### **CALL-STACK**

W7-48-128

This area contains a CALL stack. It can hold up to 6 entries. Each entry contains a program name followed by a two-byte computational field. The computational field contains the displacement of the next statement within the calling module when control is returned to it. **DO NOT USE THIS FIELD NAME IN A PROGRAM.** This field name has been provided for documentation purposes only.

**CODE-1**

W7-01-252

This area contains the Code1 field value from the Command Line. This field receives a 'I' when an Inquiry Only Password is used to access the system. When this is the case, the I writes over any other option placed in the Code1 field on the Command Line. If special form processing uses the Code1 field, the form code must check the SCREEN-CODE1 field instead of the CODE-1 field for the option.

**CODE-2**

W7-01-253

This area contains the Code2 field value from the Command Line.

**CODE**

W7-02-252

This area is used to hold the CODE-1 and CODE-2 field values.

**CODE-SET-SWITCH**

W7-01-449

This area contains a switch indicating how Code Sets are treated. It is set by the ESTABLISH-QUOTE verb.

Values are:

Y = Treat Code Set type '9' as 'Y'.

N = Treat Code Set normally.

**COLOR-INDICATOR**

W7-01-441

This field contains a monitor color indicator. A value of 'C' indicates color monitor. A value of blank indicates a monochrome monitor.

**COMM-CHAR**

W7-01-298

This field contains the value of the Code1 field from the Command Line. This field is then used by the COMM-MODE field when building form output lines. See also COMM-MODE.

**COMM-MODE**

W7-01-299

This field is used by the START-LINE and REPEAT verb loop to create either an entry box line or inquiry data line when building a form. This field acts as a switch to determine which type of output line is being built. To do this, the START-LINE verb checks to see what mode the form is in by looking at the COMM-CHAR field. Based on the entry in COMM-CHAR, either one or two passes are performed to build the form line.

**COMM-CHAR    COMM-MODE    LINE BUILT**

E	E	entry boxes-one pass
I	I	inquiry data-one pass
blank	E	entry boxes-1st pass
I		inquiry data-2nd pass

Output lines are built field by field.

### **COMPANY-NUMBER**

W7-06-240

This area contains the Control 1-2 value from the Command Line.

### **CONTROL-COUNTRY**

W7-01-429

This field contains the country code from the AF transaction in the company header record. It is set when you log in.

### **CURRENT-DATE**

W7-06-116

This area contains today's date as YYMMDD.

### **CURRENT-DATE-CYYMDD**

W7-06-391

This field contains today's date in the complement form.

### **CURRENT-TIME**

W7-06-122

This area contains the current time as HHMMSS. It is updated whenever a field with an edit pattern of "35" is PRINTed.

### **DATE-FORMAT**

W7-01-443

This area contains a code that determines the format of an edited date. It is set by CBSVO. Values are:

0 = MM-DD-YY

1 = DD-MM-YY

### **HIGH-VALUE**

W7-01-238

This area contains a literal of a high value. See also W7-01-239.

### **HORIZONTAL**

W7-01-399

This field contains an 'H' if horizontal entry verbs are in use.

### **KEY-FIELD**

W7-10-254

This area contains the Key field value from the Command Line.

**KEY-WORD**

W7-06-254

This area is used by the MENU and DOCPRn programs to hold the Key word menu name entered in the Key field.

**KEY01-AREA**

W7-24-010

This area is used to supply the program with a key for FILE01 prior to a READ-UNIQUE operation.

**KEY01-SIZE**

W7-02-008

This area is used to specify the length of the FILE01 key in the field W7-24-010 (KEY01-AREA).

**KEY02-AREA**

W7-32-046

This area is used to supply the program with a key for FILE02 prior to an I/O operation. See also W7-02-044 (KEY02-SIZE).

**KEY02-SIZE**

W7-02-044

This area is used to specify the length of the FILE02 key in field W7-32-046 (KEY02-AREA).

**LINE-ADVANCE**

The LINE-ADVANCE command is used in -RP programs to define the number of blank lines you want to print following an output print line (carriage Control). There are two special LINE-ADVANCE values:

99—suppresses the output print line.

00—forces a top of page prior to printing the output print line. Headings are NOT repeated.

If this command is used, you must also account for the lines added by calculating LINE-COUNT. The special 99 value can be used to eliminate duplicate total lines for an employee.

**LOW-VALUE**

W7-01-034

This area contains a literal of low values.

**MACHINE-SWITCH**

W7-01-403

This field contains a space or an 'S', depending on the machine type.

**MENU-PAGE**

W7-03-445

This area is used by the MENU program to retain the first three digits of the sequence field of the last menu used.

**MENU-STOP**

W7-06-372

**MORE-HIGH-VALUE**

W7-01-239

This area contains another literal of a high value. See also W7-01-238 (HIGH VALUE or HIGH VALUES).

**MORE-LOW-VALUES**

W7-09-035

This area contains nine bytes of low values. See also LOW-VALUE and LOW-VALUES

**MORE-SPACES**

W7-61-177

This area contains sixty-one spaces. See also W7-01-176 (SPACE or SPACES).

**OPERATING-SYS-CODE**

W7-03-400

This field contains the operating system code that identifies the computer type in use.

**OPERATOR-ID**

W7-04-280

This area contains the Operator-ID that appears on the right side of the Command Line.

**OPTION-NUMBER**

W7-02-254

**PAGE-LENGTH**

W7-02-370

This field is a computational field used in batch programs. It contains the maximum number of lines to be printed on a page.

**PAGE-LINES**

W7-02-368

This field is a computational field used in batch programs to hold a line count.

**PRODUCTION-VERSION**

A Y entered in this field defines this version of the Control File as the production file. The Y causes certain programs to behave differently than if this field was left blank. The Y directs FILE02 to produce a normal Audit Trail report.

**PROGRAM-FIELD**

W7-06-246

This area contains the Program name value from the Command Line.

**QM-PHASE**

W7-01-442

This area is used by the Query Maintenance Facility to keep track of what phase it is in.

**QUERY-ALT-KEY**

W7-02-322

This area contains the QUERY Alternate Key value.

**QUERY-FROM-DUP**

W7-02-343

This area contains the QUERY From Duplicate Key value.

**QUERY-FROM-VALUE**

W7-19-324

This area contains the QUERY From key value.

**QUERY-PROGRAM**

W7-06-316

This area contains the QUERY Program name.

**QUERY-SWITCH**

W7-01-366

This area contains a switch used by QUERY.

**QUERY-TO-DUP**

W7-02-364

This area contains the QUERY To Duplicate Key value.

**QUERY-TO-VALUE**

W7-19-345

This area contains the QUERY To key value.

**QUOTE**

W7-01-005

This field contains a literal of a quote. The quote is placed in this field by the ESTABLISH-QUOTE verb in the CYB88 programs. This field can be used in MOVE statements to place a quote on a form or report.

**RECORD-LOCKED**

W7-02-094

This area is used to indicate the first pointer number associated with a record that is to be written or read. For example, the UPDATE-EMPLOYEE verb sets RECORD-LOCKED to "28" prior to calling CYBGET. Also, if RECORD-LOCKED is set to "40" in a form, the data in the SCREEN area is written to FILE02 as a "ZZ" (or Audit Trail) record.

**RECORD-UPDATED**

W7-01-097

This field is initially set up to a space when a form is entered. If any data is detected by an ENTRY verb, it is set to a "Y".

**SAVE-SEGMENT-KEY**

W7-14-078

This area is used by the ENTRY verb to save a segment key.

**SCREEN-ERROR**

W7-01-093

This area is initially set to an "F" by the system before calling a form. If there is an error found on the form, the value is changed to an @ (at) sign and a reject message (\*\*REJECT\*\*) is returned on line two of the returned form. Once the form error is corrected, this field is again set to an 'F'.

**SCREEN-WARNING**

W7-01-096

This area is initially set to a "W" by the system before calling a form. Other values are:

@ If an ENTRY verb detects an error, this area is set to an "@" and a warning message is returned to the form.

**SESSION-ID**

This field contains the current session number.

**SESSION-STATUS**

W7-01-428

This area contains the current session's status.

### **SPACE**

W7-01-176

This area contains a literal of a space. When used in a compare, each position of the compare field is checked for spaces. The compare field must be the second field. For example,

IF SPACE EQUALS KEY-FIELD MOVE 'MAIN M' TO KEY-FIELD.

In this example, each position of the Key field is checked for spaces. See also W7-61-177 (MORE-SPACES).

### **STAT-KEY**

W7-02-000

The STAT-KEY is set by CBSV COBOL programs after every I/O operation. The meanings of the returned values are:

00 = Good I/O

01 = Record read has greater key

10 = EOF

22 = Write failed due to duplicate record (1 or greater than 23)

23 = Record not found (START, DELETE or REWRITE)

24 = Space exhausted

90 = Invalid request

91 = Invalid file number (less than 23)

93 = Invalid file number (8 or 9)

95 = Open failed

99 = Invalid key

### **TRACE-HOLD**

W7-13-378

This area is used by the TRACE facility to hold:

# the name of the program

# a switch indicating whether the trace is on or off

# a starting paragraph number

### **TRACE-READY**

W7-01-004

This is an internal switch used by the TRACE program.

### **TRACE-SWITCH**

W7-01-397

This area contains a switch used by the TRACE program. A value of T means the CBSVOT or CBSVBT program is executing.

**TRANS-NUMBER**

W7-04-112

This area contains the current transaction number.

---

**NOTES**

---

## Appendix E: Form Footers Using Section 9

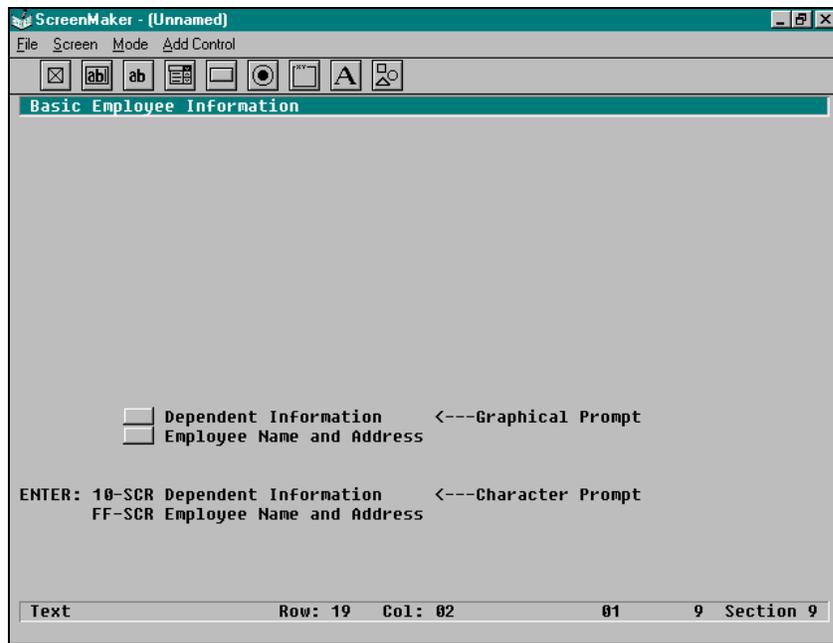
---

### Table of Contents

Creating the screen footer .....	291
----------------------------------	-----

## Screen Footer—Prompt

	<u>Position</u>	<u>Text</u>
First Prompt	1–6	ENTER:
	7	blank
	8–13	Program Name
	14	blank
	15–75	Program Title or message
Example:	ENTER: XX-SCR for additional information	
	<u>Position</u>	<u>Text</u>
Subsequent Prompts:	1–6	Program Name
	7	blank
	8–68	Program Title or message
Example:	YY-SCR for additional information	



## NOTES

## Creating the screen footer

### Screen footers

As you recall, footers contain prompts that provide links between forms. These link buttons are implemented using specially formatted text that is replaced with a small push button in graphical mode, while displaying as regular text in character mode.

In Form Builder, click the Add Text button or select the following from the menu:

Add Control ► Text

**Result:** The Text dialog box displays.

### Text

This field is used to type up to 60 characters to display on the Form Builder workspace. The Screen Prompts must comply with the format given above.

### When shown

The When Shown radio buttons are used to control the display of items based on the mode.

### Section

The Section field identifies a text item as being part of a Header, Footer, or Screen Body. Type a nine (9) in the Section field to define this as the Screen Footer.

### Completing the prompt

To complete the Text dialog, fill in each field as needed to create either the first or subsequent prompt.

- Position the first prompt starting in Position 2 of the workspace.
- Position subsequent prompts starting in Position 9 of the workspace.
- Prompts must be on consecutive lines of the workspace.

---

**NOTES**

---

## Glossary

---

### **ACTIVATE-OK-CANCEL**

The ACTIVATE-OK-CANCEL verb will activate the OK and CANCEL buttons in a GUI environment.

ACTIVATE-OK-CANCEL.

### **AUDIT-nn-FIELD**

The AUDIT-nn-FIELD verb causes an IS/WAS entry to be made in a table in memory. It is generated by the Generate Code from Form Appearance Table program (GENER8) as part of the code generated for a radio button.

- Before executing any of these verbs you must move the field name to W8-20-402, the old value to W8-30-300, and the new value to W8-30-270.

AUDIT-COMPANY-FIELD.

AUDIT-EMPLOYEE-FIELD.

AUDIT-OTHER-FIELD.

AUDIT-TAX-FIELD.

### **AUTO-HEADERS**

The AUTO-HEADERS verb specifies that a field name should appear as a heading on a form. AUTO-HEADERS automatically formats a field heading on a form by using specifications on the Field Name Table. These specifications are set by F-NAME when you add a new field to the system.

- Specify AUTO-HEADERS for each line of entry fields that you want displayed on a form.
- The AUTO-HEADERS verb must precede the ENTRY or INQUIRY verb.

AUTO-HEADERS.

### **BEGIN-ENTRY**

The BEGIN-ENTRY verb is used in conjunction with the ENTRY or INQUIRY verb and paired with the END-ENTRY verb and is used to find a particular segment occurrence in a multiple occurrence segment.

- BEGIN-ENTRY includes coding for AUTO-HEADERS and does not require it to be coded.
- BEGIN-ENTRY must precede the ENTRY and INQUIRY verbs.

BEGIN-ENTRY.

### **BUTTON-ANSWER**

The BUTTON-ANSWER verb must be coded at the beginning of any program that uses pushbuttons.

- This verb is coded just once at the beginning of the program.

BUTTON-ANSWER.

**BUTTON-GROUP**

The **BUTTON-GROUP** is used, when a GUI is in use, to indicate that the previous one-position field is to be replaced with a radio button. The current value of the field (?Y' or ?N') determines whether or not it is selected. **BUTTON-GROUP** moves ?%>' to the SCREEN area. A preceding PRINT statement must move a button group number ?n' to the SCREEN area. You may create up to ten different groups of buttons (?n' = 0-9).

- This is used with the **BUTTON-ON** and **BUTTON-OFF** verbs to create a radio button.

**BUTTON-ON**

The **BUTTON-ON** and **BUTTON-OFF** verbs enable you to create a **BUTTON-OFF** one-position entry field whose value is the contents of YES-NO.

**BUTTON-ON**            If it is the first pass in a form, YES-NO field is set to ?Y', otherwise WORK is moved to YES-NO.

**BUTTON-OFF**        If it is the first pass in a form, YES-NO field is set to ?N', otherwise WORK is moved to YES-NO.

- This is used with the **BUTTON-GROUP** verb to create a radio button.

*BUTTON-GROUP.*

**CHECKBOX**

The **CHECKBOX** verb allows you to indicate that the previous one-position field is to be replaced with a check box. The current value of the field (?Y' or ?N') determines whether or not it is checked. **CHECKBOX** moves ?:>' to the SCREEN area if a GUI-CLIENT is greater than a space. The GUI logic looks for the literal ?:>' and creates the check box.

- A check box should be used only for fields with a Yes/No value (?Y' or ?N').

*CHECKBOX.*

**CONFIRM-EMP-SEG-ADD**

The **CONFIRM-EMP-SEG-ADD** verb may be used to confirm that a segment will be added to an employee record. This verb should be placed after all field and relational edits have been performed. If a segment is not going to be added to an employee record, warning message PP012W will be displayed.

*CONFIRM-EMP-SEG-ADD.*

**CURSOR-ACTION-CODE**

The **CURSOR-ACTION-CODE** verb is used to place the cursor at the ACTION field on the Command Line, when the form is built.

*CURSOR-ACTION-CODE*

**CURSOR-ADDITIONAL**

The **CURSOR-ADDITIONAL** verb is used to place the cursor at the ADDITIONAL-KEY field on the Command Line, when the form is built.

*CURSOR-ADDITIONAL*

### **CURSOR-CODE-1**

The CURSOR-CODE-1 verb is used to place the cursor at the CODE-1 field on the Command Line, when the form is built.

CURSOR-CODE-1

### **CURSOR-C1-2**

The CURSOR-C1-2 verb is used to place the cursor at the CONTROL-1-2 field on the Command Line, when the form is built.

CURSOR-C1-2

### **CURSOR-MARK**

The CURSOR-MARK verb is used to place the cursor at the first entry field found after an ENTRY verb, when the form is built. When used this verb must be coded prior to the ENTRY command.

CURSOR-MARK

### **CURSOR-SCREEN-NAME**

The CURSOR-SCREEN-NAME verb is used to place the cursor at the PROGRAM field on the Command Line, when the form is built.

CURSOR-SCREEN-NAME

### **CURSOR-THIS-KEY**

The CURSOR-THIS-KEY verb is used to place the cursor at the KEY field on the Command Line, when the form is built.

CURSOR-THIS-KEY

### **DASH-OFF**

The DASH-OFF verb is used to place a line of dashes across the form.

DASH-OFF

### **DISALLOW-DELETE**

The DISALLOW-DELETE verb will gray the "Delete This Entry" option found under "Actions". This will not stop the user from using ZDELETE to delete an entry.

DISALLOW-DELETE.

### **EMPTY-BOX**

The EMPTY-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

**END-ENTRY**

The END-ENTRY verb is paired with the BEGIN-ENTRY verb to determine the end of a forms Data Entry Line.

- BEGIN-ENTRY must follow the ENTRY and INQUIRY verbs for a Data Entry Line.

*END-ENTRY***END-ITEM**

The END-ITEM verb enables you to signal the end of the generated code for an item. It is generated only if the SKIP-ITEM phrase was used for this item.

- This verb must be used with the SKIP-ITEM verb.

*END-ITEM***END-LABEL**

The END-LABEL verb enables you to test the GUI switch. If a GUI is in use, it will PRINT ?>!. This verb is used after a label is printed. It marks the end of the label for the Windows program.

- This is used after a PRINT statement for a check box, group box, or radio button.

*END-LABEL***ENTRY**

The ENTRY verb enables you to create fields that permit users to enter data on a form. These fields are called data entry fields or unprotected fields. The ENTRY verb also causes the current value for the field to be displayed beneath it, unless the field is a Key field.

- Your program must execute an UPDATE verb before it executes the ENTRY verb.
- If you include the optional AUTO-HEADERS and START-LINE verbs in your program, they must be executed before the ENTRY verb.
- Place fields from different segments on separate lines unless they have the same segment Key elements.
- Use the START-LINE and REPEAT verbs with the ENTRY verb if you want the system to locate a stacked segment. Otherwise, the ENTRY verb assumes that you are pointing to the correct segment and position in the record.
- Use the NEXT-LINE verb to mark the end of an ENTRY line.

**ERRORS-EXIST**

The ERRORS-EXIST verb is used to check if any system level errors have occurred. This verb checks to see if SCREEN-ERROR equals an '@' at-sign.

*IF ERRORS-EXIST*

*Imperative Statement . . .*

## **FIND**

The FIND verb enables you to locate a specific occurrence of a segment.

- Follow a FIND statement with an IF FOUND or IF NOT-FOUND statement to test the result of the FIND operation.
- Unless you use one of these valid qualifiers, FIND begins at the first segment in the pointer:
- FROM HERE—Begins the FIND operation where the pointer is currently positioned. Be sure that you know the position if you use this qualifier.
- STARTING WITH—Begins the FIND operation at a specified key value.

## **FORCE-OK-CANCEL**

The FORCE-OK-CANCEL verb will activate the OK and CANCEL buttons in a GUI environment and force the "Save Changes?" prompt to appear.

FORCE-OK-CANCEL

## **FORMAT-CENTURY**

The FORMAT-CENTURY verb is used to format the century of a date. This verb should be used when you need to associate a century to a literal or a RUNREP date, particularly a date that is to be used as a 21st century date. This verb will assume that all dates that have a year within the range of 00-25 will be a 21st century date. The date to be formatted with the century must be moved in to WORK-DATE. Following the execution of FORMAT-CENTURY the new formatted date will reside in WORK-DATE.

For example, if a date were used in a report and supplied to the report via the RUNREP form and needed a 21st century reference, the coding may be as follows:

```
MOVE SPECIAL-YYMMDD TO WORK-DATE.  
FORMAT-CENTURY.  
MOVE WORK-DATE TO SPECIAL-YYMMDD.
```

## **FORMAT-INTER-DATE**

The FORMAT-INTER-DATE is used to convert a date stored in the YYMMDD format into an internationally accepted format of DAY MONTH YEAR. As an example, the date of January 1, 1999 would appear as 01 Jan 99 when printed on a form or report. To convert a date into this format it is necessary to, first, move the date into WORK-DATE and then execute the FORMAT-INTER-DATE verb. The result will reside in INTERNATIONAL-DATE.

```
MOVE SPECIAL-YYMMDD TO WORK-DATE.  
FORMAT-INTER-DATE.  
PRINT INTERNATIONAL-DATE.
```

**FORMAT-INTER-DATE-CC**

The **FORMAT-INTER-DATE-CC** is used to convert a date stored in the **YYMMDD** format into an internationally accepted format of **DAY MONTH CENTURY YEAR**. As an example, the date of January 1, 1999 would appear as **01 Jan 1999** when printed on a form or report. To convert a date into this format it is necessary to, first, move the date into **WORK-DATE** and then execute the **FORMAT-INTER-DATE-CC** verb. The result will reside in **INTERNATION-DATE-CC**.

*MOVE SPECIAL-YYMMDD TO WORK-DATE.*

*FORMAT-INTER-DATE-CC.*

*PRINT INTERNATION-DATE-CC.*

**GO-TO-NEXT-SEGMENT**

The **GO-TO-NEXT-SEGMENT** verb is used in forms that selectively display inquiry records. If no occurrences have been selected the **GO-TO-NEXT-SEGMENT** should be used.

Examples of this verb's usage is in the activity forms (**01-SCR**, **04-SCR**, **08-SCR**, **95-SCR** and **96-SCR**).

*GO-TO-NEXT-SEGMENT.*

**GRAYBUTTON**

The **GRAYBUTTON** verb, when a GUI is in use, allows you to indicate that the previous two positions in the **SCREEN** area contain the ID number of a pushbutton.

**GRAYBUTTON** moves **?@>'** to the **SCREEN** area to signal the GUI logic to create the pushbutton as a gray non-selected button.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of **?00'** to **?19'**.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

**GUI-IN-USE**

The **IF GUI-IN-USE** verb tests whether a graphical user interface (GUI) is in use. It tests **W7-01-458** for greater than a space. If true, it means that this user has a graphical interface and the form is in graphics mode.

- This verb should be used only after the **NEW-SCREEN-STYLE** verb has been executed, as that is when **W7-01-458** is set to its proper value.

### GUI-ONLY

The GUI-ONLY verb enables you to write a form which will not run in a non-GUI environment. It tests W7-01-464 for greater than a space. If false, an error message is displayed and a RETURN is executed.

- This verb should be placed after sequence #00010, but before all other Cyborg Scripting Language/English Language (CSL/EL) statements.

### INDENTBOX

The INDENTBOX allows you to indicate that the previous five characters in the SCREEN area contain the height, width, and color of a rectangle that is to be drawn. INDENTBOX moves ?,>' to the SCREEN area. The GUI logic looks for the literal ?,>' and creates the rectangle.

- This verb should be used only when a GUI is in use.
- The 'hhwwc' represents the height (in rows), the width (in columns), and the interior color of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- The color values are:

0= transparent	5= yellow
1= red	6= light blue
2= green	7= reserved
3= blue	8= reserved
4= purple	9= entire rectangle is invisible (tab group)

### INQUIRY

The INQUIRY verb enables you to create protected, or display-only, fields. These fields do not permit users to enter data in them. They generally display the current or default value for the field.

- You can place up to 78 characters, including spaces, on each form line.
- The system must execute a READ or UPDATE verb before the INQUIRY verb.
- You can place both INQUIRY and ENTRY fields on the same form line.
- If you do not want to display data from the first occurrence of a segment
- Precede the INQUIRY verb with a FIND statement.
- Use the START-LINE/REPEAT verbs and precede the INQUIRY verb with an ENTRY statement that contains all the key fields for the segment.
- Use INQUIRY-EMPLOYEE to display a name in last name, first name format. Position the output in the first 48 positions on the form line to allow for a name up to 30 characters long.

*INQUIRY field-name-1 literal.field-name-2...*

### INQUIRY-MODE

The INQUIRY-MODE verb is used in an IF statement to check if the user selected inquiry mode. If CODE-1 (W7-01-252) equals an 'I' a true condition will be met.

IF INQUIRY-MODE*Imperative Statements. . .***NEW-SCREEN-STYLE**

The NEW-SCREEN-STYLE verb signals the COBOL code ENTRY verb logic that a non-key field may have the existing value of the field within the entry box. It moves an ?N' to W7-01-460. This verb should be used in all new style forms, that is, those using the Form Appearance Table (SAT). The switch at W7-01-460 is initialized to an ?O' by CYB90 before any form is called.

- This must be executed before any SCREEN-SECTION verbs.

**NEXT-LINE**

The NEXT-LINE verb marks the end of a formatted line and tells the system to display that line on a terminal (output). It also acts as a carriage control indicator.

- End every form line with at least one NEXT-LINE verb, including prompts, entry and inquiry lines, and the Form Title.
- NEXT-LINE moves the form pointer to the beginning of the next form line.
- You can use NEXT-LINE to format a blank display line, or create double-spacing between lines by repeating the NEXT-LINE verb.

**NO-INQUIRY-SELECT**

The NO-INQUIRY-SELECT verb is used in conjunction with the GO-TO-NEXT-SEGMENT verb in the inquiry/selection (SCREEN-SECTION '8') of certain forms. This verb will check to see if any lines have been displayed for this section and if none have, the entry version of the form will be returned.

NO-INQUIRY-SELECT.**NO-SAVE-CANCEL**

The NO-SAVE-CANCEL verb enables you to inactivate buttons labeled OK and Cancel at the bottom of a form. It places an ?N' in column three of row two if a GUI is in use. This signals the Windows code to not activate the ?OK' and ?Cancel' buttons at the bottom of the form when anything is changed.

- This must be executed prior to exiting form.
- If using form-sections, this must be executed after the SCREEN-SECTION verb.

## **PRINT**

The PRINT verb moves field data and literal values to Pointer 11 (SCREEN). Unlike the OUTPUT verb, the PRINT verb edits fields according to the edit length and edit routine specified on the Field Name Table.

- PRINT does not edit fields unless they have a specific edit length and edit pattern.
- You can use PRINT to cause field headings and spaces between fields to display on a form.
- If you use PRINT to move data, you must specify O in the PRINT TOTAL field on the RTEDIT form.

## **PRINT-MESSAGE**

The PRINT-MESSAGE verb is used to access and display memo messages.

## **PRINT-REJECT**

The PRINT-REJECT verb is used to access and display reject and file error messages.

## **PRINT-WARNING**

The PRINT-WARNING verb is used to access and display warning messages.

## **PROCESS**

The PROCESS verb enables you to establish a process loop. A process loop is a series of statements that are executed repetitively. Process loops are an effective way to check multiple-occurrence segments, because the PROCESS logic is executed for each occurrence of a particular segment code.

- If you do not specify an optional qualifier, the PROCESS verb begins at the first occurrence in the stack and continues until it processes all occurrences.
- Qualifiers restrict the process to specific start and end points. Valid qualifiers are:
  - FROM HERE—Begins processing where the pointer is positioned when the program encounters the PROCESS verb.
  - STARTING WITH—Begins processing at a specified key value.
  - ENDING WITH—Ends processing at a specified key value.
- Do not execute a RETURN verb in a process loop.
- You can use the BYPASS-ENTRY verb in a process loop. This verb causes processing to advance to the next occurrence.

**PUSHBUTTON**

The PUSHBUTTON verb, when a GUI is in use, allows you to indicate that the previous two positions in the SCREEN area contain the ID number of a pushbutton. PUSHBUTTON moves ?;>' to the SCREEN area to signal the GUI logic to create the pushbutton.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of ?00' to ?19'.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

**QUERY-FIRST-PASS**

The QUERY-FIRST-PASS verb is used with a conditional IF statement to determine the timing for initialization or any preprocessing tasks.

**QUERY-HEADERS**

The QUERY-HEADERS verb instructs the program to use default heading information for the fields as defined on the Field Name Table. If used this verb must be specified before any fields are displayed on the form.

- The headings are not painted until the INQUIRY verb is used.

**QUERY-ONLY**

The QUERY-ONLY verb instructs the system to inhibit execution of the program to only the QUERY Facility. This statement should be the first command coded in a query program.

**RAISEDBOX**

The RAISEDBOX allows you to indicate that the previous five characters in the SCREEN area contain the height, width, and color of a raised rectangle that is to be drawn.

- This verb should be used only when a GUI is in use.
- The 'hhwwc' represents the height (in rows), the width (in columns), and the interior color of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- See INDENTBOX for color values.

**READ-COMPANY**

The READ-COMPANY verb is used to read the company record data into working storage, and allow inquiry of the data. READ-COMPANY uses the data in the CONTROL-1-2 field as the Key to the record.

### **READ-EMPLOYEE**

The READ-EMPLOYEE verb is used to read the employee record data into working storage, and allow inquiry of the data. READ-EMPLOYEE uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

### **READ-TAXES**

The READ-TAXES is used to read the tax record into working storage, and allow for inquiry of the data. READ-TAXES uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

### **RECTANGLE**

The RECTANGLE verb allows you to indicate that the previous four characters in the SCREEN area contain the height and width of a group box that is to be drawn. RECTANGLE moves '?!>' to the SCREEN area. The GUI logic looks for the literal '?!>' and creates the group box.

- This verb should be used only when a GUI is in use.
- The '?hhww' represents the height (in rows) and the width (in columns) of the rectangle. The rectangle is drawn with its upper left corner replacing these four characters.
- To label the group box, follow the RECTANGLE verb with a PRINT statement for a literal ending with a greater than (>) symbol. The first position of the literal must not be a space.

### **REPEAT :n TIMES**

The START-LINE/REPEAT :n TIMES verb combination:

- Indicates the start of a line of form fields.
- Locates the correct segment in a FILE02 record.
- Specifies the number of times that a line of data should be repeated on a form.
- Determines the end of a line of fields to be displayed on a form.
- If you do not use the START-LINE/REPEAT :n TIMES verb combination, the ENTRY or INQUIRY verb assumes that you are pointed at the correct segment and position in the FILE02.
- To produce a single form line, code REPEAT :0 TIMES to end the form line.
- The code for each form line must end with at least one NEXT-LINE verb.
- You can use the verb BEGIN-ENTRY in place of the AUTO-HEADERS and START-LINE verb combination. END-ENTRY can replace REPEAT :n TIMES.

### **RESET-RECORD-UPDATE**

The RESET-RECORD-UPDATE verb is used to refresh a form. A space will be moved into the RECORD-UPDATED switch (W7-01-097) and an 'S' will be moved into the COMM-ACTION field (W7-01-301).

RESET-RECORD-UPDATE.

**RESET-SCREEN-STYLE**

The RESET-SCREEN-STYLE verb is used to reset a form to a non-GUI form after the NEW-SCREEN-STYLE verb has been used.

RESET-SCREEN-STYLE.

**SAVE-CANCEL**

The SAVE-CANCEL verb enables you to activate buttons labeled OK and Cancel at the bottom of a form. It places an ?S' in column three of row two if a GUI is in use. This signals the Windows code to activate the ?OK' and ?Cancel' buttons at the bottom of the form when anything is changed. The Generate Code from Form Appearance Table program (GENER8) generates this verb for any form section containing an ENTRY field.

- This must be executed prior to exiting form.

**SCREEN-SECTION**

The SCREEN-SECTION verb is used to display a specific section of a form that was created using Form Builder. This command MUST be on its own line. No other code can appear before or after this verb on the same line. This command cannot be used within an IF statement.

SCREEN-SECTION 'X'. 'X' represents the form section from the Form Builder.

**SELECTION-MODE**

The SELECTION-MODE verb is used in an IF statement to check if the user selected the selection mode. If AUTO-KEY-SWITCH (W7-01-448) equals a 'Z' a true condition will be met.

IF SELECTION-MODE

*Imperative Statement . . .*

**SET SCREEN**

The SET SCREEN TO :nn verb changes the current or active address TO :nn of Pointer 11 (SCREEN).

- Positions are counted relative to one (1).
- SET manipulates the fields on the Pointer Table. Therefore, it is important to know where to position the address before you code the SET statement.

**SET-2ND-PANEL**

The SET-2ND-PANEL-UPDATE verb is used in multi panel-UPDATE forms. This verb will force the 2nd panel to update and return the 2nd panel form to the terminal.

SET-2ND-PANEL-UPDATE.

### **SET-AUTO-KEY**

The SET-AUTO-KEY verb will move a 'Y' to W7-01-448. This verb is used in forms that have no key fields to insure the data entered for one employee is not passed to the next employee. This verb should be executed in the prompts paragraph.

SET-AUTO-KEY.

### **SET-FIELD-ERROR**

The SET-FIELD-ERROR verb is used during the field error subroutine. If an error is to be produced by the FIELD-EDIT-ROUTINE this verb sets the indicators to display the message.

This verb move an '@' at-sign into SCREEN-ERROR, W8-01-480, and W8-01-330.

SET-FIELD-ERROR.

### **SET-FOR-MESSAGES**

The SET-FOR-MESSAGES verb issues 0, 1, or 2 NEXT-LINEs, depending on how close to line 24 you are. You should issue it just before any relational edits are done which might cause error messages. When SET-FOR-MESSAGES is used, the position at which the printed message will appear is set at 0, 1 or 2 lines below the last form field. The number of lines depends on how much room is left on the form after the last field.

- This is used after an ENTRY verb or a SCREEN-SECTION containing an ENTRY verb.

### **SET-GRAY**

The SET-GRAY verb moves '?@' to BUTTON-STATE (W8-01-394).

- These verbs are used to set the state of a user defined pushbutton.

### **SET-NORMAL**

The SET-NORMAL verb moves '?;' to BUTTON-STATE (W8-01-394).

- These verbs are used to set the state of a user defined pushbutton.

### **SET-PGxx-y**

SET-PGxx-y is a group of verbs where 'xx' is either UP or DN, and 'y' is either GRAY or NORMAL. The use of one of these verbs forces a PGUP or PGDN button on the button bar to an 'on' (normal) or 'off' (gray) setting.

- Typically, these would not be used; the SET-nn-UPDN verbs would be automatically generated by the GENER8 program.
- If used, these would be used only to code utilities. They would not be used for standard forms, that is, employee- or company-level form.

The SET-PGDN-GRAY verb sets column one of line two to a 'G'.

The SET-PGDN-NORMAL verb sets column one of line two to a 'Y'.

The SET-PGUP-GRAY verb sets column two of line two to a 'G'.

The SET-PGUP-NORMAL verb sets column two of line two to a ?Y'.

**SET-PGDN-OFF**

The SET-PGDN-OFF verb will set the left arrow button into a non-selective state.

SET-PGDN-OFF.

**SET-PGUP-OFF**

The SET-PGUP-OFF verb will set the right arrow button into a non-selective state.

SET-PGUP-OFF.

**SET-SCREEN-TO-ENTRY**

The SET-SCREEN-TO-ENTRY verb forces the form into entry mode.

SET-SCREEN-TO-ENTRY.

**SET-SCREEN-TO-INQRY**

The SET-SCREEN-TO-INQRY verb forces the form into inquiry mode.

SET-SCREEN-TO-INQRY.

**SET-nn-UPDN**

SET-nn-UPDN is a group of verbs where ?nn' is the number of a pointer for an application segment which can occur multiple times.

The Generate Code from Form Appearance Table program (GENER8) generates this verb after the ?REPEAT :0 TIMES' verb. It determines if there are additional segments of the same type, both up and down from the current occurrence. Then, if a GUI is in use and there is more than a single occurrence, columns one and two of the second line of the form are set to ?Y' (normal) or ?G' (gray). The Windows logic will generate pushbuttons labeled Page Up, Page Down, and Select. If column two of line two is a ?G' the Page Up button will be grayed.

If the program contains more than one page, the PGDN button will be normal when a page other than the last page is displayed. The PGUP button will be normal when a page other than the first page is displayed.

- This is automatically generated by the GENER8 program.

**SKIP-ITEM**

The IF [condition] SKIP-ITEM verb allows control to be transferred down to the next END-ITEM statement. This verb is generated by the Generate Code from Form Appearance Table program (GENER8).

- This verb may be used to skip over form formatting and positioning verbs such as ENTRY, PRINT, and SET SCREEN. It is not recommended as a way of skipping over other types of verbs.

### **SPACE-OVER :nn**

The SPACE-OVER :nn verb moves the specified number of spaces to Pointer 11 (SCREEN).

- :nn is a two-digit numeric literal value from 01 through 60.

### **START-LINE**

The START-LINE/REPEAT :n TIMES verb combination:

- Indicates the start of a line of form fields.
- Locates the correct segment in a FILE02 record.
- Specifies the number of times that a line of data should be repeated on a form.
- Determines the end of a line of fields to be displayed on a form.
- If you do not use the START-LINE/REPEAT :n TIMES verb combination, the ENTRY or INQUIRY verb assumes that you are pointed at the correct segment and position in the FILE02.
- To produce a single form line, code REPEAT :0 TIMES to end the form line.
- The code for each form line must end with at least one NEXT-LINE verb.
- You can use the verb BEGIN-ENTRY in place of the AUTO-HEADERS and START-LINE verb combination. END-ENTRY can replace REPEAT :n TIMES.

### **STORE-INQUIRY-ONLY**

The STORE-INQUIRY-ONLY verb will move the content of the COMM-CHAR field (W7-01-298) in to W6-01-298. It is used to test for an inquiry only condition when executing a table form.

STORE-INQUIRY-ONLY.

### **TIME-TO-PRINT-TITLE**

The TIME-TO-PRINT-TITLE verb is used with a conditional IF statement to determine the timing for printing the Query Title at the top of each form.

### **UPDATE-COMPANY**

The UPDATE-COMPANY verb is used to read the company record data into working storage, and allow updating of the data. UPDATE-COMPANY uses the data in the CONTROL-1-2 field as the Key to the record.

### **UPDATE-EMPLOYEE**

The UPDATE-EMPLOYEE verb is used to read the employee record data into working storage, and allow updating of the data. UPDATE-EMPLOYEE uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

### **UPDATE-TAXES**

The UPDATE-TAXES verb is used to reads the tax record into working storage, and allow updating of the data. UPDATE-TAXES uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.

**USER-BUTTON**

The USERBUTTON verb, when a GUI is in use, allows you to indicate that the previous two positions in the SCREEN area contain the ID number of a pushbutton. USERBUTTON moves BUTTON-STATE followed by '>' to the SCREEN area to signal the GUI logic to create the pushbutton.

- For a given form, each pushbutton must be assigned a unique ID number.
- The pushbutton ID number must be in the range of ?00' to ?19'.
- The optional button label must not exceed 20 characters.
- The size of the pushbutton is based on the size of the button label. If there is no label, the button will be three columns wide and one row high. If the label is one to ten characters in length the button will be ten columns wide and one row high.

**VALUE-BELOW-BOX**

The VALUE-BELOW-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

VALUE-BELOW-BOX.ENTRY field-name-1 literal...field-name-2...

**VALUE-IN-BOX**

The VALUE-IN-BOX verb is used immediately before an ENTRY verb for a single field to cause the current field value to appear in the requested position.

- For pointer 42 fields, the default is EMPTY-BOX.
- For other pointers, the defaults are:
  - VALUE-IN-BOX for converted forms, and
  - VALUE-BELOW-BOX for unconverted forms.

VALUE-BELOW-BOX.ENTRY field-name-1 literal field-name-2...

**WARNINGS-EXIST**

The WARNINGS-EXIST verb is used to check if any warnings exist. This verb will check if SCREEN-WARNING (W7-01-096) equals a 'W'.

IF WARNINGS-EXIST

*Imperative Statement . . .*

---

**NOTES**

---



**A**

ADDITIONAL-KEY ..... 281  
 ADDL-KEY-PROGRAM ..... 281  
 Alphanumeric field ..... 117  
 Alternate key utilities ..... 215  
 Alternate keys ..... 209  
 APPL-INDICATOR ..... 281  
 AUTOHEADERS-SWITCH ..... 281  
 AUTO-KEY-SWITCH ..... 281

**B**

BACKREF-HOLD ..... 281  
 BALNCE-NUMBER ..... 281  
 Batch layout section header ..... 23, 55  
 Big option list ..... 39  
 Build Alternate Keys program  
 (KEY-00) ..... 211  
 Button bar ..... 19

**C**

CALL-STACK ..... 281  
 Century date field ..... 125  
 Change the composite view ..... 53  
 Check Box item ..... 47  
 Check for system errors ..... 189  
 Check for user input ..... 189  
 CODE ..... 282  
 Code description field ..... 121  
 Code field ..... 121  
 CODE-1 ..... 282  
 CODE-2 ..... 282  
 CODE-SET-SWITCH ..... 282  
 COLOR-INDICATOR ..... 282  
 COMM-CHAR ..... 282  
 COMM-MODE ..... 282  
 COMPANY-NUMBER ..... 283  
 Context-sensitive menus ..... 157  
 CONTROL-COUNTRY ..... 283  
 Course introduction ..... 3  
 Course logistics ..... 5  
 Course materials ..... 7  
 Create a FIND- verb ..... 163  
 Create A FIND Verb (MKVERB) ... 163  
 Creating messages ..... 183  
 Creating option lists ..... 99  
 Creating report parameter entry forms 195  
 Creating the screen footer ..... 293  
 Cross Reference Data Entry form  
 (D-XREF) ..... 165  
 CSL Compiler program (RELOAD) .. 82  
 CURRENT-DATE ..... 283  
 CURRENT-DATE-CYYMDD ..... 283  
 CURRENT-TIME ..... 283

**D**

Data entry lines ..... 75  
 Data inquiry lines ..... 75  
 Data requirements analysis ..... 141  
 Data verification ..... 77  
 DATE-FORMAT ..... 283  
 Delete This Entry function  
 (ZDELETE) ..... 265  
 Design application modes ..... 25  
 Design applications and Cyborg  
 Scripting Language ..... 13  
 Designing the form ..... 27  
 DISPLACEMENT ..... 105  
 DISPLAY LENGTH ..... 107  
 Display message ..... 191  
 Displaying messages ..... 189  
 Documenting messages ..... 187  
 Message documentation  
 conventions ..... 187  
 Sequence number convention ..... 187  
 Edit command list ..... 272  
 Editing ECM records ..... 79

**E**

ENTRY LENGTH ..... 107  
 Error messages ..... 185  
 Displaying message ..... 189  
 Documenting messages ..... 187  
 Message conventions ..... 185  
 ERRORS-EXIST ..... 77  
 Exit routines ..... 31, 37  
 Extracting the form ..... 81  
 Extracting the form in batch ..... 85

**F**

Field definitions ..... 101  
 Field documentation ..... 127  
 Field error subroutine ..... 193  
 Field maintenance ..... 103  
 Field Maintenance And Edit form  
 (F-NAME) ..... 103  
 Field name table ..... 101  
 Field reports ..... 97  
 Field Table List (FTLIST) ..... 97  
 Field Table Menu (F-MENU) ..... 97  
 Field to Program Cross Reference  
 (CROSSX/CROSSP) ..... 97  
 F-NAME fields ..... 103  
 Footers ..... 21  
 Form body items ..... 33  
 Double Column ..... 33  
 Single Column ..... 33  
 Form body layout ..... 33  
 Form Builder ..... 17, 25

- 
- Form design ..... 27  
Form design steps ..... 17  
Form display modes ..... 15  
Form sections ..... 15  
form title ..... 29  
format of the Select/Inquiry Form  
    section ..... 57
- G**  
GENER8 ..... 87  
Generate Form Appearance Code  
    program (GENER8) ..... 82  
Generating the form ..... 83  
Generating the form in batch ..... 87  
Get Copy of Screen Appearance  
    Table program (GETSAT) ..... 81  
Grouping ..... 49  
Grouping buttons ..... 45
- H**  
HEADER SWITCH ..... 113  
Headers ..... 21  
Help form ..... 265  
HIGH-VALUE ..... 283  
HORIZONTAL ..... 283  
Key field ..... 115  
Key fields ..... 33  
KEY-00 ..... 219  
KEY01-AREA ..... 284  
KEY01-SIZE ..... 284  
KEY02-AREA ..... 284  
KEY02-SIZE ..... 284  
KEYDEL ..... 217  
KEY-FIELD ..... 284  
KEY-WORD ..... 284
- L**  
Large segment layouts ..... 155  
LINE-ADVANCE ..... 284  
List box dialog ..... 39  
LOW-VALUE ..... 284
- M**  
MACHINE-SWITCH ..... 285  
Maintaining alternate keys ..... 209  
Maintaining alternate keys in batch ..... 215  
Menu bar ..... 19  
MENU-PAGE ..... 285  
MENU-STOP ..... 285  
Message (M) ..... 181  
Message placement ..... 189  
Modify KEY-00 ..... 211  
Modify QUERY ..... 213  
MORE-HIGH-VALUE ..... 285
- MORE-LOW-VALUES ..... 285  
MORE-SPACES ..... 285  
Moving a group of items ..... 49
- N**  
Name field ..... 119  
New Fields Definition Writer  
    (NEWSCR) ..... 253  
    Action form ..... 255  
    General Information form ..... 257  
    Updating FILECL ..... 265  
NEW-SCREEN-STYLE ..... 73  
Numeric field ..... 117
- O**  
Operating system codes ..... 271  
OPERATING-SYS-CODE ..... 285  
OPERATOR-ID ..... 285  
Option button ..... 43  
OPTION-NUMBER ..... 285
- P**  
PAGE-LENGTH ..... 285  
PAGE-LINES ..... 285  
Partial key field ..... 115  
Pointer 7 field definitions ..... 281  
Pointer 7 fields ..... 278  
Pointer 7 fields in alphabetical order ..... 280  
Pointer 7 fields in pointer order ..... 279  
Primary keys ..... 209  
PRINT-MESSAGE ..... 191  
PRINT-REJECT ..... 191  
PRINT-WARNING ..... 191  
PRODUCTION-VERSION ..... 286  
Program Memory Map (MAPRPT) ..... 97  
Program processing ..... 179  
Program structure ..... 69  
PROGRAM-FIELD ..... 286  
PROPAGATE ..... 111
- Q**  
QM-PHASE ..... 286  
QUERY control record for  
    KEY-00 ..... 219  
QUERY Control Record for  
    KEYDEL ..... 217  
Query keys ..... 209  
Query programming ..... 205  
Query programs ..... 205  
Query verbs ..... 205  
QUERY-ALT-KEY ..... 286  
QUERY-FIRST-PASS ..... 207  
QUERY-FROM-DUP ..... 286  
QUERY-FROM-VALUE ..... 286
-

QUERY-HEADERS ..... 207  
 QUERY-ONLY ..... 207  
 QUERY-PROGRAM ..... 286  
 Query-related programs ..... 163  
 QUERY-SWITCH ..... 286  
 QUERY-TO-DUP ..... 286  
 QUERY-TO-VALUE ..... 286  
 QUOTE ..... 286

**R**

RDBMS field ..... 113  
 RDBMS users ..... 139  
 Record access ..... 71  
 RECORD-LOCKED ..... 287  
 RECORD-NOT-UPDATED ..... 77, 189  
 RECORD-UPDATED ..... 189, 287  
 Regular date field ..... 123  
 Reject (R) ..... 181  
 RELOAD processing ..... 87  
 Reporting parameter selection ..... 195  
 Required field ..... 119  
 ROUNDING ..... 111

**S**

SAVE-SEGMENT-KEY ..... 287  
 Saving the design ..... 49  
 Schedule Report Groups form  
   (RGMSTR) ..... 195  
 Screen footers ..... 293  
 Screen Label To Field Name  
   Cross Reference (FLABEL) ..... 97  
 SCREEN-ERROR ..... 189, 287  
 SCREEN-SECTION ..... 73, 75  
 SCREEN-WARNING ..... 287  
 Section number ..... 21  
 Segment cross-reference ..... 165  
 Segment layout ..... 143  
 Segment layout example ..... 145  
 Segment Layout form (LSEGS) ..... 161  
 Segment Layout Report  
   (SRTFLD/F-SEGM) ..... 97  
 Select/Inquiry section ..... 55  
 Sequence number convention  
   Sequence number convention ..... 185  
 SESSION-ID ..... 287  
 SESSION-STATUS ..... 287  
 SET-FOR-MESSAGES ..... 77, 189  
 Size a group box ..... 45

SPACE ..... 288  
 Spin button ..... 35  
 STAT-KEY ..... 288  
 Status bar ..... 19  
 STORAGE LENGTH ..... 105  
 System Control Repository  
   object codes ..... 270

**T**

Tabbing ..... 49  
 Text box dialog ..... 35  
 The design application window ..... 19  
 The relationship between the  
   application and CSL ..... 67  
 Time-span date field ..... 123  
 TIME-TO-PRINT-TITLE ..... 207  
 TRACE-HOLD ..... 288  
 TRACE-READY ..... 288  
 TRACE-SWITCH ..... 288  
 TRANS-NUMBER ..... 289

**U**

Update FILECL ..... 153  
 Update Form Appearance Table  
   program (PUTSAT) ..... 83  
 UPDATE-COMPANY ..... 71  
 UPDATE-EMPLOYEE ..... 71  
 UPDATE-TAXES ..... 71  
 User-defined alternate keys ..... 211  
 User-defined entry form ..... 157  
 User-defined segments ..... 139  
 Using alternate keys ..... 221  
 Using D-XREF ..... 167  
 Using the action form ..... 255  
 Using the general information form ..... 259  
 Using the new field name entries  
   form ..... 263

**V**

Verify segment layout ..... 161  
 Verify user-defined definitions ..... 159

**W**

Warning (W) ..... 181  
 When shown ..... 37  
 When Shown buttons ..... 31  
 Work fields ..... 193  
 Workspace ..... 19

# Hewitt

---

## **Introduction to Cyborg Scripting Language Participant Guide**

**Document Issue: 5.1**

Document Issue Status:	Sixth Release
Document Issue Level:	5.1
Document Issue Date:	April 2004
Software Version:	5.1

### **Copyright Notice**

Copyright© 2004 by Hewitt Associates LLC. All rights reserved.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series® are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear, but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

---

# Table of Contents

---

<b>Section 1: Course Overview</b> .....	<b>1</b>
Course overview .....	3
Course logistics .....	5
Course materials .....	7
<b>Section 2: The Solution Series Architecture</b> .....	<b>9</b>
Introduction .....	11
The Solution Series architecture .....	13
Program types .....	15
Physical data structures .....	21
Logical data structure .....	35
The Solution Series processing modes .....	41
Section summary .....	49
Section 2 exercise .....	51
<b>Section 3: Cyborg Scripting Language Basics</b> .....	<b>53</b>
Introduction .....	55
CSL programming rules .....	57
CSL syntax conventions .....	61
CSL basic commands .....	63
Creating a CSL program .....	75
Section summary .....	93
Section 3 exercise .....	95
<b>Section 4: The Solution Series Utilities</b> .....	<b>97</b>
Introduction .....	99
Additional edit commands .....	101
Copying System Control Repository data .....	119
Displaying System Control Repository data .....	121
Purging System Control Repository data .....	123
Exporting/Importing System Control Repository data .....	125
Program/Field reports .....	131
Section summary .....	149
Section 4 exercise .....	151
<b>Section 5: Cyborg Scripting Language Plus</b> .....	<b>153</b>
Introduction .....	155
Moving data .....	157
Comparing data .....	171
Calculating .....	183
Transferring control within a program .....	185
Section summary .....	201
Section 5 exercise .....	205
<b>Section 6: Work Areas and Pointers</b> .....	<b>209</b>
Introduction .....	211
Work areas .....	213
Pointer 44 .....	225

Pointer 45 .....	227
Pointer manipulation .....	229
Section summary .....	241
Section 6 exercise .....	243
<b>Appendix A: Exercise Answers.....</b>	<b>245</b>
Section 2 exercise answers .....	246
Section 3 exercise answers .....	247
Section 4 exercise answers .....	248
Section 5 exercise answers .....	251
Section 6 exercise answers .....	253
<b>Appendix B: Extra for Experts.....</b>	<b>255</b>
Introduction .....	256
Hypertext development process .....	257
Scoping the extent of work required .....	257
Windows help .....	258
RTF footnotes .....	260
Windows Help compilation .....	263
Integration—the real scope of customization .....	263
How the Module Code Reference File is used to access context-sensitive help .....	265
How the cyborg.cnt file integrates custom help .....	266
How the A.cnt links the field level help topics .....	267
What to do with *.GID files.....	268
ASCII conversion chart .....	269
Detailed Directions .....	272
Task 1: Designing your custom help .....	273
Task 2: Creating your form level context-sensitive help .....	274
Task 3: Creating your field level context-sensitive help .....	276
Task 4: Place your files in the Support subdirectory .....	278
Task 5: Configuring context-sensitive access to your form help.....	278
Task 6: Integrating your custom program help with the modular help system .....	280
Task 7: Configuring context-sensitive access to your field help .....	281
Task 8: Integrating your custom field topics with field help .....	282
<b>Appendix C: Pointer 7 Fields.....</b>	<b>285</b>
Pointer 7 fields .....	286
Pointer 7 fields in pointer order .....	287
Pointer 7 fields in alphabetical order .....	288
Pointer 7 field definitions .....	289
<b>Appendix D: Quick Solution—Technical Reference Guide.....</b>	<b>299</b>
System Control Repository File object codes .....	300
Operating system codes .....	301
Edit command list .....	302
<b>Index.....</b>	<b>i</b>

## Section 1: Course Overview

---

### Table of Contents

Course overview .....	3
Course logistics.....	5
Course materials .....	7

## Course Overview

- **Purpose and benefits**
- **Audience**
- **Prerequisites**
- **Goals**
- **Expectations**

---

NOTES

---

## Course overview

### **Purpose**

The purpose of this course is to teach you the skills needed to create basic Cyborg Scripting Language (CSL) programs.

### **Benefits**

The benefit of learning this information is that you will acquire experience in the basics of CSL programming.

### **Audience**

This course has been designed for project team members or data processing personnel who are responsible for creating CSL programs.

### **Prerequisites**

Before taking this course, you should have completed the Using The Solution Series: Administrative Solutions or eCyborg: Using the Web Client course.

### **Goals**

This course enables you to:

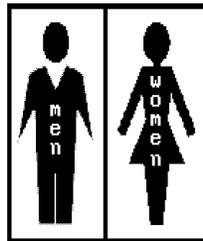
- Describe The Solution Series architecture and related concepts.
- Create CSL programs to display data on a form.

### **Expectations**

To achieve the goals of this course you should:

- Ask questions.
- Share examples of your own experiences. This sharing of information among participants enhances the learning process.
- Ask where to obtain additional information if you have an interest in a point that is introduced.

# Logistics



---

**NOTES**

---

## Course logistics

Use the space below to take notes about the course logistics.

### Meals

---

---

---

---

### Breaks

---

---

---

---

### Telephones

---

---

---

---

### Restrooms

---

---

---

---

### Security questions

---

---

---

---

# Course Materials



---

**NOTES**

---

## Course materials

All form illustrations reflect the current version of The Solution Series. Instructions to access or complete a form are provided using the Navigator.

### Table of contents

Each section has a table of contents listing on the section title page.

### Text layout

This guide is designed in the following manner:

- Upper pages typically contain copies of overhead transparencies or forms.
- Lower pages contain information about the overhead transparency or form and an area for your note taking.

### Section exercise

Exercises give you an opportunity to practice what you have learned in each section. All sections except the course overview section have exercises.

### Appendixes

The appendixes are in the back of your participant guide. Appendixes contain:

- Exercise Answers—Answers to section exercises.
- Extra for Experts—Additional documentation on related topics.
- Pointer 7 Fields—Predesignated fields and constants
- Quick Solution— Technical Reference Guide: codes and commands used by various utilities.

### Index

An alphabetical listing of content cross-referenced to page numbers.

---

**NOTES**

---

## Section 2: The Solution Series Architecture

---

### Table of Contents

Introduction.....	11
The Solution Series architecture.....	13
Program types.....	15
Physical data structures.....	21
Logical data structure.....	35
The Solution Series processing modes.....	41
Section summary.....	49
Section 2 exercise.....	51

## Objectives

- **Identify The Solution Series architecture**
- **Recognize The Solution Series program types**
- **Identify The Solution Series physical structure**
- **Identify The Solution Series logical structure**
- **Identify The Solution Series processing modes**

---

NOTES

---

## Introduction

### **Purpose**

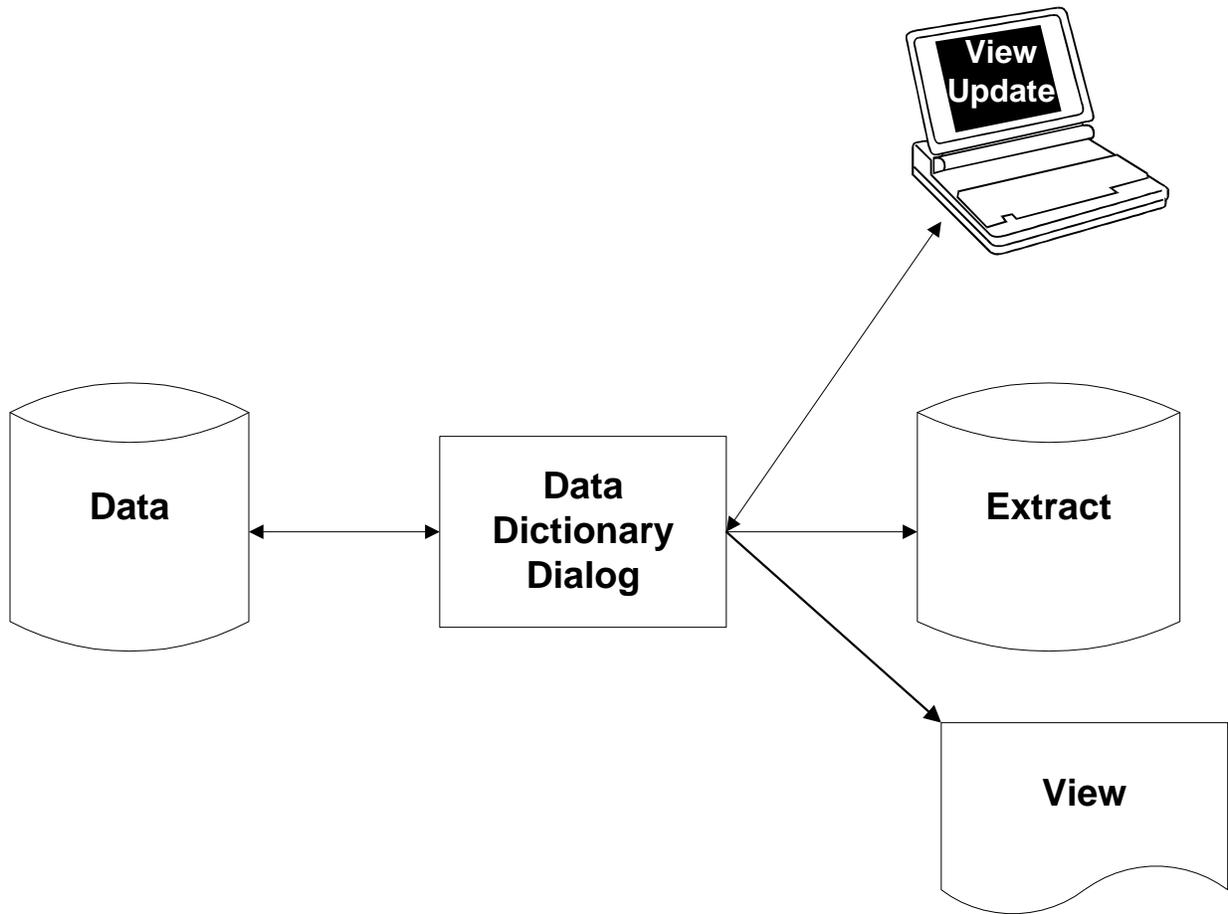
In this section you will be introduced to The Solution Series architecture.

### **Objectives**

Upon completion of this section you will be able to:

- Identify The Solution Series architecture
- Recognize The Solution Series program types
- Identify The Solution Series physical structure
- Identify The Solution Series logical structure
- Identify The Solution Series processing modes

# The Solution Series Architecture



---

## NOTES

---

## The Solution Series architecture

### The Solution Series software

The Solution Series is a tool that manipulates information for the purpose of viewing, updating, processing, or extracting.

The Solution Series software uses three components to accomplish these tasks:

- Dialog—Programming language that allows you to manipulate the data.
- Data—Information you wish to work with.
- Dictionary—Definition of the data fields and programming verbs.

This section discusses the physical and logical structure related to these three components.

## The Solution Series Programs

- **CBSVO**
- **CBSVB**
- **CBSVOT**
- **CBSVBT**

---

**NOTES**

---

## Program types

### Program types

There are two program types used by The Solution Series software:

- COBOL (Common Business Oriented Language)
- Cyborg Scripting Language (CSL)

### COBOL

There are four delivered Solution Series COBOL programs that are specific to each operating system. These programs do not contain application logic and are used to communicate with the operating system and the various input and output devices.

- CBSVO  
The online program that allows real-time, interactive updating of the System Control Repository (FILE01) and the Employee Database (FILE02).
- CBSVB  
The batch program used to produce reports and access/maintain The Solution Series files offline.
- CBSVOT  
The trace version of the CBSVO online program. This program includes diagnostics and is not used in production. It is a program-debugging tool.
- CBSVBT  
The trace version of the CBSVB batch program. This program includes diagnostics and is not used in production. It is a program-debugging tool.

## **Cyborg Scripting Language (CSL)**

- **4th generation language**
- **Specific to Solution Series applications**
- **Generic across operating platforms**
- **Powerful program maintenance utilities**
- **Comprehensive debugging facility**

---

**NOTES**

---

## Program types, continued

### **Cyborg Scripting Language (CSL)**

CSL is a fourth-generation programming language. It shares many features with COBOL. However, CSL is used specifically for coding Solution Series forms, reports, and programs.

CSL is generic to each operating system environment so programs are portable across platforms.

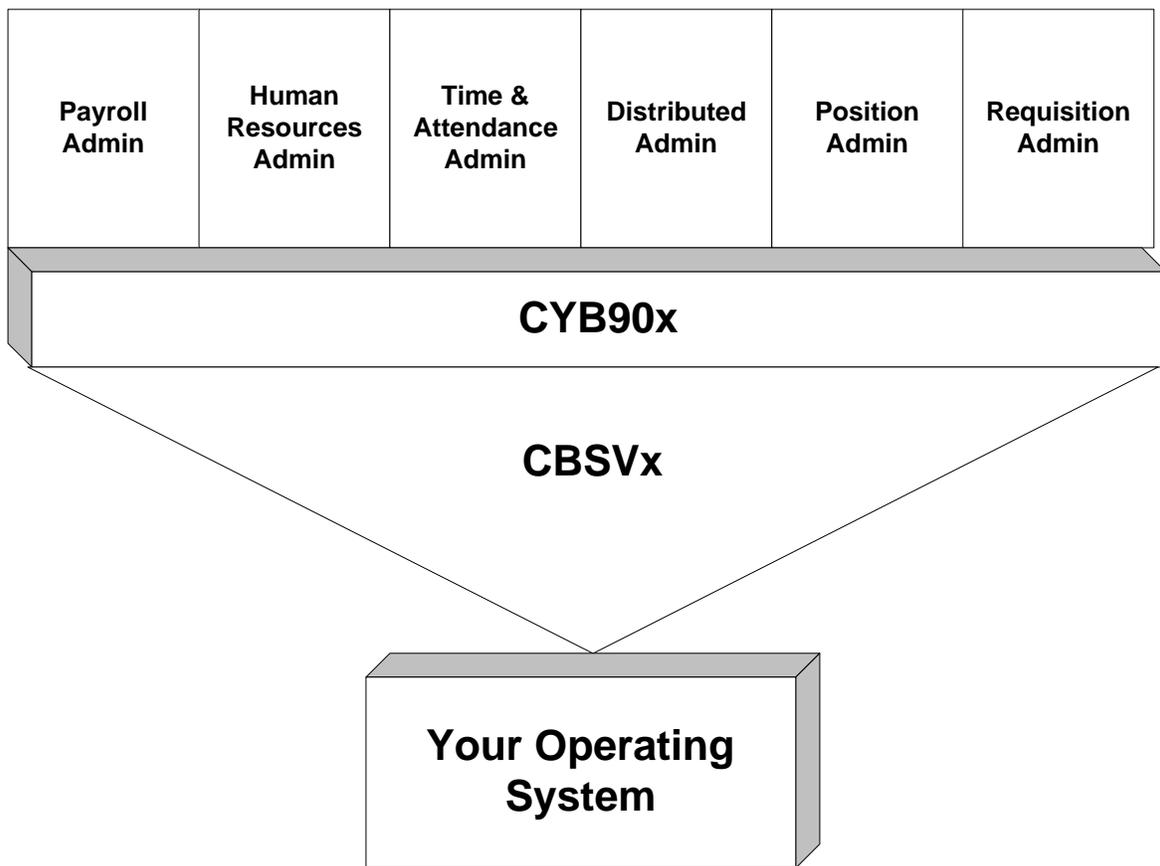
Cyborg Scripting Language has many features that make it a powerful programming tool, including:

- EDIT, a flexible program editor
- Special verbs designed for Solution Series applications
- Extensive error-trapping and debugging facilities
- Program models to create forms and reports

Many users create CSL source code to accomplish two separate but related activities:

- Create a data entry form to store data
- Retrieve the data to include in a report

# Relationship Between CSL and CBSV



---

## NOTES

---

## Program types, continued

### **CBSV and CSL relationship**

The Solution Series COBOL (CBSV) programs interpret the CSL pseudo object code and perform their functions through instructions to the operating system.

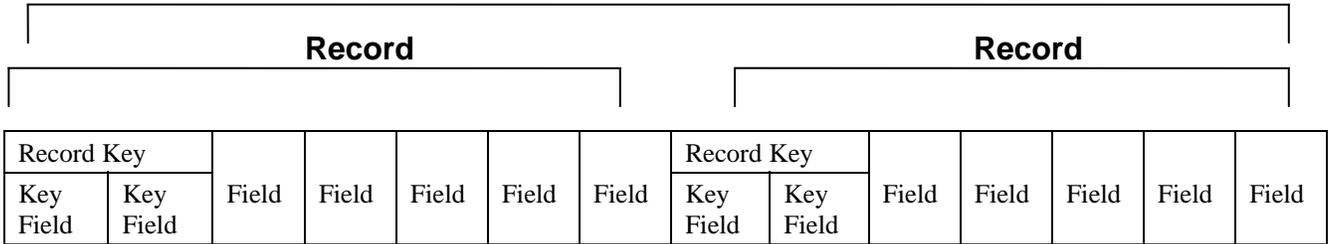
### **Analogy**

The best way to explain the relationship between CBSV programs and CSL programs is to use the following analogy:

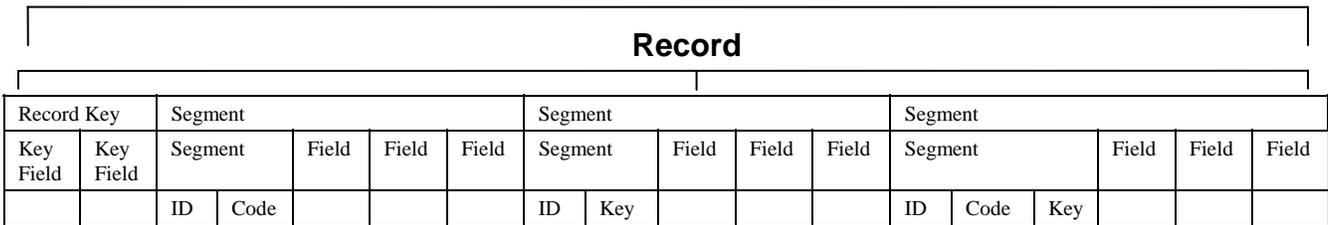
- The CBSV programs behave like a computer. They perform the functions requested by the CSL object code. There is no application logic in the CBSV programs.
- To continue the analogy, there is a CSL program named CYB90x that acts as the operating system for the computer. CYB90x reads the Command Line (or Control Record) and turns control over to the CSL application or utility program requested. When that CSL program is complete, control is passed back to the CYB90x program.
  - The x in CYB90x represents (B) for batch, or (O) for online, dependent on which mode you are processing.

# Physical Data Structures

## System Control Repository (FILE01)



## Employee Database (FILE02)




---

## NOTES

---

## Physical data structures

### File

A file is an overall classification of data pertaining to a specific business use or application.

- System Control Repository (FILE01) contains programs, documentation, option lists, and error messages.
- Employee Database (FILE02) contains your company and employee data.

### Record

A record is a logical grouping of unique information within a file.

- CSL programs reside on the System Control Repository as P (or Program) records.
- Employee data on the Employee Database resides in an Employee Master Record.

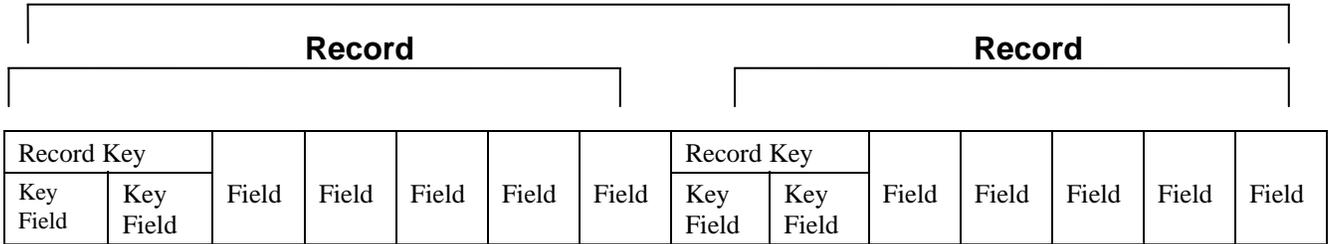
### Record key (primary key)

A record key is the set of data elements the system uses to identify and locate a record. The record key must be unique within a file.

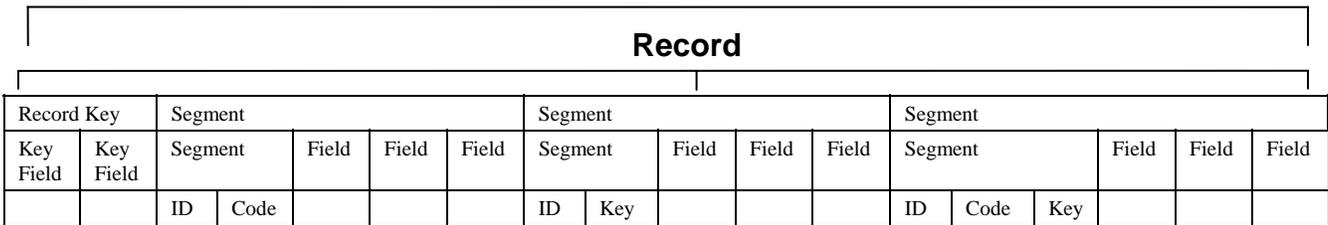
- System Control Repository records include a 1-3 position Record Type Code and up to 23 additional characters of key information (Key length = 24).
- Employee Database records typically include a Organization number value, a Record Type Code and up to 25 additional characters of key information (Key length = 32).

# Physical Data Structures

## System Control Repository (FILE01)



## Employee Database (FILE02)




---

## NOTES

---

## Physical data structures, continued

### Segment

A segment is a subdivision of a record that contains data of a particular kind.

- Segments can be single- or multiple-occurrence and are only applicable to company, employee, and tax data.

### Segment ID

Segment IDs are one-character identifiers that distinguish one segment from another.

- Segment IDs reside in the first position of a segment.

### Segment codes

Some multiple occurrence segments are further subdivided using segment codes.

- Segment codes are two character identifiers that reside in the second and third position of a segment.
- The HRMS Module stores data in the L segment ID and uses segment codes to group HR data.

### Segment Key (Secondary Key)

A segment key is further information that identifies a particular segment occurrence. The segment ID (and possibly segment code) is always part of the segment key.

### Field

A field is a unique piece of data within a record or a segment within a record.

- An employee record is made up of many fields, such as EMPLOYEE-NAME and BIRTH-DATE.
- Field names and their attributes are stored on the System Control Repository in the field name table (Data Dictionary).
- Field data is stored on files in a record.

## The Solution Series Files

File Name	Organization	Assignment	Input/Output	Record Size	Purpose
FILE01	Random	Disk	Input/Output	80	System Control Repository
FILE02	Random	Disk	Input/Output	Variable (3060 max)	Employee Database
FILE03	Sequential	Printer	Output	132	Audit/Report/Message Print File
FILE04	Sequential	Reader	Input	80	Control Records File
FILE05	Sequential	Disk	Input	80	Data Input File
FILE06	Random	Disk	Output	80	Installation Control File
FILE07	Random	Disk	Output	Variable (3060 max)	Installation Master File
FILE10	Sequential	Disk/Tape	Output	80	Data Output File
FILE11	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE12	Sequential	Disk/Tape	Output	256	Payroll Process Batch Master File
FILE13	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE14	Sequential	Disk	Input	150	Report Extract Input File
FILE15	Sequential	Disk	Output	150	Report Extract Output File
FILE17	Sequential	Printer	Output	132	Alternate Print File
FILE18	Sequential	Printer	Output	132	Alternate Print File
FILE19	Sequential	Printer	Output	132	Alternate Print File
FILE23	Random	Disk	Input/Output	80 (Default)	User Defined File
FILE24	Sequential	Disk	Input	512 (Default)	User Defined File
FILE25	Sequential	Disk	Output	512 (Default)	User Defined File
FILE30	Sequential	Disk	Output	320	Savings Bond File
FILE31	Sequential	Printer	Output	132	PAY-CP online check utility

---

## NOTES

---

## Physical data structures, continued

### The Solution Series files

The files listed in the table have been defined by The Solution Series to be used throughout the system by utilities and processing programs. Some additional information you should know about the files include:

#### FILE01

System Control Repository—Contains all application programs, fields, documentation, option lists, tables and other system data.

#### FILE02

Employee Database—Contains company data, employee data, and Cyborg-delivered demonstration data.

#### FILE03

Audit/Report/Message Print File—Contains any error messages, audit trails, and/or reports for the process.

#### FILE04

Control Record File—Used during a batch run to tell the system which program(s) to execute. The format of the Control Record resembles the Command Line of an online form.

#### FILE05

Data Input File—A general-purpose sequential input file. Commonly, it is used to import data.

#### FILE06

Installation Control File—Used to build the System Control Repository (FILE01).

#### FILE07

Installation Master File—Used to build the Employee Database (FILE02).

## The Solution Series Files

File Name	Organization	Assignment	Input/Output	Record Size	Purpose
FILE01	Random	Disk	Input/Output	80	System Control Repository
FILE02	Random	Disk	Input/Output	Variable (3060 max)	Employee Database
FILE03	Sequential	Printer	Output	132	Audit/Report/Message Print File
FILE04	Sequential	Reader	Input	80	Control Records File
FILE05	Sequential	Disk	Input	80	Data Input File
FILE06	Random	Disk	Output	80	Installation Control File
FILE07	Random	Disk	Output	Variable (3060 max)	Installation Master File
FILE10	Sequential	Disk/Tape	Output	80	Data Output File
FILE11	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE12	Sequential	Disk/Tape	Output	256	Payroll Process Batch Master File
FILE13	Sequential	Disk/Tape	Input	256	Payroll Process Batch Master File
FILE14	Sequential	Disk	Input	150	Report Extract Input File
FILE15	Sequential	Disk	Output	150	Report Extract Output File
FILE17	Sequential	Printer	Output	132	Alternate Print File
FILE18	Sequential	Printer	Output	132	Alternate Print File
FILE19	Sequential	Printer	Output	132	Alternate Print File
FILE23	Random	Disk	Input/Output	80 (Default)	User Defined File
FILE24	Sequential	Disk	Input	512 (Default)	User Defined File
FILE25	Sequential	Disk	Output	512 (Default)	User Defined File
FILE30	Sequential	Disk	Output	320	Savings Bond File
FILE31	Sequential	Printer	Output	132	PAY-CP online check utility

---

## NOTES

---

## Physical data structures, continued

### **FILE10**

Output File—A general-purpose sequential output file.

### **FILES 11, 12, and 13**

FILE11, FILE12, and FILE13 are the payroll process' batch master file, commonly referred to as the P20 file. These files are used by the PAYXTR process to prepare the random Employee Database data for the payroll process and by the PAYMRG process to update the Employee Database (FILE02) after the payroll process.

### **FILE14**

Input File—Contains sorted extract data. It is used as input to the print phase of a report process.

### **FILE15**

Output File—Contains extracted data. It is used in the report extract phase to provide selected information for printing.

### **FILES 17, 18, and 19**

Alternate Print Files—These files are similar to FILE03 in that they can be customized to route your output somewhere other than the default location (report processing only).

### **FILES 23, 24, and 25**

User Defined Files—Files that can be activated for whatever purpose you decide in batch and/or online.

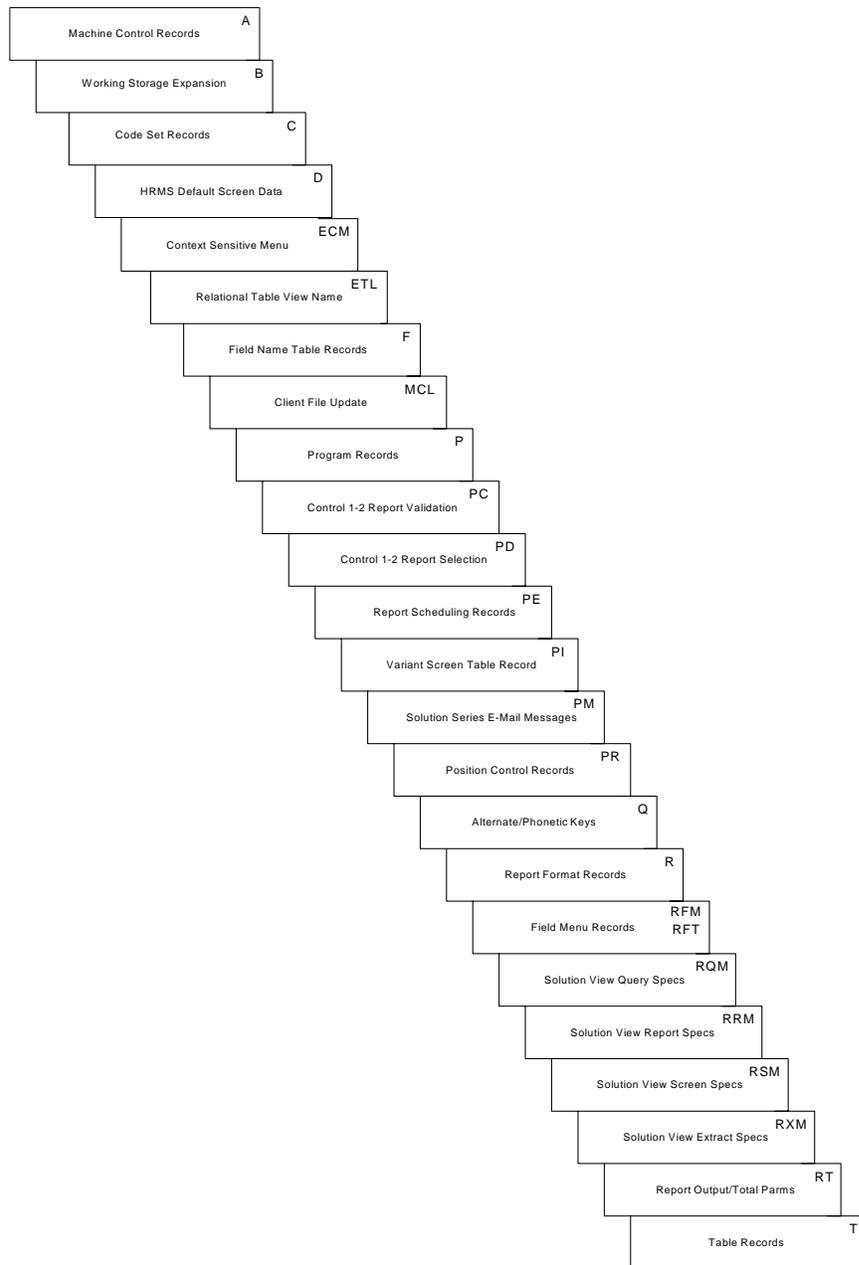
### **FILE30**

Savings Bond tape file—File that is forwarded to the Federal Reserve Board (FRB).

### **FILE31**

PAY-CP online check utility print file.

# System Control Repository Records



---

## NOTES

---

## Physical data structures, continued

### System Control Repository definition

The System Control Repository (FILE01) contains programs, delivered option lists, fields, and any other information needed to make the system run. Any time you sign on to the system, you use the System Control Repository.

### System Control Repository structure

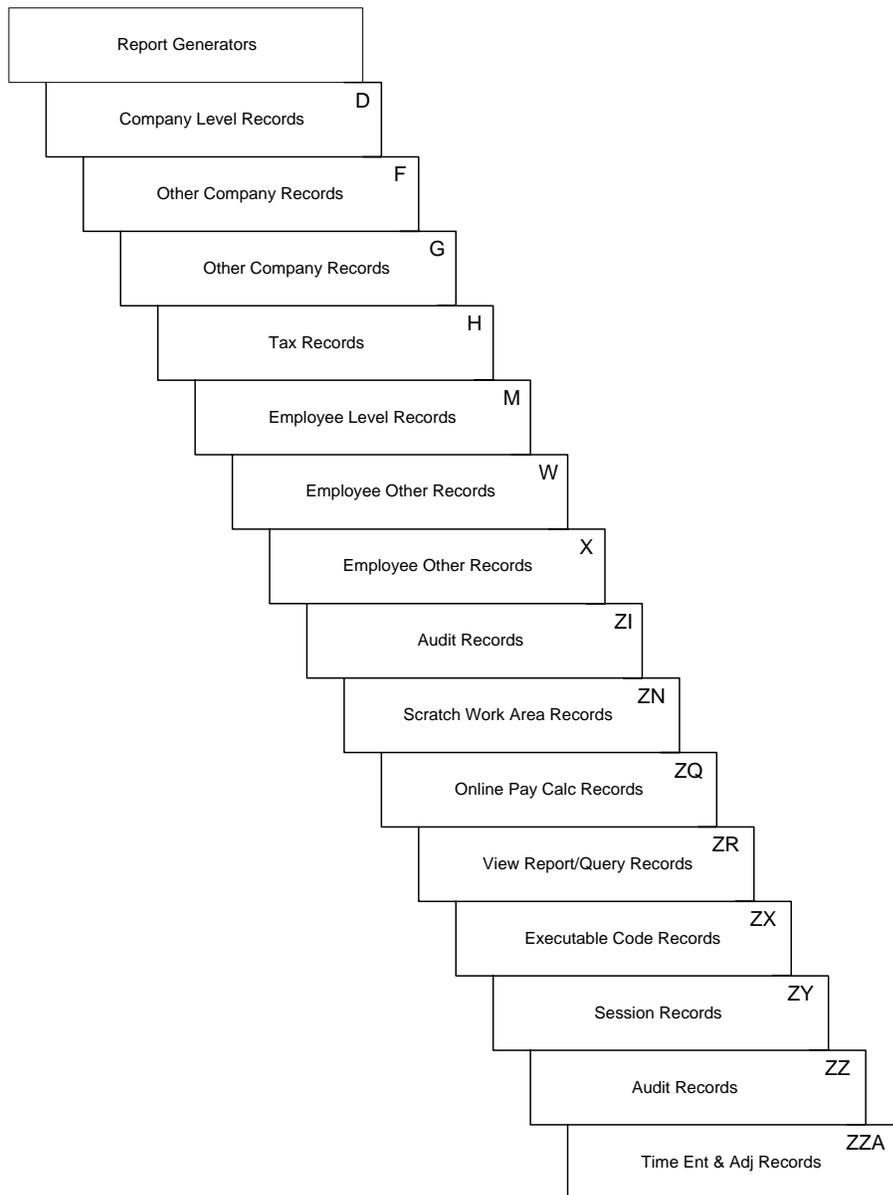
The System Control Repository (FILE01) contains fixed, 80-byte records with a 24-character key.

### System Control Repository records

The System Control Repository consists of several record types with some records having subsidiary record types. The record types that we will be using with CSL programming include:

- C     Option lists
- F     Field Name Table (Data Dictionary)
- P     Program Records (subdivided by type)
- R     Report Format Record
- RT    Report Output Position/Totaling Parameters
- T     Tables

# Employee Database Records



---

## NOTES

---

**Physical data structures, continued****Employee Database definition**

The Employee Database (FILE02) contains your data. Both company level and employee level information are defined and reside on the Employee Database.

**Employee Database structure**

The Employee Database contains variable length records with a maximum length of 3060. The Record Key contains 32 characters. The record key begins in the byte following Cyborg's record length descriptor.

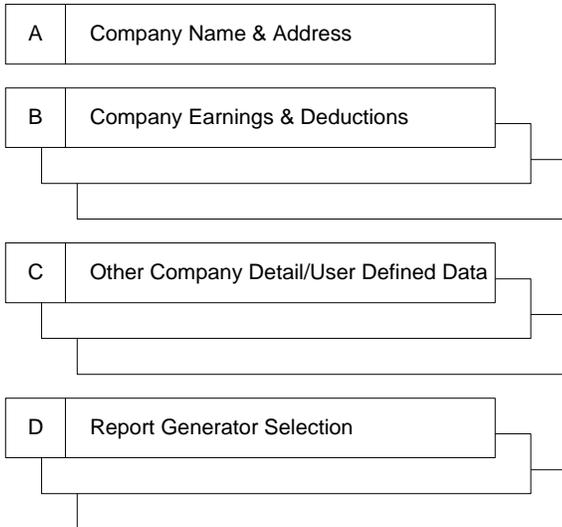
**Employee Database records**

The Employee Database consists of several record types, some of which records have subsidiary record types. The record types that we will be using for CSL programming include (xxxxxx represents the Control 1-2 value that precedes some record types):

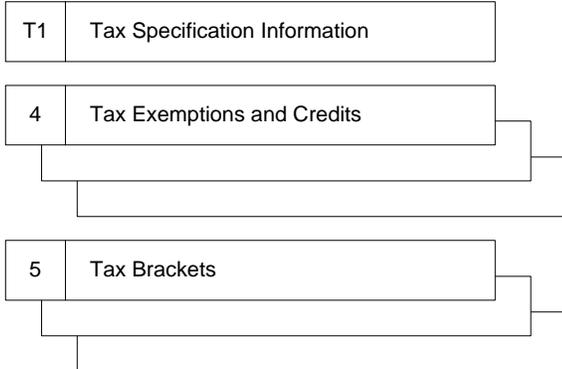
xxxxxxD	Company Records
xxxxxxF	Company Other Records
xxxxxxG	Company Other Records
xxxxxxH	Tax Records
xxxxxxM	Employee Records
xxxxxxW	Employee Other Records
xxxxxxX	Employee Other Records
ZX	Executable Code

# Record Segment Layouts

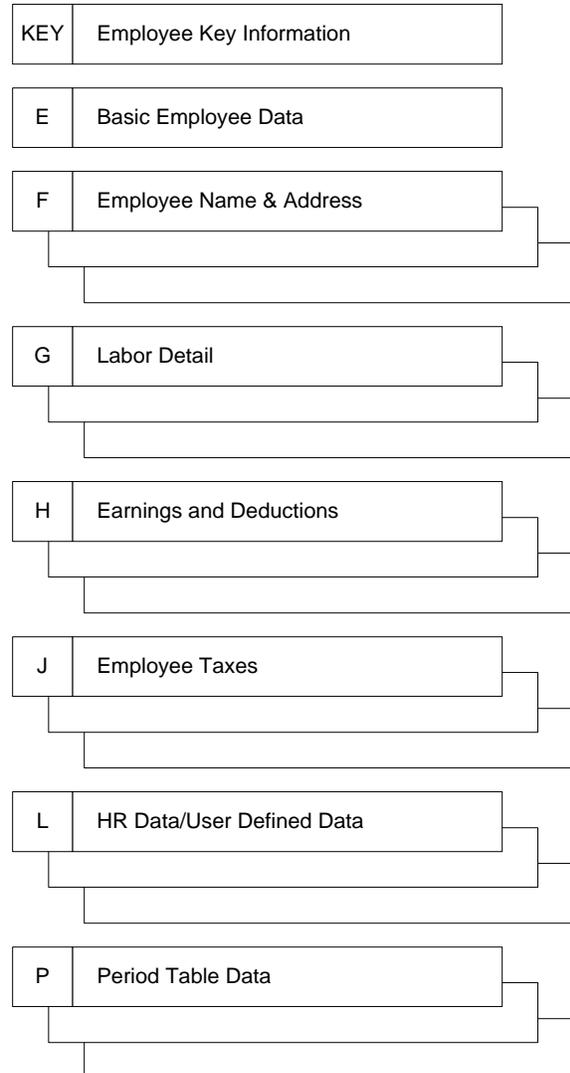
## Company Record Segments, Record Type: D



## Tax Record Segments, Record Type: H



## Employee Record Segments, Record Type: M



---

## NOTES

---

## Physical data structures, continued

### Master record segments

Data on master records (company, tax, and employee records) are subdivided into segments, with each segment containing information of a certain type. Each segment corresponds to a pointer.

There are two types of segments:

- Single-occurrence—Only one occurrence of the data exists for a particular master record.
- Multiple-occurrence (Stacked)—Many occurrences of the segment type can exist for a particular master record.

Stacked segments use segment code and segment key fields to differentiate and locate particular occurrences within the master record.

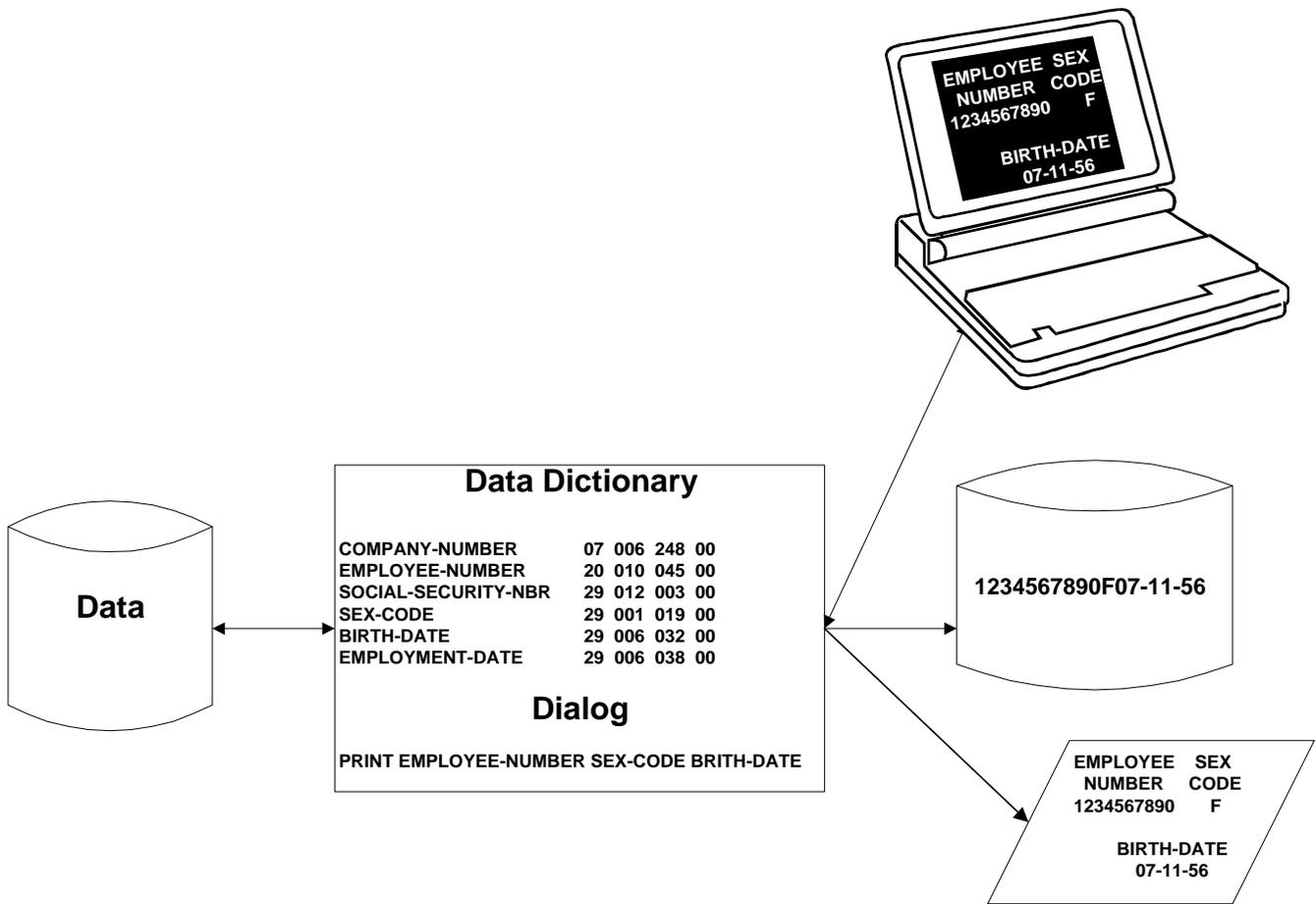
### Payroll segments

In Payroll Administration, the segment key information field is usually a code. For example, the F segment uses a 3-digit code to distinguish between the employee's legal name/address and a mailing name and address.

### HR segments

In HR Administration, the segment keys usually include a date. For example, when a salary change is made, the 'salary effective date' will distinguish it from other salary changes.

# Logical Data Structure



---

## NOTES

---

## Logical data structure

### Logical structure

The Solution Series does not directly manipulate any record within a file. Instead, The Solution Series copies the record to be processed to memory where the manipulation occurs. Once all manipulation is complete the manipulated record in memory then replaces the original record in the file.

The Solution Series accomplishes this task by creating a logical view of the data as it resides in the program's memory.

- The CBSV and CSL programs provide the dialog to determine what records are copied into the memory as well as what manipulation will occur.
- The data dictionary provides the logical view of the data as it resides in the program's memory.
  - The data dictionary is commonly referred to as the field name table.
  - The Solution Series memory is known as working storage.
  - Working storage is accessed/addressed using pointers.

## Working Storage Areas

Area 1	Area 2
Pointer Table Screen (data to workstation) I/O Buffer	Work (data from workstation) Tax Master Record Employee Master Record

Area 3	Area 4
CSL Object Report Extract	Company Master Record

---

### NOTES

---

## Logical data structure, continued

### What is working storage?

Working storage is the memory area of The Solution Series software.

- Working storage resides in the CBSV COBOL programs.
- There are four main areas of working storage called AREA1, AREA2, AREA3 and AREA4, which are accessible to CSL programs.
- Each of the four main areas is subdivided into a number of workspaces. These workspaces are called 'Pointer Spaces', or simply 'Pointers'.

### What are pointers?

Pointers are a way to address data. The Solution Series uses pointers as an indexing technique to address every data element in working storage.

### Working storage and pointers

Each subdivision of working storage, each pointer space, can be viewed as a two-dimensional table. Each entry, or 'occurrence', in a table has a length defined by the pointer table that contains the address of each pointer.

The system manipulates the pointer table to determine the location of a field in working storage by adding together the field's displacement, as specified in the data dictionary, and the value of the pointer's address. The system uses the result as a displacement within the specific AREA in which the field resides.

# Segment/Pointer Relationship

## Record/Pointer/Segment

<u>Record</u>	<u>Ptr</u>	<u>Seg</u>	<u>Description</u>
Company	21	A	Company Name & Address
	22	B	Company Earning & Deductions (HEDs)
	23	C	Other Company detail/user-Defined Data
	24	D	Report Generator Selection
Tax	25	T1	Tax Body Information
	26	4	Tax Exemptions and Credits
	27	5	Tax Brackets
Employee	28	Key	Employee key information
	29	E	Basic employee data
	30	F	Employee name and address
	31	G	Labor detail
	32	H	Earnings and Deductions (HEDs)
	33	H	Earnings and Deductions history
	34	J	Employee taxes
	35	J	Employee tax history
	36	L	Human Resources data/user-defined data
	37	P	Period Table data

## Area 2 - Employee Record

29  
↓

28→	999999M12345678999	EE 308-82-3775 ... .
30→	F001AUSTIN, STEVEN...	F999AUSTIN, STEVEN...
31→	G0105000CHGOMANU...	G020500NYC SALE...
32/33→	H0010100...	H0030100... H5010153...
34/35→	J101 01...	J102 10... J103 10...
36→	LO100204AUSTIN, J...	LO2002ACME MANUF...
	LO3CHICAGO, IL606...	LZF213A319I1500
	LZF214A319S0700...	LZQ213A31SASMITH...
	LZR218G029C1600...	LZR218A319C1500...
37→	P910101...	P910201... P910131... P910401...

---

## NOTES

---

## Logical data structure, continued

### Pointers and segments

Employee database records are divided into segments. Segments are the logical subdivisions of records and are sequentially ordered in a master record.

- Each segment is assigned a unique one-character type code. For example, an employee's name and address reside on the segment with type code F.
- When a record is read into a specific area of working storage each segment is addressed by a pointer entry in the pointer table.

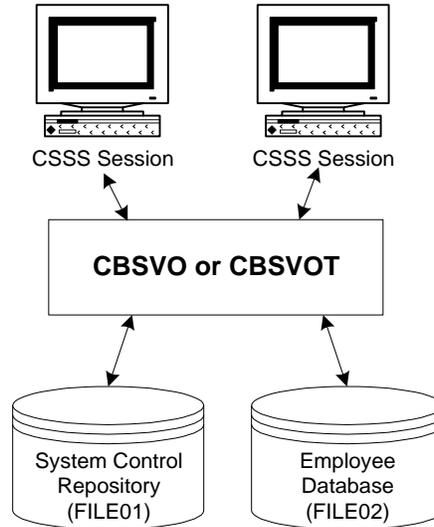
Looking at the preceding diagram, it is clear that the master record for an employee can contain many different types of data divided into several segments and that each segment has one pointer used to address the location of the data within working storage.

Some segments are defined as multiple-occurrence, or stacked, segments. A stacked segment contains the kind of data that can occur more than once, such as the names of an employee's dependents.

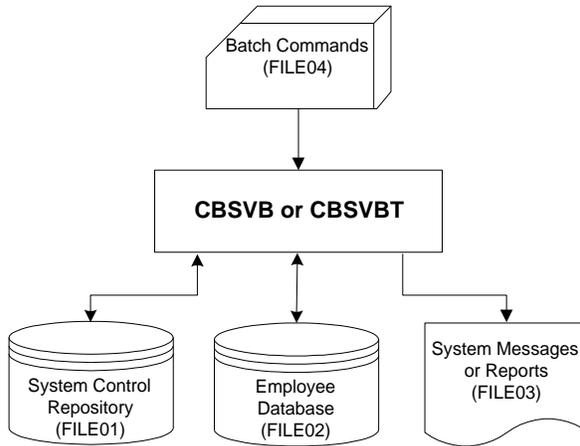
Since only one pointer addresses each segment, it is the job of the programmer and the programming language to manipulate the address of the pointer to point to the proper occurrence of the data. This topic will be discussed throughout this course and subsequent CSL programming courses.

# The Solution Series Processing Modes

## Online



## Batch



---

## NOTES

---

## The Solution Series processing modes

### Processing modes

There are two processing modes that use the System Control Repository (FILE01) and the Employee Database (FILE02). They are:

- Online
- Batch

### Online

Online processing is the procedure of entering, viewing, or manipulating data through an interactive Solution Series session.

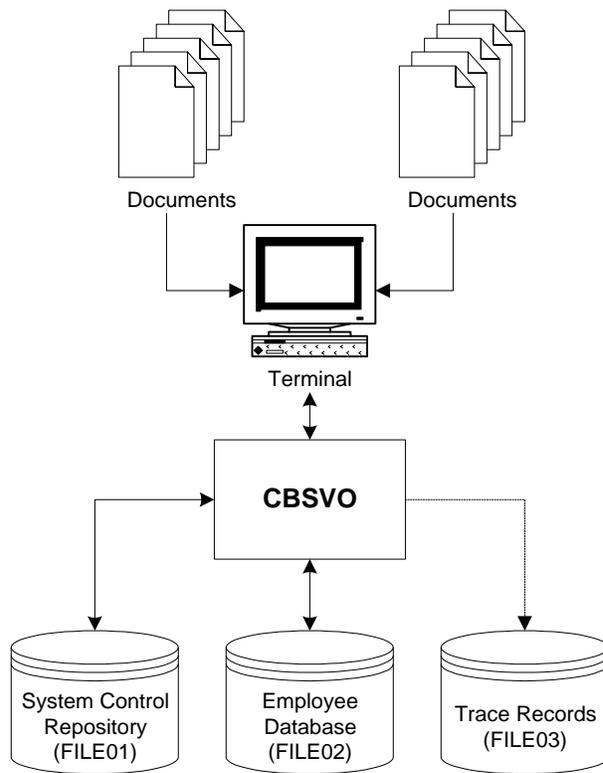
- Data is updated to the Employee Database (FILE02) or the System Control Repository (FILE01) instantly.
- All verification is performed instantly with corresponding messages for further interaction.

### Batch

Batch processing is the procedure of grouping a number or series of tasks to be accomplished in a background mode.

- These groups are then updated to the Employee Database (FILE02) or System Control Repository (FILE01) when a particular job stream is processed.

# Online Requirements



---

## NOTES

---

## The Solution Series processing modes, continued

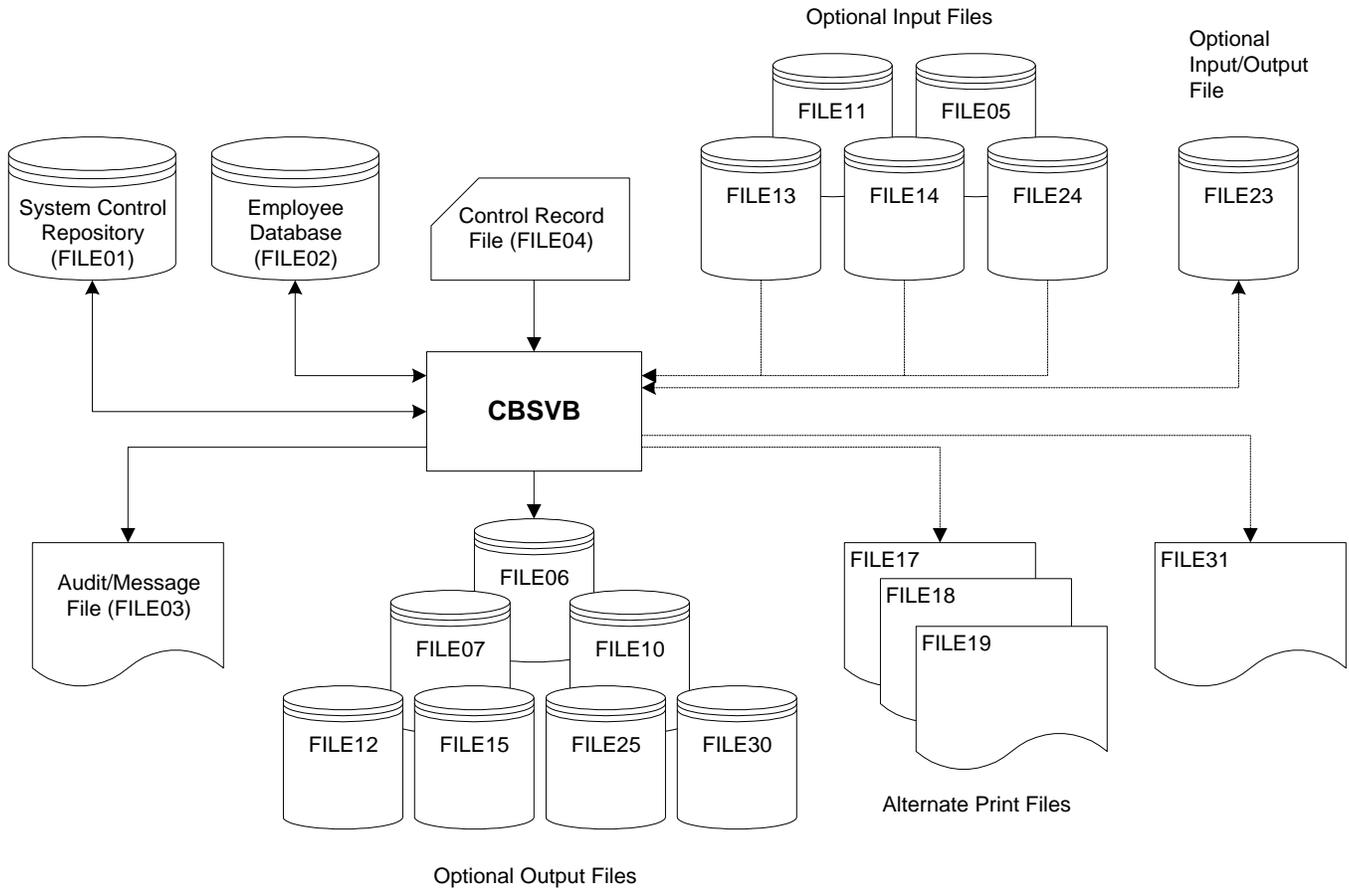
### Online requirements

Processing The Solution Series in an online mode requires two and/or three files to be opened, as well as the execution of the CBSVO program.

Online requirements include:

- Program:
  - CBSVO online production program or,
  - CBSVOT online trace program
- Files:
  - System Control Repository (FILE01)
  - Employee database (FILE02)
  - Audit/Report File (FILE03)—Required only for CBSVOT
- Workstation:
  - Keyboard
  - CRT display

# The Solution Series Processing Modes



---

## NOTES

---

## The Solution Series processing modes, continued

### Batch requirements

To process The Solution Series in a batch mode, the four required files must be defined and the CBSVB program must be executed.

Batch requirements include:

- Programs:
  - CBSVB batch production program or
  - CBSVBT batch trace program
- Required files:
  - System Control Repository (FILE01)
  - Employee database (FILE02)
  - Audit/Report file (FILE03)
  - Control record file (FILE04)
- Optional files:
  - FILE05, FILE06, FILE07, FILE10, FILE11
  - FILE12, FILE13, FILE14, FILE15, FILE17
  - FILE18, FILE19, FILE23, FILE24, FILE25, FILE30, FILE31.

*Note: Not all files are used in each job. Check documentation for which files are used in a particular job. There are additional files used in payroll processing and interfacing with third party processing tools.*

# Batch Control Record Layout

	1	1	2	2	3	3	4	4	5	5	8		
1	5	0	5	0	5	0	5	0	5	0	5	//	0
P	CONTROL	J00010	999999	AUDIT	0001		9999						

- 1 - 15 = Comment Area
- 16 = Action
- 17 - 22 = Control 1-2
- 23 - 28 = Program
- 29 - 30 = Code
- 31 - 40 = Key field
- 41 - 55 = Additional Key field

---

## NOTES

---

## The Solution Series processing modes, continued

### Batch control record

Batch processing in The Solution Series requires a control record in FILE04 that tells the CBSVB program what process to execute.

- The control record directs the processing of CBSVB similar to the way the command line is used to direct an online session using CBSVO.
- Multiple control records can be entered into FILE04. Each command record is executed sequentially based on its content.

### Control record format

The following defines the layout of the batch control format:

<b>Position</b>	<b>Field/Description</b>
1–15	Comments, no entry is required
16	Action field
17–22	Control 1-2 field
23–28	Program field
29–30	Code field: Code-1 and Code-2
31–40	Key field
41–55	Additional Key field

## Section Summary

- **The Solution Series architecture**
- **Program types**
- **Physical data structures**
- **Logical data structure**
- **The Solution Series processing modes**

---

### NOTES

---

## Section summary

In this section, you learned about several components of The Solution Series Architecture. Specifically you learned about:

### The Solution Series architecture

---

---

---

---

### Program types

---

---

---

---

### Physical data structures

---

---

---

---

### Logical data structure

---

---

---

---

### The Solution Series processing modes

---

---

---

---

---

# Section 2 Exercise

---

**NOTES**

---

## Section 2 exercise

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in the section.

### Directions

Answer the questions below.

Match each term with its proper definition.

- |                          |    |   |
|--------------------------|----|---|
| _____ 1. CBSVO           | a. | A unique set of data elements used to identify and locate a record.   |
| _____ 2. CBSVBT          | b. | A way to address data in Working Storage.                             |
| _____ 3. Record Key      | c. | A piece of data within a record or a records segment.                 |
| _____ 4. Segment         | d. | 80-byte sequential output file.                                       |
| _____ 5. Pointer         | e. | A sub-division of a record that contains data of a particular kind.   |
| _____ 6. Field           | f. | Batch COBOL program that includes diagnostics for debugging purposes. |
| _____ 7. Working Storage | g. | The memory area of The Solution Series software.                      |
| _____ 8. Control Record  | h. | Online COBOL program that allows real-time, interactive updating.     |
| _____ 9. FILE10          | i. | The Command Line for batch processing.                                |

---

**NOTES**

---

## Section 3: Cyborg Scripting Language Basics

---

### Table of Contents

Introduction.....	55
CSL programming rules.....	57
CSL syntax conventions .....	61
CSL basic commands.....	63
Creating a CSL program.....	75
Section summary.....	93
Section 3 exercise .....	95

## Objectives

- **Recognize the CSL programming rules**
- **Identify the CSL syntax conventions**
- **Identify basic CSL commands**
- **Create a CSL program using EDIT**

---

## NOTES

---

## Introduction

### Purpose

The purpose of this section is to introduce you to the basics of CSL programming. These basics are the foundation to CSL and will be expanded upon in subsequent sections and courses.

### Objectives

Upon completion of this section you will be able to:

- Recognize the CSL programming rules.
- Identify the CSL syntax conventions.
- Identify the basic CSL commands.
- Create a CSL Program using EDIT.

*Note: The examples used in this course are transferable to every programming type (for example, report, form, and/or process programs). When a technique is specific to one program type, a notation is made designating such usage.*

## General Programming Rules

<b>Program names:</b>	Delivered - yyySCR or yyyRPT - where 'y' is an alphanumeric, but never X in the first position.  User-defined - Xyyyyy or XyyyPT- where 'y' is any alphanumeric.
<b>Field/Verb names:</b>	Delivered - alphanumeric with hyphens separating words, but never beginning with X.  User-defined - Begin with X, then alphanumeric, or separated with an '*' asterisk.
<b>Coding:</b>	Words\Verbs\Fields must be separated by at least one space. All statements must end with a period.

---

### NOTES

---

## CSL programming rules

### Programming rules

CSL has very few rules, but the rules that do exist are very important. To easily identify the rules we will break them into two different categories:

- General format rules
- Special symbol rules

### General format rules

The general format rules provide the guidelines for naming and coding programs. The rules include:

- User defined program names may be from 1 to 6 positions and must begin with X.
- User defined verb and field names may not contain embedded spaces and must begin with an X.
- When coding, all words and commands in a statement must be separated by at least one space.
- All statements must end with a period. This includes paragraph labels, if used.

## Special Symbol Rules

- @ Comment indicator
- ' Literal value indicator
- : Numerical literal value indicator
- . End of a sentence or programming statement indicator

OR

- Decimal point indicator for a numeric literal value
- + Addition indicator
- Subtraction indicator
- \* Multiplication indicator
- / Division indicator
- = Calculation result indicator

---

**NOTES**

---

## CSL programming rules, continued

### Special symbol rules

Several special symbols are used in the CSL programming language. The special symbols are standard keyboard characters that have unique meaning when they are not used as a literal value. The special symbols include:

- @ Comment indicator  
Any text on the same line following an '@' sign is ignored by the compiler and the system programs.
- ' Single quote marks  
Used to delimit literal values in programming statements.
- : Colon  
Used to denote a numeric literal.
- . Period  
Used to denote the end of a sentence or programming statement, or to indicate the number of decimal places in a numeric value used in a calculation.

Additionally, there are special symbols used in calculations to indicate arithmetic operations. These are:

- + Plus sign means add
- - Dash means subtract  
The field/value following the dash is to be subtracted from the field/value preceding the dash.
- \* Asterisk means multiply  
Indicates that the field or value preceding the asterisk is to be multiplied by the field/value following the asterisk.
- / Slash means divide  
The field/value preceding the slash is to be divided by the field/value following the slash.
- = Equal sign  
Used to end a calculation. The result is placed in the field following the equal sign.

# CSL Syntax Conventions

**Format:** VERB operand { *operand* } [ *operand...* ].

## Syntax Rules

1. Capitalized words are CSL reserved words.
2. Underlined words are required elements in the statement or option specified.
3. Lowercase words are programmer-supplied.
4. Braces { } denote that one of the enclosed items is required.
5. Brackets [ ] denote that one of the enclosed items is optional.
6. Punctuation, when included in the format is required.
7. The use of three dots (...) indicates that the preceding item may be repeated if desired.

---

## NOTES

---

## CSL syntax conventions

### Syntax conventions

Before discussing the CSL statements in detail, let us define the format of the syntax that will be used to describe each statement.

All instructions will be represented in an entry entitled 'Format' that is used to indicate the rules for that instruction. The illustration will help you interpret the syntax for each instruction.

# CSL Commands

- **Basic verbs**
- **Macro verbs**

---

NOTES

---

## CSL basic commands

### Description

CSL programs are composed of a series of statements that tell the computer what steps you want it to perform. Each statement, in turn, requires a verb, or action word, that specifies what you want the computer to do and the order it should follow when processing the program statements.

Two kinds of verbs are used which simplify the task of creating forms, reports, and programs. These verbs are:

- Basic verbs
- Macro verbs

### Basic verbs

'Basic verbs' are the verbs that are built into CSL as designed. They are common to many programs in The Solution Series because they perform basic actions related to The Solution Series components.

- The PRINT verb generates output data, such as field headers, prompts, and messages, to a form.
- FIND is a 'basic verb' that instructs the program to look for a particular kind of data.

### Macro verbs

'Macros' are verbs that are small programs in themselves. They carry out a number of steps (series of 'basic verbs') when they are executed.

- FIND-JOB-EFFECTIVE is a 'macro' that looks for the most recent job change for an employee.
- FIND-SALARY is a 'macro' that looks for the most recent salary for an employee. The 'macro' reduces the number of programming statements you have to code.

## Record Access Verbs

Format:                    READ-COMPANY.

Format:                    READ-EMPLOYEE.

Format:                    READ-TAXES.

Example:                  READ-COMPANY.    READ-EMPLOYEE.

---

### NOTES

---

## CSL basic commands, continued

### Record access verbs

READ- verbs are used to access records on the Employee Database and inquire on the data. No updates can be made using these verbs.

- Access verbs read the entire Employee Database record into working storage.
- At least one of the access verbs must be the first command coded in a form. The verb used depends on the type of data being accessed.
- More than one access verb can be used in a program.

There are separate verbs for accessing employee, company and tax records. They are:

- **READ-COMPANY**  
Reads the company record data into working storage. READ-COMPANY uses the data in the CONTROL-1-2 field as the Key to the record.
- **READ-EMPLOYEE**  
Reads the employee record data into working storage. READ-EMPLOYEE uses the data in the CONTROL-1-2 field and KEY field as the Key to the record.
- **READ-TAXES**  
Reads the tax record into working storage. READ-TAXES uses the data in the CONTROL-1-2 field and KEY field as the key to the record.

# Data Display Verbs

## PRINT

### Format

$$PRINT \left\{ \begin{array}{l} \textit{literal} \\ \textit{field - name - 1} \end{array} \right\} \left[ \begin{array}{l} \textit{literal...} \\ \textit{field - name - 2...} \end{array} \right].$$

### Example: Displaying an formatted employee's birth date:

```
READ-EMPLOYEE.  
PRINT 'Birth Date: ' BIRTH-DATE.
```

```
Birth Date: 10-22-1982
```

## OUTPUT

### Format

$$OUTPUT \left\{ \begin{array}{l} \textit{literal} \\ \textit{field - name - 1} \end{array} \right\} \left[ \begin{array}{l} \textit{literal...} \\ \textit{field - name - 2...} \end{array} \right].$$

### Example 1: Displaying a form title and the Company Name.

```
READ-COMPANY.  
OUTPUT 'Company Name and Address:' COMPANY-NAME.
```

```
Company Name and Address: Acme Manufacturing
```

### Example 2: Displaying a form prompt

```
OUTPUT ' ENTER: EE2SCR Optional '  
'Employee Information'.
```

```
ENTER: EE2SCR Optional Employee Information
```

---

## NOTES

---

## CSL basic commands, continued

### Data display verbs

The Solution Series provides two verbs for displaying data to an output area. The output area is pointer 11 (SCREEN) which will later be directed by the program to the appropriate output device (for example, an output file or form).

- Both verbs will display field data and alphanumeric literals, including blanks.
- You can specify more than one operand to appear on the same line in the output.
- The literal must be in single quotes that cannot span more than one line.

### PRINT

The PRINT verb moves field data and literal values to pointer 11 (SCREEN).

- PRINT edits fields according to the edit length and edit routine specified on the field name table.

### OUTPUT

The OUTPUT verb moves field data or literal values to pointer 11 (SCREEN).

- OUTPUT moves data without editing or converting it in any way and displays it as it is stored in the record.
- Be cautious when using the OUTPUT verb since some fields contain unprintable characters that can corrupt a form's display.

# Special Display Verbs

## SPACE-OVER :nn.

### Format

SPACE-OVER :nn.

### Example: Displaying a form title and the Company Name.

```
READ-COMPANY.  
PRINT 'Company Name and Address'.  
SPACE-OVER :24. PRINT COMPANY-NAME.
```

## INQUIRY-NAME.

### Format

INQUIRY-NAME.

### Example: Generating an employee form title.

```
READ-EMPLOYEE.  
PRINT 'Screen Title'. SPACE-OVER :36.  
INQUIRY-NAME.
```

---

## NOTES

---

## CSL basic commands, continued

### Special display verbs

In addition to using the PRINT and OUTPUT verbs to display data, The Solution Series provides verbs to easily display specific types of data. These verbs are:

- SPACE-OVER
- INQUIRY-NAME

### SPACE-OVER :nn

The SPACE-OVER :nn verb is used to move spaces to pointer 11 (SCREEN) over a specified number of positions on the form line.

- The number of spaces must be indicated by a colon and a two-digit number.
- The two-digit numeric literal value can be from 01 to 60.

### INQUIRY-NAME

The INQUIRY-NAME verb is used to output the employee-name to pointer 11 (SCREEN) for the EMPLOYEE-NUMBER entered in the key field when the form is accessed.

- The name is output in Inquiry mode only in Last name, First name format.

*Note: This applies to The Solution Series/ST version 3.2 only.*

# Form Line Control

## Format

NEXT-LINE.

## Example: Displaying multiple lines of information.

```
READ-COMPANY. PRINT 'Company Name and Zip Code Information'.  
NEXT-LINE. NEXT-LINE.  
PRINT 'Company Name: ' COMPANY-NAME. NEXT-LINE.  
PRINT 'Zip Code: ' COMPANY-ZIP-CODE. NEXT-LINE.
```

---

## NOTES

---

## CSL basic commands, continued

### form line control

To control the placement of information on a form, The Solution Series provides a verb to move to the 'next line' of the form.

### NEXT-LINE

The NEXT-LINE verb marks the end of a formatted form line and writes it to pointer 11 (SCREEN). It also acts as a carriage control indicator.

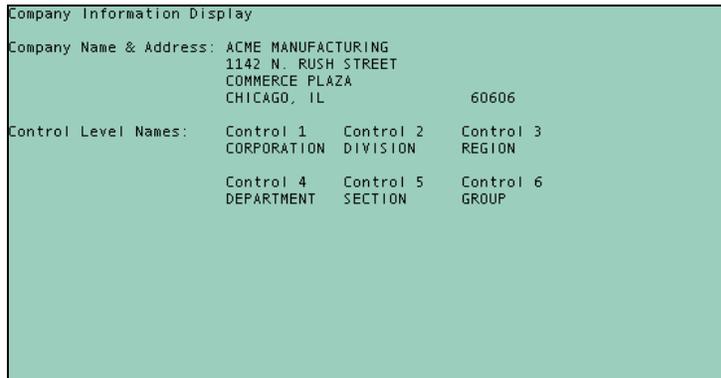
- End every form line with at least one NEXT-LINE verb.
- NEXT-LINE moves the screen pointer to the beginning of the next form line.
- You can use NEXT-LINE to format a blank display line, or create double-spacing between lines by repeating the NEXT-LINE verb.

*Note:* The NEXT-LINE verb is specific to online form programming. Equivalent carriage control techniques for report programs are discussed in the Cyborg Scripting Language Report Customization course.

*Note:* If NEXT-LINE is executed on the last line, the program will end.

# CSL Program

```
SECURITY ' '. @ Company Information Display DIPP
@LAST MODIFIED ON: 02-15-99 BY: USER AUTHOR: USER
READ-COMPANY.
PRINT 'Company Information Display'.
NEXT-LINE. NEXT-LINE.
PRINT 'Company Name & Address: ' COMPANY-NAME. NEXT-LINE.
SPACE-OVER :24. PRINT COMPANY-ADDRESS. NEXT-LINE.
SPACE-OVER :24. PRINT COMPANY-ADDRESS-2. NEXT-LINE.
SPACE-OVER :24. PRINT COMPANY-CITY/STATE ' '
COMPANY-ZIP-CODE. NEXT-LINE. NEXT-LINE.
PRINT 'Control Level Names: Control 1 Control 2'
' Control 3'. NEXT-LINE. SPACE-OVER :24.
PRINT CONTROL-1-NAME ' ' CONTROL-2-NAME ' ' CONTROL-3-NAME.
NEXT-LINE. NEXT-LINE. SPACE-OVER :24.
PRINT 'Control 4 Control 5 Control 6'. NEXT-LINE.
SPACE-OVER :24.
PRINT CONTROL-4-NAME ' ' CONTROL-5-NAME ' ' CONTROL-6-NAME.
```



```
Company Information Display
Company Name & Address: ACME MANUFACTURING
                        1142 N. RUSH STREET
                        COMMERCE PLAZA
                        CHICAGO, IL                60606
Control Level Names: Control 1 Control 2 Control 3
                     CORPORATION DIVISION REGION
                     Control 4 Control 5 Control 6
                     DEPARTMENT SECTION GROUP
```

---

## NOTES

---

## CSL basic commands, continued

### Program example

The program shown on the opposite page provides an example of some of the verbs discussed in this section.

The structure that is followed includes:

- Access a record using a READ- verb.
- Print a title for the display using the PRINT verb.
- Move down the display form using the NEXT-LINE verb.
- Print the detail information for each field.
- Position fields within a line using the SPACE-OVER:nn verb and PRINT with blank literal values.

## Creating a CSL Program

- **System Control Repository editor (EDIT)**
- **CSL compiler (RELOAD)**

---

### NOTES

---

## Creating a CSL program

### CSL programming utilities

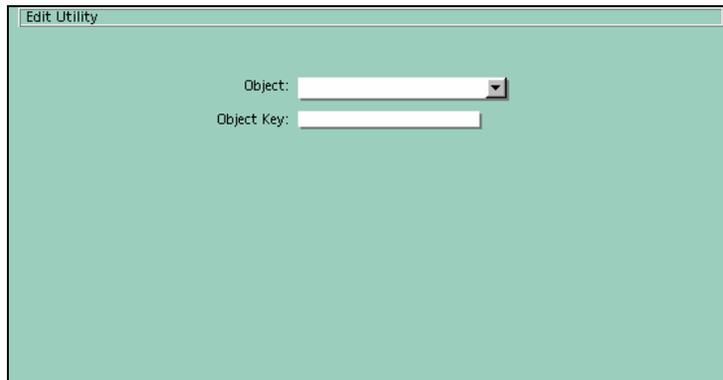
The Solution Series provides several powerful utilities to support the development of CSL programs.

These utilities include, but are not limited to:

- System Control Repository editor (EDIT) that is used to edit CSL programs.
- CSL compiler (RELOAD), the ‘compiler’, used to translate source code into code that The Solution Series can execute.

*Note: The following discussion covers the basics of creating and compiling a CSL program. Other utilities and additional features of EDIT are presented in the next section—The Solution Series Utilities.*

# System Control Repository Editor (EDIT)



---

## NOTES

---

## Creating a CSL program, continued

### System Control Repository edit

The System Control Repository editor (EDIT) program is an editor that allows you to make changes to your System Control Repository (FILE01) records.

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		System Control Repository Utilities
<b>Task:</b>		Edit Control Repository Object

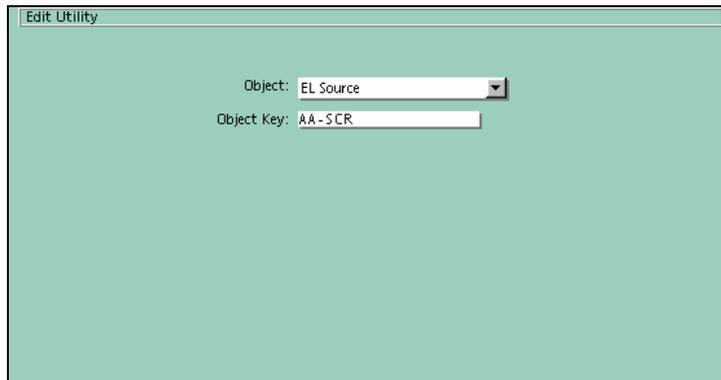
**Result:** The Edit Utility form (EDIT) is displayed.

### Edit prompt fields

Two fields display on the Edit Utility form (EDIT). They are Object and Object Key.

- **Object**  
Each System Control Repository record type is assigned an object code. A single record type may have several object codes assigned to allow specific display.
- **Object Key**  
The Object Key field allows you to specify the System Control Repository record group you want to edit. The value of the Object Key field is dependent on the Object field.

# Editing CSL Source



```
COMMAND: 
Seq SC English Language Source
00000 SECURITY 'PP'. @ Company Name And Address DEPP
00001 @LAST MODIFIED ON: 04-02-98 BY: KEYD AUTHOR: RMF
00003 @The AA-SCR screen is used to make additions or changes to
00004 @the legal company name, company address, or Control Level
00005 @titles.
00100 P100-START-SCREEN.
00200 UPDATE-COMPANY. NEW-SCREEN-STYLE.
00300 IF GUI-IN-USE SET-PGDN-GRAY
00400 SET-PGUP-GRAY DISALLOW-DELETE.
00800 SCREEN-SECTION '0'.
00900 P200-ENTRY-SCREEN.
01000 SCREEN-SECTION '1'.
01100 IF CONTROL-COUNTRY EQUAL '1'
01200 GO TO P210-SECTION-3.
01300 SCREEN-SECTION '2'.
01400 GO TO P220-SECTION-4.
01500 P210-SECTION-3.
01600 SCREEN-SECTION '3'.
01700 P220-SECTION-4.
```

---

## NOTES

---

## Creating a CSL program, continued

### EDIT, continued

To EDIT a CSL program you must specify the Object and Object Code parameters:

1. Select 'EL Source' to indicate CSL Source code.
2. Type the name of the program to be edited, for example, AA-SCR.
3. Click the Save this form button on the toolbar or press Enter.

**Result:** Your form displays your requested object or displays an error message if the object is not found.

# Editing CSL Source

```
COMMAND: |
Seq SC English Language Source
00000 SECURITY 'PP'. @ Company Name And Address DEPP
00001 @LAST MODIFIED ON: 04-02-98 BY: KEVD AUTHOR: RMF
00003 @The AA-SCR screen is used to make additions or changes to
00004 @the legal company name, company address, or Control Level
00005 @titles
00100 P100-START-SCREEN.
00200 UPDATE-COMPANY. NEW-SCREEN-STYLE.
00300 IF GUI-IN-USE SET-PGDN-GRAY
00400 SET-PGUP-GRAY DISALLOW-DELETE.
00800 SCREEN-SECTION '0'.
00900 P200-ENTRY-SCREEN.
01000 SCREEN-SECTION '1'.
01100 IF CONTROL-COUNTRY EQUAL '1'
01200 GO TO P210-SECTION-3.
01300 SCREEN-SECTION '2'.
01400 GO TO P220-SECTION-4.
01500 P210-SECTION-3.
01600 SCREEN-SECTION '3'.
01700 P220-SECTION-4.
```

---

## NOTES

---

## Creating a CSL program, continued

### Edit Utility (EDIT)

The Edit Utility (EDIT) has two main areas:

- The command line is used to perform specific actions on the object within the EDIT's text body.
- The text body contains each line of text to be modified.

### EDIT command line

The Edit Utility's (EDIT) command line is comprised of two fields.

- The COMMAND field is used to indicate what action you want to perform within the EDIT program. Type a '?' and press Enter to list the available command codes.
- The parameter field allows you to provide options to the command field.
- Some codes are valid for any type of object, while others are limited to specific objects.

*Note:* Detailed information on each command and required parameters are available in the *Quick Solution—Technical Reference Guide* in the appendices of this documentation or the online documentation for *Command Codes and Parameters*.

# Editing CSL Source

```
COMMAND: |
Seq SC English Language Source
00000 SECURITY 'PP'. @ Company Name And Address DEPP
00001 @LAST MODIFIED ON: 04-02-98 BY: KEVD AUTHOR: RMF
00003 @The AA-SCR screen is used to make additions or changes to
00004 @the legal company name, company address, or Control Level
00005 @titles
00100 P100-START-SCREEN.
00200 UPDATE-COMPANY. NEW-SCREEN-STYLE.
00300 IF GUI-IN-USE SET-PGDN-GRAY
00400 SET-PGUP-GRAY DISALLOW-DELETE.
00800 SCREEN-SECTION '0'.
00900 P200-ENTRY-SCREEN.
01000 SCREEN-SECTION '1'.
01100 IF CONTROL-COUNTRY EQUAL '1'
01200 GO TO P210-SECTION-3.
01300 SCREEN-SECTION '2'.
01400 GO TO P220-SECTION-4.
01500 P210-SECTION-3.
01600 SCREEN-SECTION '3'.
01700 P220-SECTION-4.
```

---

## NOTES

---

## Creating a CSL program, continued

### Edit text body

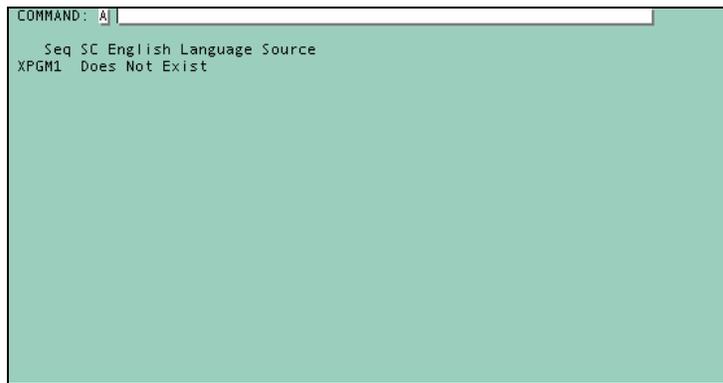
The Edit Utility's text body is comprised of many lines which contain different column formats based on the object you are editing.

### Text body columns

The column names and format for an Object of EL Source are as follows:

- The line command is the first column of the text body area. This column is unlabeled and indicates how the current line will be modified. The valid line commands are:
  - A Add
  - C Change
  - D Delete
  - \*Cancel all following Line Commands
- The Seq (Sequence Number) column is used to distinguish one line from another.
- The SC (Sequence Code) column is used to display an error indicator when a syntax error is encountered.
- The English Language Source column allows you to enter up to 60 characters of information per sequence line.

# Adding a New Program



---

## NOTES

---

## Creating a CSL program, continued

### Add a new program

The procedure to add a new program is similar to the steps that we walked through on the previous pages. However, since the program does not already exist in the control file there is one additional step to complete:

Make the following selections from the Navigator:

<b>Component:</b>		Development Tools
<b>Process:</b>		System Control Repository Utilities
<b>Task:</b>		Edit Control Repository Object

**Result:** The Edit Utility form (EDIT) is displayed.

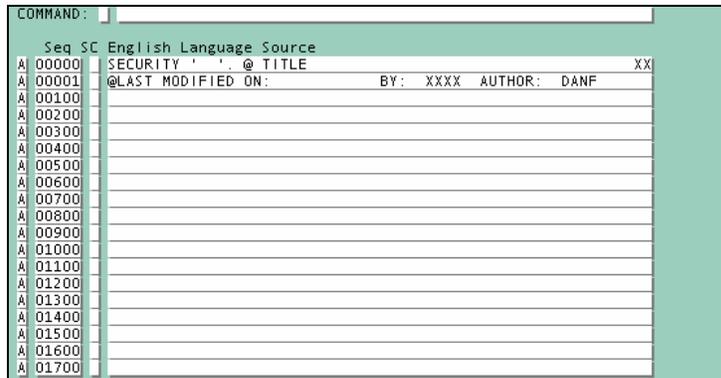
1. Select EL Source to indicate CSL Source code.
2. Type the name of the program to be edited, for example, XPGM1.
3. Click the Save this form button or press Enter.

**Result:** An error message is displayed when the program is not found.

4. Type A to indicate Auto Add mode.
5. Press Enter.

**Result:** The Edit Utility form (EDIT) displays and is ready for entry.

# Adding a New Program Reserved Sequence Numbers



COMMAND:

Seq	SC	English Language Source	XX
A	00000	SECURITY ' ' @ TITLE	
A	00001	@LAST MODIFIED ON: BY: XXXX AUTHOR: DANF	
A	00100		
A	00200		
A	00300		
A	00400		
A	00500		
A	00600		
A	00700		
A	00800		
A	00900		
A	01000		
A	01100		
A	01200		
A	01300		
A	01400		
A	01500		
A	01600		
A	01700		

---

## NOTES

---

## Creating a CSL program, continued

### Prescribed sequence lines

The first eleven (11) lines of source code have prescribed purposes and layouts.

- Sequence line 00000—Security and Title line. This is a required line in CSL source code and is created automatically by the system. It contains the SECURITY verb, program title, and Help form codes.
- Sequence line 00001—Program Modification line. This is a required line in CSL source code and is created and updated automatically by the system.
- Sequence line 00002—Placement Key line for forms and the Parameter Entry Program line for Batch Reports. This line is optional depending on the type of command line key information required to access the form or run-time parameters necessary to execute the batch report.
- Sequence lines 00003 through 00006—Optional and may contain brief documentation on the program.
- Sequence lines 00007 through 00010—Optional and may contain other pieces of Help form key information. These lines are used in conjunction with sequence line 00002 and special verbs.

# Adding a New Program

```

COMMAND:
Seq SC English Language Source
A 00000 SECURITY ' ' @ TITLE XX
A 00001 @LAST MODIFIED ON: BY: XXXX AUTHOR: DANF
A 00100 READ-COMPANY. PRINT 'Company Information Display'.
A 00200 NEXT-LINE. NEXT-LINE.
A 00300 PRINT 'Company Name & Address: ' COMPANY-NAME. NEXT-LINE.
A 00400 SPACE-OVER :24. PRINT COMPANY-ADDRESS. NEXT-LINE.
A 00500 SPACE-OVER :24. PRINT COMPANY-ADDRESS-2. NEXT-LINE.
A 00600 SPACE-OVER :24. PRINT COMPANY-CITY/STATE ' '
A 00700 COMPANY-ZIP-CODE. NEXT-LINE. NEXT-LINE.
A 00800 PRINT 'Control Level Names: Control 1 Control 2'
A 00900 ' Control 3'. NEXT-LINE.
A 01000 SPACE-OVER :24.
A 01100 PRINT CONTROL-1-NAME ' ' CONTROL-2-NAME ' ' CONTROL-3-NAME.
A 01200 NEXT-LINE. NEXT-LINE. SPACE-OVER :24.
A 01300 PRINT 'Control 4 Control 5 Control 6'. NEXT-LINE.
A 01400 SPACE-OVER :24. PRINT CONTROL-4-NAME ' ' CONTROL-5-NAME
A 01500 ' ' CONTROL-6-NAME.
A 01600
A 01700
    
```

```

COMMAND:
Seq SC English Language Source
00000 SECURITY ' ' @ TITLE XX DANF
00001 @LAST MODIFIED ON: 03-19-99 BY: DANF AUTHOR: DANF DANF
00100 READ-COMPANY. PRINT 'Company Information Display'. DANF
00200 NEXT-LINE. NEXT-LINE. DANF
00300 PRINT 'Company Name & Address: ' COMPANY-NAME. NEXT-LINE. DANF
00400 SPACE-OVER :24. PRINT COMPANY-ADDRESS. NEXT-LINE. DANF
00500 SPACE-OVER :24. PRINT COMPANY-ADDRESS-2. NEXT-LINE. DANF
00600 SPACE-OVER :24. PRINT COMPANY-CITY/STATE ' ' DANF
00700 COMPANY-ZIP-CODE. NEXT-LINE. NEXT-LINE. DANF
00800 PRINT 'Control Level Names: Control 1 Control 2' DANF
00900 ' Control 3'. NEXT-LINE. DANF
01000 SPACE-OVER :24. DANF
01100 PRINT CONTROL-1-NAME ' ' CONTROL-2-NAME ' ' CONTROL-3-NAME. DANF
01200 NEXT-LINE. NEXT-LINE. SPACE-OVER :24. DANF
01300 PRINT 'Control 4 Control 5 Control 6'. NEXT-LINE. DANF
01400 SPACE-OVER :24. PRINT CONTROL-4-NAME ' ' CONTROL-5-NAME DANF
01500 ' ' CONTROL-6-NAME. DANF
01600 DANF
01700 DANF
    
```

## NOTES

## Creating a CSL program, continued

### Editing sequence lines

As you will recall, the line command indicates how the current line will be modified. The functions include adding (A), changing (C), and deleting (D) source code from the program.

- A  
The A line command indicates that a new sequence number's source code is to be added.
- C  
The C line command indicates that an existing sequence number's source code is to be changed.
- D  
The D line command indicates that an existing sequence number's source code is to be deleted.
- \*  
The \* (asterisk) line command indicates that all line commands that follow will be canceled.

### Auto add mode

The A command automatically adds sequence line numbers to the end of your program in increments of 100. It places an A (Add) in the first column of added lines making them ready to accept code entries.

### End auto add mode

The E command ends the Auto Add Mode and returns the form to manual operation.

### Sequence line errors

Sequence line errors occur when you attempt to add a line using an existing sequence number or change/delete a line that does not exist.

- Lines in error appear at the top of the edit form.
- To correct a sequence line error, remove or enter the correct line command.

# Compiling CSL (RELOAD)

```
COMMAND: R
Seq SC English Language Source
00000 SECURITY ' ' @ TITLE XX DANF
00001 @LAST MODIFIED ON: 03-19-99 BY: DANF AUTHOR: DANF DANF
00100 READ-COMPANY. PRINT 'Company Information Display'. DANF
00200 NEXT-LINE. NEXT-LINE. DANF
00300 PRINT 'Company Name & Address: ' COMPANY-NAME. NEXT-LINE. DANF
00400 SPACE-OVER :24. PRINT COMPANY-ADDRESS. NEXT-LINE. DANF
00500 SPACE-OVER :24. PRINT COMPANY-ADDRESS-2. NEXT-LINE. DANF
00600 SPACE-OVER :24. PRINT COMPANY-CITY/STATE ' ' DANF
00700 COMPANY-ZIP-CODE. NEXT-LINE. NEXT-LINE. DANF
00800 PRINT 'Control Level Names: Control 1 Control 2' DANF
00900 ' Control 3'. NEXT-LINE. DANF
01000 SPACE-OVER :24. DANF
01100 PRINT CONTROL-1-NAME ' ' CONTROL-2-NAME ' ' CONTROL-3-NAME. DANF
01200 NEXT-LINE. NEXT-LINE. SPACE-OVER :24. DANF
01300 PRINT 'Control 4 Control 5 Control 6'. NEXT-LINE. DANF
01400 SPACE-OVER :24. PRINT CONTROL-4-NAME ' ' CONTROL-5-NAME DANF
01500 ' ' CONTROL-6-NAME. DANF
01600 DANF
01700 DANF
```

```
XPGM1 99999E RELOAD IS OK. Ver-4.0 LENGTH 327 03-19-99 11:44:16
```

---

## NOTES

---

## Creating a CSL program, continued

### Compiling CSL source

CSL source code must be translated or compiled into an executable format. The Solution Series utility to accomplish this task is known as RELOAD.

RELOAD performs the various tasks:

- Checks for syntax errors and generates Error (E) records for any errors found.
- Checks to make sure the fields are defined on the Field Name Table.
- Creates new object code and deletes any old object (on FILE01 and FILE02) when a successful compile is achieved.
- Creates a RELOAD recap record (an E record) telling the status of the RELOAD, the version, the object code length and the date/time.

### Using RELOAD

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Programming Utilities
- Task:**  Compile a Source Program

**Result:** The source code is RELOADed and the Display Utility form (DISPLY) displays the RELOAD status.

## Section Summary

- **CSL programming rules**
- **CSL syntax conventions**
- **CSL basic commands**
- **Creating a CSL program**

---

NOTES

---

## Section summary

In this section, you learned the basics of CSL programming. Specifically you learned:

### CSL programming rules

---

---

---

---

---

### CSL syntax conventions

---

---

---

---

---

### CSL basic commands

---

---

---

---

---

### Creating a CSL program

---

---

---

---

---

# Section 3 Exercise

---

**NOTES**

---

## Section 3 exercise

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in the section. In order to complete this exercise, build a form that will display the following employee information:

- Form Title—EMPLOYEE INFORMATION
- Form Fields:
  - SOCIAL-SECURITY-NBR
  - WORKERS-COMP-CODE
  - BIRTH-DATE
  - EMPLOYMENT-DATE
  - DATE-OF-TERMINATION

Make sure to display the employee name at the top of the form and to the right of the title. Also, make sure to display the name of the field and each field's content.

---

**NOTES**

---

## Section 4: The Solution Series Utilities

---

### Table of Contents

Introduction.....	99
Additional edit commands .....	101
Copying System Control Repository data .....	119
Displaying System Control Repository data.....	121
Purging System Control Repository data.....	123
Exporting/Importing System Control Repository data .....	125
Program/Field reports .....	131
Section summary.....	149
Section 4 exercise .....	151

## Objectives

- **Use the additional edit commands**
- **Copy, purge, import, and export System Control Repository data**
- **Generate field/program reports**

---

### NOTES

---

## Introduction

### **Purpose**

This section is devoted to The Solution Series utilities that will allow you to manipulate and maintain your CSL programs.

### **Objectives**

Upon completion of this section you will be able to:

- Use the Additional Edit Commands
- Copy, Purge, Import, and Export System Control Repository Data
- Generate Field/Program Reports

## **Additional Edit Commands**

- **Navigation**
- **Manipulation**
- **Miscellaneous**

---

**NOTES**

---

## Additional edit commands

### Edit commands

As you will recall, the Edit Utility program (EDIT) is used to maintain CSL programs. In the previous section you used the Auto Add, End Auto Add, and the Reload commands to manipulate a program. In addition to these commands, the edit utility provides a variety of commands to:

### Edit navigation commands

- Back up
- Next Screen
- Go to a Sequence Number
- Key Change
- Paragraph Locate
- Object change

### Edit manipulation commands

- Change
- Hold
- Find
- Insert
- Delete
- Transfer
- Log and Undo
- Resequence

### Edit miscellaneous commands

- Execute a program

# EDIT Navigation Commands

## Next Screen

COMMAND:	N		
----------	---	--	--

## Backup

COMMAND:	B		
----------	---	--	--

## Go To

COMMAND:	G	00100	
----------	---	-------	--

## Paragraph Locate

COMMAND:	P	100	
----------	---	-----	--

---

## NOTES

---

## **Additional edit commands, continued**

### **Edit navigation**

There are several ways to maneuver within the editor. The following commands detail these options:

### **Next screen command**

The N command is used to advance to the next page of text or source. Typically, your display scrolls automatically when you press the Enter key. You may use the N command to force the display to immediately advance to the next page when you make modifications and press the Enter key.

### **Backup command**

The B command is used to display the previous page of your object.

### **Go to command**

The G command can select a line to appear as the first line on the form. The sequence key must be entered in the first position of the command parameter area.

### **Paragraph locate command**

The P command is used to locate and display a specified paragraph in your program. The paragraph label must begin in the first position of the code line.

# EDIT Navigation Commands

## Key Change

COMMAND:	K	XPGM2
----------	---	-------

## Object Change

COMMAND:	O	C/VHR00
----------	---	---------

Object ←      → Object Key

---

## NOTES

---

## Additional edit commands, continued

### Edit navigation

In addition to navigating within an object, EDIT provides the capability to navigate between different object keys and objects.

### Key change command

The K command is used to quickly change the object key you are editing to a different object key within the same record type (Object code).

- If you are editing EL Source (P/) and wish to access another EL Source (P/) object, you should use this quick method.

### Object change command

The O command is used to access a record type that has an object code that is different from the record type you are editing.

- If you are editing a Option List Value (C/V) and wish to change to EL Source (P/) object, you must use this method.

# EDIT Manipulation Commands

## Logging

COMMAND:	L		*000
----------	---	--	------

## Undo One

COMMAND:	U		*004
----------	---	--	------

## Undo All

COMMAND:	U	ALL	*003
----------	---	-----	------

---

## NOTES

---

**Additional edit commands, continued****Edit manipulation commands**

In addition to the line commands of add, change, and delete, edit provides the capability to perform these functions on more than one line at a time. Since these commands have a global impact on the object you are editing, it is recommended that you use the log and undo feature of EDIT to insure that you do not inadvertently change too much.

**Logging command**

The L command is used to activate Logging. Logging saves the original lines of code as they were prior to any editing and places them in the System Control Repository. They are saved under your Operator ID and are 'remembered' for as long as Logging stays on.

**Undo command**

The U command is used with the L command (Logging) to reverse or 'undo' the previously performed edit. For example, if you delete lines of code by mistake, UNDO will restore them.

*Note:* *Undo assumes that the L command (Logging) is active and the counter to the right of the edit command line contains a number.*

*CAUTION:* *If you 'undo' all prior editing, logging automatically turns itself off.*

# EDIT Manipulation Commands

## Find String - One

COMMAND:	F	/COMPANY-NAME/1
----------	---	-----------------

## Find String - Wildcard

COMMAND:	F	/CONTROL - = - CODE/
----------	---	----------------------

## Change String - One

COMMAND:	C	/READ-EMPLOYEE//READ-COMPANY/1
----------	---	--------------------------------

## Change String - All

COMMAND:	C	/READ-EMPLOYEE//READ-COMPANY/
----------	---	-------------------------------

---

## NOTES

---

## Additional edit commands, continued

### Find string command

The F command works like a global search. It will display all lines containing the specified word or string together or one line at a time.

- You must use literals, that is, the exact characters as they appear in the string. You can also use an equal sign (=) as a wild card in all but the first position of the search argument.
- The string must be WITHIN delimiters. We suggest you use single quotes (') or slashes (/) as delimiters. Do not use a character as a delimiter if it is used in the string.
- To display one line at a time, type the number 1 immediately after the end delimiter.

### Change string command

The C command works like a global change. You can request that the system display all lines containing the specified word or string together, or one line at a time.

- You must use literals, that is, the exact characters as they appear in the string.
- The string must be WITHIN delimiters.
- To display one line at a time, type the number 1 immediately after the end delimiter.
- If the system determines the requested change is not possible, it places an 'X' in the first column of that line. You must change that line manually.

# EDIT Manipulation Commands

## Delete Line - Non-Sequential Numbers

COMMAND:	D	00200, 00240, 00300
----------	---	---------------------

## Delete Line - Range

COMMAND:	D	00200/00300
----------	---	-------------

## Delete Line - Non-Sequential Nbrs & Range

COMMAND:	D	00200, 00240/00900
----------	---	--------------------

---

## NOTES

---

## Additional edit commands, continued

### Delete line command

The D command allows you to delete lines. You can delete individual lines or ranges of lines.

- Enter a range of sequence line numbers separated by a slash.
- Enter non-sequential line numbers separated by commas.
- Designated lines appear at the top of the form with D in the first column.
- If the lines should not be deleted, cursor down, blank out the Ds, and press Enter again.
- If you are sure you want to delete them, press Enter again. The designated lines are deleted.

# EDIT Manipulation Commands

## Hold a Range

COMMAND:	H	00100/00500
----------	---	-------------

## Insert a Held Range using the default increment

COMMAND:	I	01500
----------	---	-------

## Insert a Held Range using a specified increment

COMMAND:	I	01510/005
----------	---	-----------

---

## NOTES

---

**Additional edit commands, continued****Hold range command**

The H command allows you to specify line numbers that you will later copy to another place. The line numbers you specify are placed in HOLD and retained by the system. You may subsequently INSERT (I) the lines elsewhere in the same object or TRANSFER (T) them to another object key.

- Type one sequence line number or a range of sequence numbers separated by a slash (from/to).

**Insert held range command**

The I command allows you to copy text or code within the same object.

- There must be room between the existing lines to insert the copied lines. The inserted line numbers must not duplicate existing line numbers. The inserted lines are renumbered in increments of 10 unless otherwise specified.
- You must use the HOLD command before INSERT.
- Type the beginning sequence line number (non-current number) at which you want to copy the lines (in HOLD), in the Parameter field.
- To specify an increment other than 10 (default), type a slash after the beginning sequence number and then a 3-digit increment, for example, 5 = 005, 20 = 020.

# EDIT Manipulation Commands

Hold a Range

COMMAND:	H	00100/00500
----------	---	-------------

Transfer (Move) the Held Range

COMMAND:	T	01210/005
----------	---	-----------

Hold a Range

COMMAND:	H	00100/00500
----------	---	-------------

Change the Object Key

COMMAND:	K	XPGM2
----------	---	-------

Transfer (Copy) the Held Range

COMMAND:	T	01210/005
----------	---	-----------

---

## NOTES

---

**Additional edit commands, continued****Transfer held range command**

The T command allows you to move text or code within the current object key or copy text of code to another object key.

- There must be room between the existing lines to transfer the lines.
- You must use the hold command before a transfer.
- Type the beginning sequence line number (non-current number) at which you want to copy the lines (in Hold), in the parameter field.
- To specify an increment other than 10 (default), type a slash after the beginning sequence number and then a 3-digit increment.

To MOVE code within the current object key complete the following sequence of commands:

- Execute the H (Hold) command, specifying the lines to be copied.
- Enter a T in the command field
- Type the beginning sequence line number (non-current number) at which you want to copy the lines (in Hold), in the parameter field.

To COPY code to a new object key complete the following sequence of commands:

- Execute the H (Hold) command, specifying the lines to be copied.
- Execute the K (Key Change) command to access the program to which you wish to copy the lines.
- Enter a T in the command field, and the beginning sequence line number at which you want to copy the lines (in Hold) in the parameter field.

# EDIT Miscellaneous Commands

## Resequence Source Code

COMMAND: S

## Execute a Program

COMMAND: X EF-SCR 1234

Position 1-6 =  
Program name

Positions 7-8 =  
Code field

Positions 9-18 =  
Key field

Positions 19-33 =  
Additional Key field

## Sign-off The Solution Series

COMMAND: Q

*Note: The Q command is accurate only for The Solution Series version 3.2.*

---

## NOTES

---

**Additional edit commands, continued****Resequence source code**

The S command is used to execute the P-RSEQ program that resequences the program source code in increments of 100. Do not use on delivered programs.

**Execute a program command**

The X command is used to execute a program directly from the Edit command line instead of using The Solution Series menus. It is simply a short cut.

The parameter field entries must correspond to The Solution Series command line entry fields.

- Positions 1–6—correspond to the Program field.
- Positions 7–8—correspond to the Code field.
- Positions 9–18—correspond to the Key field.
- Positions 19–33—correspond to the Additional Key field.

# COPY

## System Control Repository Record Copying Utility

The screenshot shows a window titled "Copy Utility" with a light green background. It contains three input fields:

- Object: EL Source (dropdown menu)
- Object Key: XPGM1 (text input)
- Object New Key: XPGM2 (text input)

Below the input fields, the text "----Complete----" is displayed.

---

### NOTES

---

## Copying System Control Repository data

### The Copy Utility (COPY)

The Copy Utility (COPY) is used to duplicate a specified System Control Repository (FILE01) type from its existing file name to a new name.

To duplicate a specific System Control Repository (FILE01) type, perform the following steps:

<b>Component:</b>		Development Tools
<b>Process:</b>		System Control Repository Utilities
<b>Task:</b>		Copy

**Result:** The Copy Utility form (COPY) is displayed.

1. Select the Object you want to copy.
2. Type the name of the Object Key you want to copy.
3. Type the name of the New Object Key record where you want the copy placed.
4. Click the Save this form button or press Enter.

**Result:** When the operation is complete, a complete message appears.

# DISPLY

## System Control Repository Record Display Utility

Display Utility

Object:

Object Key:

1st Line Only

```
P EF-SCR 00000 SECURITY 'PP'. @ Employee Information MEPP
P EF-SCR 00001 @LAST MODIFIED ON: 05-28-98 BY: KIMK AUTHOR: WESH 3267
P EF-SCR 00003 @The EF-SCR screen is used in two ways:
P EF-SCR 00004 @o to add an applicant or new employee to the Master File
P EF-SCR 00005 @ (using an LMODEL)
P EF-SCR 00006 @o to change existing employee data
P EF-SCR 00100 P100-START-SCREEN.
P EF-SCR 00105 NO-APPLICANTS. 3259
P EF-SCR 00110 IF W7-01-442 EQUAL 'C' OR 'D' OR 'R' OR 'S' JOES
P EF-SCR 00120 PERFORM P450-SECTION-0 JOES
P EF-SCR 00130 LINK 'ROOSCR'. JOES
P EF-SCR 00140 IF W7-01-442 EQUAL 'X' OR '1' JOES
P EF-SCR 00150 GO TO P350-LINK-ROORTN. JOES
P EF-SCR 00160 IF W7-01-442 EQUAL 'U' JOES
P EF-SCR 00170 MOVE 'F' TO SCREEN-ERROR. JOES
P EF-SCR 00200 KEY-REQUIRED. READ-COMPANY. NO-ZDELETE-ALLOWED.
P EF-SCR 00300 IF GUI-IN-USE DISALLOW-DELETE SET-PGDN-GRAY SET-PGUP-GRAY.
P EF-SCR 00400 NO-@-ALLOWED. UPDATE-EMPLOYEE. MOVE SPACES TO W6-20-100.
P EF-SCR 00500 IF W7-10-269 EQUAL 'LMODEL' MOVE SPACES TO W7-15-264.
P EF-SCR 00600 IF WORK NOT EQUAL '*' AND '@' AND '?'
```

Exit

---

## NOTES

---

## Displaying System Control Repository data

### The Display Utility (DISPLY)

The Display Utility (DISPLY) is used to display a specified System Control Repository (FILE01) type on the form.

To display a specific System Control Repository (FILE01) type, perform the following steps:

<b>Component:</b>		Development Tools
<b>Process:</b>		Option Lists
<b>Task:</b>		Display

**Result:** The Display Utility form (DISPLY) is displayed.

1. Select the Object you want to display.
2. Type the name of the Object Key you want to display. A wildcard, '=', may be used for any character of the Object Key. A blank Object Key means 'All'.
3. Mark the '1st Line Only' check box if only the first line of the object(s) is to be displayed.
4. Click the Save this form button or press Enter.

**Result:** The requested records are displayed.

# PURGE

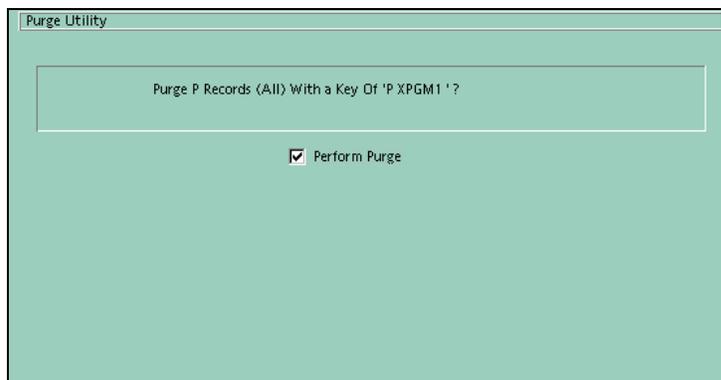
## System Control Repository Record Delete Utility



Purge Utility

Object: P Records (All)

Object Key: XPGM1



Purge Utility

Purge P Records (All) With a Key Of 'P XPGM1 ' ?

Perform Purge

---

### NOTES

---

## Purging System Control Repository data

### The Purge Utility (PURGE)

The Purge Utility (PURGE) allows you to delete a selected record of a specified object type from the System Control Repository (FILE01). The system verifies deletion of a record before the actual purge is performed and allows you to cancel the PURGE command.

<b>Component:</b>		Development Tools
<b>Process:</b>		System Control Repository Utilities
<b>Task:</b>		Purge

**Result:** The Purge Utility form (PURGE) is displayed.

1. Select the Object you want to purge.
2. Type the name of the program, option list, text file, field, and so forth, which you want to purge, in the Object Key field.
3. Click the Save this form button or press Enter.
4. Select the Perform Purge check box and press Enter. To cancel the purge, make sure the Perform Purge box is deselected and press Enter.

**Result:** The object is purged.

# Exporting/Importing System Control Repository Data

- **EXPORT**
  
- **MAINTI**

---

## NOTES

---

## Exporting/Importing System Control Repository data

### Export/Import data

There are two utilities used to export and import System Control Repository (FILE01) data. They are EXPORT and MAINTI.

### EXPORT

The EXPORT program extracts a copy of all or selected records of a specified object from the System Control Repository (FILE01).

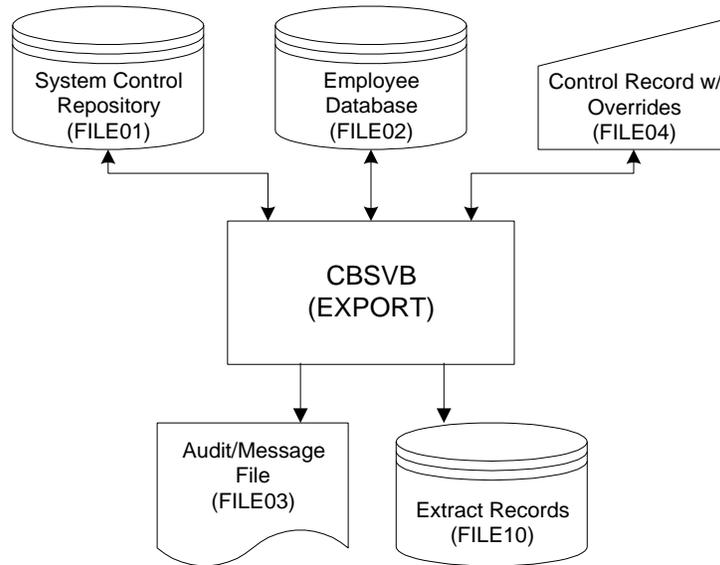
- The entire record group or just the first record of each group can be extracted.
- Records are written to FILE10 in System Control Repository record format.

### MAINTI

The MAINTI program is used to import maintenance records to the System Control Repository (FILE01). The program is used in the following ways:

- During installation of a new software release, MAINTI is used to apply user changes to the new System Control Repository (FILE01).
- When a software update is issued, MAINTI is used to apply the maintenance records to the production System Control Repository (FILE01).
- When moving programs from a test environment to the production System Control Repository (FILE01).

# Export Utility (EXPORT)



**Input Files:**     **FILE01 System Control Repository**  
                   **FILE02 Employee Database**  
                   **FILE04 Control Record File**

**Execute:**       **CBSVB**

**Output files:**  **FILE03 Audit/Message File**  
                   **FILE10 Extracted Records**

**EXPORT Control Record:**

**Extracting CSL Source Code:**

```

1 1 2 2 3 3 4 4 5 5
5...0...5...0...5...0...5...0...5...0...5
EXPORT P/ HELLO
      | |
      OBJECT OBJECT-KEY 1ST-LINE-ONLY
    
```

**Extracting ALL Human Resource Module Option List Values:**

```

1 1 2 2 3 3 4 4 5 5
5...0...5...0...5...0...5...0...5...0...5
EXPORT C/VHR==
      | |
      OBJECT OBJECT-KEY 1ST-LINE-ONLY
    
```

---

## NOTES

---

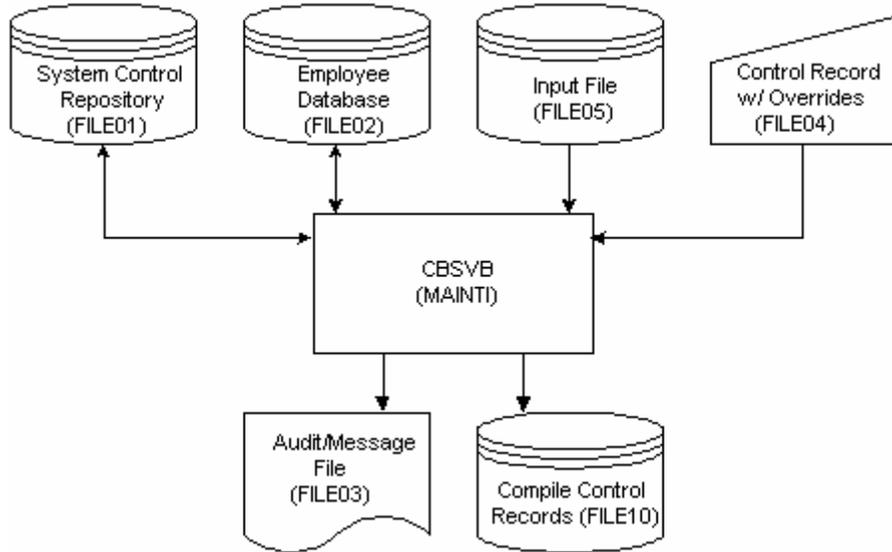
**Exporting/Importing System Control Repository data, continued****Using EXPORT**

To execute EXPORT, you must provide a control record as FILE04. The control record must reflect the object type and record name to be exported.

- To specify more than one record, you may use the equal sign (=) as a wild card for any character. For example, to print all programs that begin with an X, enter X=====.
- To specify ALL records of the designated record code type, leave the Object-key field blank.

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Program Field</b>	23–28	Type 'EXPORT'
<b>Key field</b>	31–33	Type the object code of the data to be extracted.
	34–40	Type the object-key of the specific record(s) to be extracted (first 7 positions).
<b>Additional Key</b>	41–53	Type the object-key of the specific record(s) to be extracted (8th–20th positions).
	54	Type a 'Y' to extract the first line only for each object.

# Updating The System Control Repository (MAINTI)



**Input Files:**        **FILE01 System Control Repository**  
                          **FILE02 Employee Database**  
                          **FILE04 Control Record File**  
                          **FILE05 System Control Repository Updates**

**Execute:**            **CBSVB**

**Output files:**      **FILE03 Audit/Message File**  
                          **FILE10 Compile Control Record**

**MAINTI Control Record:**

1    1    2    2    3    3    4    4    5    5  
 5 . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5 . . . . 0 . . . . 5  
 MAINTI

---

## NOTES

---

**Exporting/Importing System Control Repository data, continued****System Control Repository update**

The utility to update the System Control Repository is referred to as the Maintenance In Utility (MAINTI). To execute the utility, you must provide a control record as FILE04.

PROGRAM Field

<b>Position</b>	<b>Option/Step/Description</b>
23–28	Type 'MAINTI'

**Sequential input FILE05**

Updates to the System Control Repository (FILE01) are input in the MAINTI process from FILE05.

- Each record must be in the proper control record structure to be accepted.
- Control records must be marked with an (A) Add, (C) Change, or (D) Delete in the 80th column of each record denoting the function to be performed.

**Sequential output FILE10**

Compile transactions are output to FILE10 for each program that is updated by the MAINTI updates. These may include RELOAD, RETYPE, RECALC or REEDIT control records depending on the type of source code maintained.

- These transactions can then be used to compile the programs to have the changes take effect.

## **Program/Field Reports**

- **Field Table List (FTLIST)**
- **Field Menu (F-MENU)**
- **Screen Label to Field Name Cross–Reference (FLABEL)**
- **Segment Layout Report (SRTFLD/F-SEGM)**
- **Field to Program Cross–Reference Report (CROSSX/CROSSP)**
- **Program Memory Map (MAPRPT)**

---

**NOTES**

---

## Program/Field reports

### **Print/Display System Control Repository data**

There are several utilities available for displaying System Control Repository data that will assist in the programming effort.

### **Field table list**

The Field Table List program (FTLIST) provides an alphabetical list of field definitions online or as a printout in batch mode. The online display is in scrolling format. A batch run produces a printout with headings and page numbers.

### **Field table menu**

The Field Menu program (F-MENU) is a list in displacement order of the attributes of the data fields on file in the Field Name Table in a user-friendly, menu-driven format.

### **Label to field cross-reference**

The Screen Label To Field Name Cross Reference form (FLABEL) provides a cross-reference between the field labels used on a form and their Data Dictionary field names.

### **Segment Layout Report**

The Segment Layout Report programs (SRTFLD/F-SEGM) produce a report that displays each segment's layout.

### **Cross-Reference Report**

The Cross-Reference Report programs (CROSSX/CROSSP) produce a report that cross-references all fields and verbs.

### **Program Memory**

The Program Memory Map (MAPRPT) displays a report of the memory addresses used within a program.

# Field Table List (FTLIST)

FIELD DISPLAY/REPORT	FTLIST
<p>This program provides an inquiry-only scrolling list (online) or a printed report (batch) of all or selected fields. Fields may be selected based on pointer number and/or module code.</p>	
<p>Complete the text boxes below, then hit enter: <input type="text" value="X"/></p>	
<p>Optionally, enter the following:</p>	
<p>Pointer number: <input type="text" value="29"/> default is all pointers</p>	
<p>Module code: <input type="text" value="PP"/> default is all module codes</p>	
<p>Starting point: <input type="text" value="A"/></p>	

FIELD-NAME	PTR	LTH	DSP	DATA-TYPE	DCMLS	FIELD-TYPE	COSET	SC	MC
ACTION-CODE	29	001	040	ALPHANUMERIC				EA	PP
ADJ-BATCH	29	001	043	ALPHANUMERIC				EA	PP
BANK-CD	29	002	048	NUMERIC	0			EA	PP
BIRTH-DATE	29	006	032	CYYMDD DATE				E	PP
BIRTH-DATE-OLD	29	006	032	YYMMDD DATE				E	PP
BIRTH-DAY	29	002	036	NUMERIC	0			E	PP
BIRTH-MONTH	29	002	034	NUMERIC	0			E	PP
BIRTH-MONTH-DAY	29	004	034	YYMM OR MMDD				E	PP
BIRTH-YEAR	29	002	032	NUMERIC	0			E	PP
BIRTH-YY-MM	29	004	032	YYMM OR MMDD				E	PP
CHANGE-DATE	29	006	034	CYYMDD DATE				EA	PP
CLEAR-ANNIV	29	001	070	ALPHANUMERIC				E	PP
COMMISSION-FLAG	29	001	061	ALPHANUMERIC			PP24	E	PP
COMMISSION-FLAG-DESC	29	020	061	ALPHANUMERIC		CODESET DESC	PP24	E	PP
DATE-OF-TERM	29	006	048	CYYMDD DATE				E	PP
DATE-OF-TERMINATION	29	006	048	CYYMDD DATE				E	PP
E-BANK-CODE	29	002	048	ALPHANUMERIC				EA	PP
E-CARD-CODE	29	002	001	ALPHANUMERIC				EA	PP
E-PERIOD-DATE	29	006	059	CYYMDD DATE				EA	PP
E-RECON-CLEAR	29	001	058	ALPHANUMERIC				EA	PP

## NOTES

## Program/Field reports, continued

### Using field table list

You can select the entire Field Name Table or you can limit your selection to those Field Names with a specific pointer number and/or module code.

- The search may take several minutes. The more parameters you list, the longer the search takes.
- The data you receive are the Field Name pointer number, length, displacement, data type for your machine, whether it is connected to an Option List, segment code and module code.

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** Fields and Verbs  
**Task:**  Display Alphabetically

**Result:** The FTLIST program prompts display.

1. Type a Pointer number (optional).
2. Type a Module code (optional).
3. Type the beginning field name you wish to display in the Starting Point field (optional).
4. Click the Save this form button or press Enter.

**Result:** An alphabetic list of the fields for the pointer and module you specified displays.

# Field Menu and List (F-MENU)

Field Menu Page: 1

---

Main Menu

Clear Selection
  Company Fields
  Employee Fields
  Time Entry/Adj Flds

Table Fields
  Work Fields
  Other Record Fields

Field Menu Page: 1

---

Selected Menu: Employee Fields

Previous Menu
  Employee Identifier
  Basic Applicant Data

Basic Employee Data
  Employee Pay Data

Employee Transfer
  Applicant Name+Addr

Name and Address
  W2 Name and Address

Applicant Job Data
  Pay Allocations

Employee HED's
  Labor/Hist HED Amt's

Field Menu Page: 1

---

Selected Menu: Basic Employee Data  
Segment Type: E Segment Code: E

Main Menu
  Previous Menu
  Clear Option

Field Name	Field Type	Mc	Cdset	Length	Disp	Data Type	Rdb
E-SEGMENT	Data Field	PP		60	3	Alpha/numeric	N
SIN-1	Data Field	PP		1	3	Numeric: 0 Decimals	N
SIN-1-3	Data Field	PP		3	3	Numeric: 0 Decimals	N
SOCIAL-INSURANCE-NBR	Data Field	PP		12	3	Alpha/numeric	N
SOCIAL-SEC-NBR	Data Field	PP		12	3	Alpha/numeric	Y
SOCIAL-SECURITY-NBR	Data Field	PP		12	3	Alpha/numeric	N
SSN-1-3	Data Field	PP		3	3	Numeric: 0 Decimals	N
SIN-2	Data Field	PP		1	4	Numeric: 0 Decimals	N
SIN-3	Data Field	PP		1	5	Numeric: 0 Decimals	N
SIN-4	Data Field	PP		1	6	Alpha/numeric	N
SSN-4	Data Field	PP		1	6	Alpha/numeric	N

## NOTES

## Program/Field reports, continued

### Field menu

The Field Menu (F-MENU) program displays a Field List similar to the Field Table List (FTLIST) program except the information is categorized by record and segment and is in displacement order.

- The Field Menu (F-MENU) program navigates you through a list of records and segments by descriptive names to determine the information to display.
- The form contains the field attributes of segment-type, segment-code, field-name, field-type, module, Option List, length, displacement, and the data-type.

Make the following selections from the Navigator:

**Component:**  Development Tools  
**Process:** Fields and Verbs  
**Task:**  Menu

**Result:** The Field Menu form (F-MENU) displays.

1. Select a Main Menu option.

**Result:** The next level of the Field Menu displays.

2. Select an option from the Selected Menu choices.
3. Continue to select menu options to navigate to the Field Definition display.

# Screen Label To Field Name Cross Reference (FLABEL)

Screen Label	Field Name
--Entry Fields	
HED>	HED-NUMBER
Frequency:	HED-EARNING-FREQ-CD
Start Method:	START-CODE
Type:	HED-EARNING-TYPE-CD
Start Value:	START-FIELD
Calc Method:	HED-EARNING-METHOD-C
Stop Method:	STOP-CODE
Amount/Pct:	AMOUNT-PERCENT
Stop Value:	STOP-FIELD
One-time Rule:	ONE-TIME-CODE
One-time Amt:	ONE-TIME-AMOUNT
Code:	USER-CODE
Amount One:	AMOUNT-ONE
Number:	USER-NUMBER
Amount Two:	AMOUNT-TWO
HED>	HED-NUMBER
Frequency:	HED-DEDUCT-FREQ-CD

---

## NOTES

---

## Program/Field reports, continued

### Screen Label to Field Name Cross Reference

The Screen Label To Field Name Cross Reference (FLABEL) program displays a cross-reference between the label on the online form and the data dictionary field name used in programming.

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Fields and Verbs
- Task:**  Display Field Names for a form

1. Type the program name in the Enter Screen Name text box.
2. Click the Save this form button or press Enter.

**Result:** The Screen Label To Field Name Cross Reference form (FLABEL) displays for the requested program name.

# Segment Layout Report

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	B
1 1	B-SEGMENT-TYPE	001	X(001)	
2 4	COMPANY-HED-NUMBER	003	9(003)	K
2 4	HED-NUM	003	9(003)	K
2 4	HED-NUMBER-1	003	9(003)	K
5 6	CATEGORY-CODE	002	9(002)	
5 6	DEDUCT-CATEGORY-CODE	002	9(002)	PP02
5 6	EARNING-CATEGORY-CD	002	9(002)	PP01
5 6	DEDUCTION-CATEGORY	002	X(015)	D PP02
5 6	EARNING-CATEGORY	002	X(015)	D PP01
7 21	HED-NAME	015	x(015)	
22 22	VACATION-FLAG-CODE	001	9(001)	PP13
22 22	VACATION-FLAG	001	9(015)	D PP13
23 25	PERMANENT-ORDER	003	9(003)	
26 28	TEMPORARY-ORDER	003	9(003)	
29 29	TC-2-HOURS	001	9(001)	PP14
29 29	TC-2-HOURS-EDIT	001	9(015)	D PP14
30 30	TC-2-AMOUNT	001	9(001)	PP15
30 30	TC-2-AMOUNT-EDIT	001	9(015)	D PP15
31 31	EARNINGS-RATE-FLSA	001	9(001)	PP04
31 31	OVERTIME-FACTOR-CD	001	9(001)	PP03
31 31	OVERTIME-TYPE	001	9(001)	
31 31	EARNINGS-FLSA-RATE	001	9(015)	D PP04
31 31	OVERTIME-FACTOR	001	X(015)	D PP03
32 32	ARREARS-OVERTIME-CD	001	X(001)	PP06
32 32	FREQUENCY-FOR-TAX-CD	001	X(001)	PP05
32 32	TAX-FREQUENCY-CODE	001	X(001)	
32 32	ARREARS-OVERRIDE	001	X(015)	D PP06
32 32	FREQUENCY-FOR-TAX	001	X(015)	D PP05
33 34	DEDUCTION-FREQ-CODE	002	9(002)	PP08
33 34	DEFAULT-FREQUENCY	002	9(002)	
33 34	EARNING-FREQUENCY-CD	002	9(002)	PP07
33 34	DEDUCTION-FREQUENCY	002	9(015)	D PP08
33 34	EARNING-FREQUENCY	002	9(015)	D PP07
35 36	DEDUCTION-ARREARS-CD	002	9(002)	PP10
35 36	DEFAULT-TYPE	002	9(002)	
35 36	EARNING-TYPE-CODE	002	9(002)	PP09
35 36	DEDUCTION-ARREARS	002	9(015)	D PP10
35 36	EARNING-TYPE	002	9(015)	D PP09
37 38	DEDUCTION-METHOD-CD	002	X(002)	PP12
37 38	DEFAULT-METHOD	002	X(002)	
37 38	EARNING-METHOD-CODE	002	X(002)	PP11
37 38	DEDUCTION-METHOD	002	X(015)	D PP12
37 38	EARNING-METHOD	002	X(015)	D PP11
39 47	BA-AMT/PCT	009	9(009)	
39 45	DEDUCTION-AMOUNT/PCT	007	9(007)	
39 45	DEFAULT-AMOUNT/PCT	007	9(007)	
39 45	EARNING-AMOUNT/PCT	007	9(007)	
46 46	AUTOMATIC-SETUP-CODE	001	X(001)	PP16
46 46	AUTOMATIC-SET-UP	001	X(015)	D PP16
47 47	PERIOD-TABLE-CODE	001	9(001)	PP17
47 47	PERIOD-END-TABLE	001	X(015)	D PP17
48 48	ADD-TOTAL-HOURS-CODE	001	9(001)	PP18
48 48	ADD-TOTAL-HOURS	001	X(015)	D PP18
49 49	HED-REGISTER	001	X(001)	PP00

## NOTES

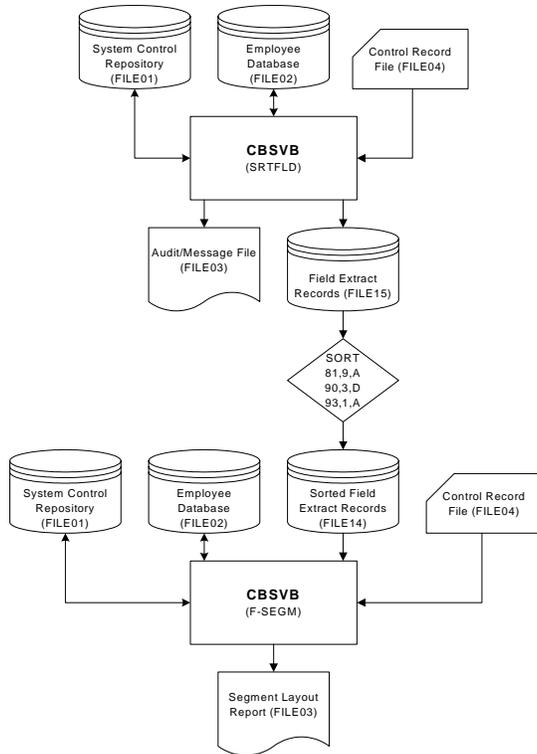
## Program/Field reports, continued

### Segment Layout Report

The Segment Layout Report shown on the opposite page is the product of the SRTFLD, Sort, and F-SEGM processes.

Notice that several fields can occupy the same displacement of a pointer. This allows the field to be redefined by various field names.

# Segment Layout Report (SRTFLD/F-SEGM)



**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File

**Execute:** CBSVB

**Output files:** FILE03 Audit/Message File  
 FILE15 Extracted Field Definitions

**SRTFLD Control Record:**

**Extracting All Field/Segment Definitions:**

```

1      1      2      2      3      3      4      4      5      8
5....0....5....0....5....0....5....0....5....0...//.0
                                SRTFLD  ALL
    
```

**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File  
 FILE14 Sorted Field/Segment Definitions

**Execute:** CBSVB

**Output files:** FILE03 Segment Layout Report

**F-SEGM Control Record:**

**Producing the Segment Layout Report for All Segments:**

```

1      1      2      2      3      3      4      4      5      5
.5....0....5....0....5....0....5....0....5....0....5
                                F-SEGM  ALL
    
```

---

## NOTES

---

## Program/Field reports, continued

### Segment Layout Report

The Segment Layout Report (SRTFLD/F-SEGM) programs provide a layout of each segment record. This procedure requires a three-step process that selects and formats the segment records for the report.

### Using SRTFLD

To execute SRTFLD, you must provide a control record as FILE04.

 Refer to Section 2: The Solution Series Architecture for a list of pointer numbers for the company, employee, and tax segments.

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Program Field</b>	23–28	Type 'SRTFLD'.
<b>Key Field</b>	31–34	Type 'ALL' to extract all segment data or the pointer number of the segment to be extracted for reporting.

### Sorting segment records

Prior to executing F-SEGM, the field definition records extracted by the SRTFLD process must be sorted.

- Reads the documentation index FILE15.
- Sorts the records on the indicated positions (81,9,A; 90,3,D; 93,1,A).
- Writes the sorted documentation to FILE14.

### Using F-SEGM

To execute F-SEGM, you must provide a control record as FILE04.

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Program Field</b>	23–28	Type 'F-SEGM'.
<b>Key Field</b>	31–34	Type 'ALL' to produce the report for all extracted segments or the pointer number of the segment to be printed on the report.

# Cross-Reference Report

	FIELD/PROGRAM	CROSS REFERENCE LIST	PAGE
			1
ANNUAL-AMOUNT-CHANGE	40-SCR 00340	40-SCR 00800 40-SCR 01560	40-SCR 02200
	40-SCR 02420	40-SCR 02740 40-SCR 02760	40-SCR 02980
	40-SCR 03320	40-SCR 03620 40-SCR 08000	40-SCR 03960
	40-SCR 04980	40-SCR 05100 40-SCR 05180	40-SCR 06080
ANNUAL-SALARY	40-SCR 00340	40-SCR 00620 40-SCR 00640	40-SCR 00760
	40-SCR 01140	40-SCR 01540 40-SCR 01940	40-SCR 01947
	40-SCR 01952	40-SCR 02040 40-SCR 02180	40-SCR 02360
	40-SCR 02380	40-SCR 02800 40-SCR 03000	40-SCR 03300
	40-SCR 03600	40-SCR 03780 40-SCR 03940	40-SCR 04760
	40-SCR 04783	40-SCR 04785 40-SCR 04940	40-SCR 04946
	40-SCR 04952	40-SCR 05000 40-SCR 05080	40-SCR 05340
	40-SCR 05940	40-SCR 05960	
ANNUALIZATION-FACTOR	40-SCR 01822	40-SCR 01824 40-SCR 01840	40-SCR 01920
	40-SCR 01943	40-SCR 01948 40-SCR 04750	40-SCR 04755
	40-SCR 04760	40-SCR 04782 40-SCR 04784	40-SCR 04900
	40-SCR 04902	40-SCR 04920 40-SCR 04942	40-SCR 04948
AUTO-KEY-SWITCH	EF-SCR 02290		
BIRTH-DATE	EF-SCR 00840	EF-SCR 00960 EF-SCR 01680	EF-SCR 01720
	EF-SCR 01760		
COMMISSION-FLAG	EF-SCR 01320		
COMPANY-ADDRESS	AA-SCR 00120		
COMPANY-ADDRESS-2	AA-SCR 00120		
COMPANY-CITY/PROV	AA-SCR 00280		
COMPANY-CITY/STATE	AA-SCR 00200		
COMPANY-NAME	AA-SCR 00080		
COMPANY-NAME-2	AA-SCR 00080		
COMPANY-POSTAL-CODE	AA-SCR 00280		
COMPANY-ZIP-CODE	AA-SCR 00200		
CONTROL-1-NAME	AA-SCR 00360		
CONTROL-2-NAME	AA-SCR 00360		
CONTROL-3-NAME	AA-SCR 00360		
CONTROL-4-NAME	AA-SCR 00380		
CONTROL-5-NAME	AA-SCR 00380		
CONTROL-6-NAME	AA-SCR 00380		
CONTROL-COUNTRY	AA-SCR 00160	EF-SCR 00520 EF-SCR 00760	EF-SCR 01000
	EF-SCR 01902	EF-SCR 02540	
COUNTER01	40-SCR 06560	40-SCR 06720 40-SCR 06720	40-SCR 06740
DATE-OF-TERMINATION	EF-SCR 01100	EF-SCR 01280	
DEF-PAY-FREQ-CODE	EF-SCR 01980	EF-SCR 02120	
DELETE-L-SEGMENT	40-SCR 07120		
EEO-RACE	EF-SCR 00800		
EMPLOYEE-NAME	EF-SCR 02240		
EMPLOYEE-NUMBER	EF-SCR 01880		
EMPLOYMENT DATE	EF-SCR 01040	EF-SCR 01220 EF-SCR 01700	EF-SCR 01720
	EF-SCR 01760		
FAIR-LABOR-CODE	EF-SCR 01140		
FREQUENCY	40-SCR 01780	40-SCR 04700 40-SCR 04860	
HOURLY-RATE	40-SCR 00260	40-SCR 00660 40-SCR 00680	40-SCR 00820
	40-SCR 00960	40-SCR 01080 40-SCR 01580	40-SCR 01900
	40-SCR 01942	40-SCR 02440 40-SCR 02820	40-SCR 03200
	40-SCR 03220	40-SCR 03420 40-SCR 03540	40-SCR 03560

---

## NOTES

---

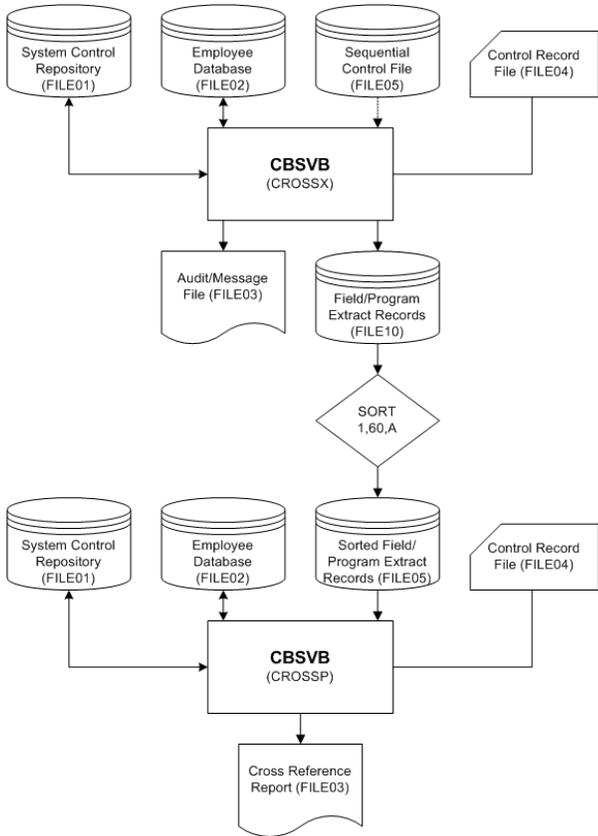
**Program/Field reports, continued**

**Cross–Reference Report (CROSSX/CROSSP)**

The Cross–Reference Report (CROSSX/CROSSP) is the product of the CROSSX, Sort, and CROSSP processes.

Notice that several programs use the same field.

# Cross-Reference Report (CROSSX/CROSSP)



## CROSSX Control Record

**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File  
 FILE05 Optional Sequential System Control Repository

**Execute:** CBSVB

**Output files:** FILE03 Audit/Message File  
 FILE10 Extracted Field/Program Records

## CROSSX Control Record:

**Extracting Cross-Reference record from FILE01:**

```

1 1 2 2 3 3 4 4 5 8
5...0...5...0...5...0...5...0...5...0...5...0...//.0
CROSS 1W
    
```

**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File  
 FILE05 Sorted Field/Program Cross-Reference

**Execute:** CBSVB

**Output files:** FILE03 Cross-Reference Report

## CROSSP Control Record:

**Producing the Cross-Reference Report with 80 char. per line:**

```

1 1 2 2 3 3 4 4 5 5
5...0...5...0...5...0...5...0...5...0...5
CROSSP N
    
```

## NOTES

**Program/Field reports, continued****Cross–Reference report**

The Cross–Reference Report (CROSSX/CROSSP) provides cross-reference for all fields and verbs used in a CSL program.

To execute CROSSX, you must first determine which file will be used as input to the cross–reference program (FILE01 or FILE05). This designation is made in the JCL and the control record FILE04.

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Program</b>	23–28	Type 'CROSSX'.
<b>Field</b>		
<b>Key Field</b>	31	Type 1 to read FILE01 rather than FILE05.
	32	Type 'W' to include the work fields.

**Sorting cross–reference records**

Prior to executing CROSSP, the field and verb records extracted by the CROSSX process must be sorted.

- Reads the extracted cross–reference records FILE10.
- Sorts each record from positions 1 for a length of 60, ascending sequence.

To execute CROSSP, you must provide a control record as FILE04.

	<b>Position</b>	<b>Option/Step/Description</b>
<b>Program</b>	23–28	Type 'CROSSP'.
<b>Field</b>		
<b>Key Field</b>	31	Type 'N' to produce a report that is 80 characters wide. 80 characters fit on standard 8-1/2 x 11 inch paper.

# Program Memory Map Report (MAPRPT)

FILE: 24-SCR

NAME	POINTER NUMBER	FIELD DISPLACEMENT	FIELD LENGTH	SEGMENT/TABLE VERB	DATA TYPE	DECIMAL PLACES	COMMENT
W6-06-083	06	083	006	VB			VERB
W7-06-012	07	012	006	VB			VERB
SCREEN-ERROR	07	093	001		0	0	
W7-04-093	07	093	004	VB			VERB
RECORD-UPDATED	07	097	001		0	0	
SPACES	07	176	001		0	0	
W7-10-176	07	176	010	VB			VERB
W7-15-176	07	176	015	VB			VERB
W7-22-176	07	176	022	VB			VERB
W7-06-246	07	246	006	VB			VERB
W7-01-254	07	254	001	VB			VERB
W7-06-254	07	254	006	VB			VERB
W7-10-254	07	254	010	VB			VERB
W7-24-255	07	255	024	VB			VERB
W7-15-264	07	264	015	VB			VERB
W7-01-298	07	298	001	VB			VERB
W7-01-301	07	301	001	VB			VERB
W7-01-302	07	302	001	VB			VERB
W7-01-464	07	464	001	VB			VERB
WORK-DATE	08	216	006		8	2	
W8-04-424	08	424	004	VB			VERB
W8-02-484	08	484	002	VB			VERB
SCREEN	11	000	000	VB	0	0	VERB
POINTER-14	14	000	000	VB			VERB
WORK	14	000	000	VB			VERB
30030004	30	004	030	VB			VERB
FLEET-ID	36	003	005	ZJ	0	0	
AUTOMOBILE-DATE	36	008	006	ZJ	8	2	
AUTOMOBILE-NUMBER	36	014	015	ZJ	0	0	
AUTOMOBILE-MAKE	36	029	010	ZJ	0	0	
AUTOMOBILE-MODEL	36	039	010	ZJ	0	0	
COLOR	36	049	005	ZJ	0	0	
MILES	36	054	006	ZJ	1	0	
DATE-LAST-SERV-MM	36	060	002	ZJ	1	0	
DATE-LAST-SERV-YY	36	062	002	ZJ	1	0	
DATE-RECOVERED	36	064	006	ZJ	8	0	
----Complete----							

---

## NOTES

---

## Program/Field reports, continued

### Program Memory Map report

The Program Memory Map program (MAPRPT) is a batch report that prints a program's memory map, exclusive of verbs.

To run the Program Memory Map report (MAPRPT), create a Report Group using the Report Group Activities form (RGMSTR), containing MAPRPT as the report name and enter the name of the program to be mapped in the parameters form. Additional program names may be entered in a batch run using FILE05 with the program names in positions 1 through 6.

## Section Summary

- **Additional edit commands**
- **Copying and displaying System Control Repository data**
- **Purging System Control Repository data**
- **Importing/Exporting System Control Repository data**
- **Program/Field reports**

---

NOTES

---

## Section summary

In this section, you learned the additional utilities used to maintain System Control Repository records. Specifically you learned:

### Additional edit commands

---

---

---

### Copying System Control Repository data

---

---

---

### Displaying System Control Repository data

---

---

---

### Purging System Control Repository data

---

---

---

### Importing System Control Repository data

---

---

---

### Exporting System Control Repository data

---

---

---

### Program/Field reports

---

---

---

## Section 4 Exercise

4.

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8  
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0

6.

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8  
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0

7.

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8  
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8  
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0

---

**NOTES**

---

## Section 4 exercise

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in section 4. Complete the tasks below:

1. Resequence the program you created in the previous section.
2. Copy the program you created in the previous section to a new program name.
3. Display the records in the Option List SC03.
4. Create the 80-character control record to extract the program you created in the previous section to FILE10.
5. Delete the program you created in the previous section.
6. Create the 80-character control record to import the program you extracted.
7. Create the two 80-character control records to generate a cross-reference report for the program you extracted to FILE10 in question 4.
8. Use FTLIST or F-MENU to answer the following questions.
  - a. What is the Data Type of the BIRTH-DATE field in Pointer 29?  
\_\_\_\_\_  
\_\_\_\_\_
  - b. What is the Segment Type for the 'Basic Employee Data'?  
\_\_\_\_\_  
\_\_\_\_\_
  - c. What is the Option List for Sex Code in the 'Basic Employee Data'?  
\_\_\_\_\_  
\_\_\_\_\_

---

**NOTES**

---

## Section 5: Cyborg Scripting Language Plus

---

### Table of Contents

Introduction.....	155
Moving data .....	157
Comparing data.....	171
Calculating .....	183
Transferring control within a program.....	185
Section summary.....	201
Section 5 exercise .....	205

## Objectives

- **Identify and apply the rules associated with moving data**
- **Identify and apply the rules associated with comparing data**
- **Perform calculations**
- **Transfer control within a program**
- **Transfer control to other programs**

---

### NOTES

---

## Introduction

### **Purpose**

In this section you will learn additional CSL verbs and techniques.

### **Objectives**

Upon completion of this section you will be able to:

- Identify and apply the rules associated with moving data
- Identify and apply the rules associated with comparing data
- Perform calculations
- Transfer control within a program
- Transfer control to other programs

# MOVE

## Format

$$\underline{\text{MOVE}} \left\{ \begin{array}{l} \textit{literal} \\ \textit{field - name - 1} \\ \textit{figurative - constant} \end{array} \right\} \underline{\text{TO}} \textit{field - name - 2}.$$

## Example

```
MOVE :10000.00 TO ANNUAL-SALARY.  
MOVE 'F' TO SEX-CODE.  
MOVE EMPLOYMENT-DATE TO ORIGINAL-HIRE-DATE.  
MOVE SPACES TO UNION-CODE.
```

## Review of literals

### Numeric literals

- Must be preceded with a : colon.
- Decimal point (optional, but may not be the last character).

### Alphanumeric literals

- Must be enclosed in ' ' single quotes.
- Any characters may be used (except the single quote mark).
- 

---

## NOTES

---

## Moving data

### **MOVE statement**

The MOVE verb is used to transfer data from one field to another.

- The data represented by the literal, field-name-1, and figurative-constant are called the sending field.
- The data represented by field-name-2 is called the receiving or result field.
- The contents of the sending field will be transmitted, or copied, to the receiving field as a result of the MOVE operation.

### **Field name**

Recall that a Field is a unique piece of data. The Field Name is used in a program to indicate the content of the Field's data.

### **Literals**

Recall that there are two kinds of literals: numeric and alphanumeric. The rules for forming these literals are detailed on the opposite page.

### **Figurative constant**

A Figurative Constant is a CSL reserved field, such as SPACE (SPACES) or QUOTE, which represents a specific value.

# Numeric MOVE

## Example 1:

MOVE FIELDA TO FIELDB.

### FIELDA:

Numeric no decimals

Contents: 123

### FIELDB:

Numeric no decimals

Contents: 456

#### Before MOVE

FIELDA	FIELDB
1   2   3	4   5   6

#### After MOVE

FIELDA	FIELDB
1   2   3	1   2   3

## Example 2:

MOVE FIELDC TO FIELDDD.

### FIELDC:

Numeric 2 decimals

Contents: 12.34

### FIELDDD:

Numeric 2 decimals

Contents: 45.67

#### Before MOVE

FIELDC	FIELDDD
1   2   3   4	4   5   6   7

#### After MOVE

FIELDC	FIELDDD
1   2   3   4	1   2   3   4

---

## NOTES

---

## Moving data, continued

### Numeric MOVE

Often in CSL it is necessary to move one numeric field or literal to another numeric field. It is also possible to move data to a field with a different format.

To move numeric data successfully between different numeric fields, it is necessary to understand the following types of numeric moves:

- Sending and receiving fields have the same format.
- Sending and receiving fields have different formats.

### Sending and receiving field same format

When the sending and receiving fields have the same format (that is, Data Type and Decimal Places), the receiving field will contain the same content as the sending field.

#### Example 1

If the format of both fields is identical, the contents of FIELDB will be replaced with the contents of FIELDA; the sending field will be unchanged.

#### Example 2

Because both fields have the same format, the receiving contents of FIELDD will be replaced with the contents of FIELDC; the sending field will be unchanged.

12^34 is interpreted by the computer as 12.34 (decimal points do not actually appear in numeric fields). In other words, the MOVE operation maintains decimal alignment.

*Note:* The '^' symbol represents an implied decimal point that does not actually appear in numeric fields.

# Numeric MOVE

## Example 3:

MOVE FIELD1 TO FIELD2.

### FIELD1:

Numeric no decimals

Contents: 123

### FIELD2:

Numeric no decimals

Contents: 4567

## Example 4:

MOVE FIELD3 TO FIELD4.

### FIELD3:

Numeric 0 decimals

Contents: 1234

### FIELD4:

Numeric 0 decimals

Contents: 456

#### Before MOVE

FIELD1	FIELD2
1   2   3	4   5   6   7

#### After MOVE

FIELD1	FIELD2
1   2   3	0   1   2   3

#### Before MOVE

FIELD3	FIELD4
1   2   3   4	4   5   6

#### After MOVE

FIELD3	FIELD4
1   2   3   4	2   3   4

---

## NOTES

---

## Moving data, continued

### **Sending and receiving field different format**

When the sending and receiving fields have different formats, for example, data types or decimal places, the receiving field determines the result of the MOVE operation.

### **Numeric MOVE rule**

Moving integer portions (numbers to the left of the decimal) of numeric fields.

- When moving an integer-sending field or an integer portion of a numeric-sending field to a numeric-receiving field, movement is from right to left. All non-filled high-order (left most) integer positions of the receiving field are replaced with zeros.

### **Example 3**

Movement is from right to left, and all non-filled high-order positions are zero filled. No portion of the original contents of the receiving field is retained.

### **Example 4**

Movement is still from right to left in this example, however the receiving field is smaller and the MOVE terminates after the high-order position is filled. This is known as truncation.

# Numeric MOVE

## Example 5

MOVE FIELDE TO FIELDF.

### FIELDE:

Numeric 1 decimals

Contents: 12^3

### FIELDF:

Numeric 2 decimals

Contents: 456^78

#### Before MOVE

FIELDE			FIELDF				
1	2	3	4	5	6	7	8

#### After MOVE

FIELDE			FIELDF				
1	2	3	0	1	2	3	0

## Example 6:

MOVE FIELDG TO FIELDH.

### FIELDG:

Numeric 2 decimals

Contents: 123^45

### FIELDH:

Numeric 1 decimals

Contents: 45^6

#### Before MOVE

FIELDG					FIELDH		
1	2	3	4	5	4	5	6

#### After MOVE

FIELDG					FIELDH		
1	2	3	4	5	2	3	5

---

## NOTES

---

**Moving data, continued****Numeric MOVE rule:**

Moving decimal portions (numbers to the right of the decimal) of numeric fields.

- When moving a decimal or fractional portion of a numeric sending field to the fractional portion of a numeric receiving field, movement is from left to right.
- All non-filled low-order (right most) decimal positions of the receiving field are replaced with zeros.
- When the receiving field's decimal portion is smaller than the sending field's, the decimal result is rounded based on the receiving field's rounding rule.

**Example 5**

According to the first rule, movement is from right to left for the integer portion and all non-filled high-order positions are zero filled. According to this rule, movement is from left to right for the decimal portion and all non-filled low-order positions are zero filled.

**Example 6**

Movement is still from right to left for the integer portion and right to left for the decimal portion in this example, however the receiving field's integer and decimal portions are smaller and the MOVE terminates after the high-order position and low-order positions are filled. However, the decimal portion is not truncated, it is rounded based on the receiving field's rounding rule.

# Alphanumeric MOVE

**Example 1:**

MOVE FIELD-A TO FIELD-B.

**FIELD-A:**

Alphanumeric 3 positions

Contents: ABC

**FIELD-B:**

Alphanumeric 3 positions

Contents: DEF

**Before MOVE**

<b>FIELD-A</b>	<b>FIELD-B</b>
A   B   C	D   E   F

**After MOVE**

<b>FIELD-A</b>	<b>FIELD-B</b>
A   B   C	A   B   C

---

## NOTES

---

**Moving data, continued****Alphanumeric MOVE**

An alphanumeric MOVE is movement of an alphanumeric, alphabetic or numeric field or literal to an alphanumeric field.

To move alphanumeric data successfully between different formats (that is, storage length) it is necessary to understand the following types of alphanumeric moves:

- Sending and receiving fields have the same format.
- Sending and receiving fields have different formats.

**General rule**

When moving an alphanumeric, data is transmitted from the sending field to the receiving field from left to right. However, right most positions of the receiving field that do not receive data are not filled with spaces.

A good rule of thumb to follow in alphanumeric MOVE operations is to be sure that the receiving field is at least as many positions as the sending field.

**Sending and receiving field same format**

When the sending and receiving fields have the same storage length, the receiving field will contain the same content as the sending field.

**Example 1**

If the size of both fields is identical, the contents of FIELD-B will be replaced with the contents of FIELD-A; the sending field will be unchanged.

# Alphanumeric MOVE

## Example 2:

MOVE FIELD-C TO FIELD-D.

### FIELD-C:

Alphanumeric 6 positions

Contents: A1B2C3

### FIELD-D:

Alphanumeric 3 positions

Contents: 123

#### Before MOVE

FIELD-C	FIELD-D
A   1   B   2   C   3	1   2   3

#### After MOVE

FIELD-C	FIELD-D
A   1   B   2   C   3	A   1   B

## Example 3:

MOVE FIELD-E TO FIELD-F.

### FIELD-E:

Alphanumeric 4 positions

Contents: BILL

### FIELD-F:

Alphanumeric 6 positions

Contents: XYZ123

#### Before MOVE

FIELD-E	FIELD-F
B   1   L   L	X   Y   Z   1   2   3

#### After MOVE

FIELD-E	FIELD-F
B   1   L   L	B   1   L   L   ?   ?

---

## NOTES

---

## Moving data, continued

### **Sending and receiving field different format**

When the sending and receiving fields have different storage lengths special rules apply:

If the sending field or literal is longer than the receiving field, MOVE transfers data based on the receiving field's size, therefore truncation will occur.

If the sending field or literal value is shorter than the receiving field, MOVE transfers data based on the location of the sending field. In other words, you must know what pointer defines the sending field. These rules apply:

- For fields in Pointers 1–17 and Pointer 40  
the length of the receiving field governs the amount of data moved. The receiving field is completely filled by the operation. This means that MOVE transfers the sending field and as many bytes that follow as needed to fill the receiving field.
- For fields in Pointers 18–39 and 41–63  
MOVE transfers the sending field and leaves any remaining data in the receiving field intact.

### **Example 2**

According to the general rule, movement is from left to right. However, the receiving field is shorter and the MOVE terminates after the low-order position is filled. This is known as truncation.

### **Example 3**

Movement is still from left to right in this example, however the receiving field is longer and the MOVE operation fills the low-order positions based on the pointer of the sending field.

# MOVE Summary

## General

- The format of the receiving field determines the type of MOVE operation that is performed—numeric or alphanumeric.

## Numeric MOVE

- Receiving field is numeric
- Integer portion
  - Movement is from right to left.
  - Non-filled high-order positions are replaced with zeros.
  - Truncation of high-order digits occur if the receiving field's integer position is not long enough to accommodate the results.
- Decimal portion
  - Movement is from left to right, beginning at the decimal point.
  - Non-filled low-order positions are replaced with zeros.
  - Rounding of low-order digits occur if the receiving field is not long enough to accommodate the results.

## Alphanumeric MOVE

- Receiving field is alphanumeric.
- Movement is from left to right.
- Non-filled low-order positions are not replaced with spaces, but will remain intact or are filled with the contents in memory to the right of the sending field based on the pointer of the sending field.
- Truncation of low-order digits occurs if the receiving field is not long enough to accommodate the results.

---

## NOTES

---

## Moving data, continued

### **MOVE summary**

In summary, the format of the receiving field determines the rules to be applied in a MOVE operation.

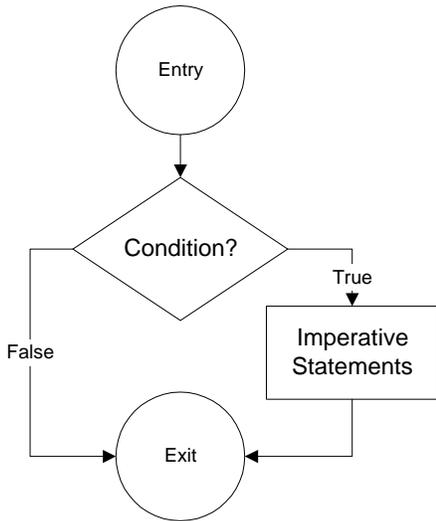
- By format we mean the data type, number of decimals, and the storage length of the field.
- The outline shown on the opposite page summarizes the MOVE rules.

# Comparing Data IF/ELSE

**Format:**     *IF condition*  
                  *imperative statement ...*  
                  [*ELSE*  
                  *imperative statement ...*]

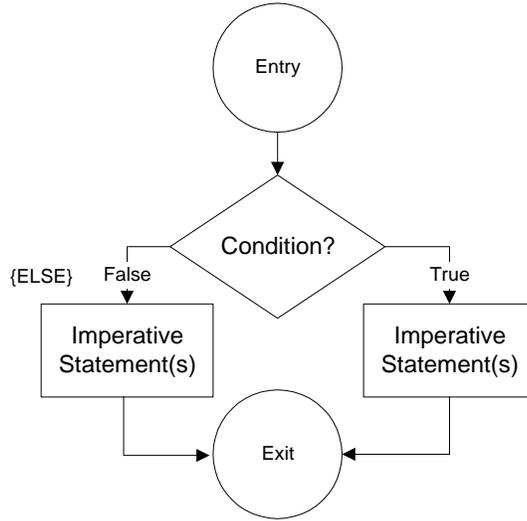
**Example 1:**

```
IF SALARY GREATER :1000.00  
  MOVE 'Y' TO USER-FIELD-E.
```



**Example 2:**

```
IF SEX-CODE EQUALS 'F'  
  PRINT 'FEMALE'  
ELSE  
  PRINT 'MALE' .
```



---

## NOTES

---

## Comparing data

### Conditional statement

A conditional statement is a statement that performs an operation dependent on the existence of some condition. Such statements, in CSL, generally begin with the verb IF.

### Imperative statement

An imperative statement is a statement that issues a command to be executed regardless of any existing conditions.

### Relational conditions

A condition may test for a specific relation. A simple condition may be a single test of the following form:

- EQUALS
- [IS] EQUAL [TO]
- [IS] GREATER [THAN]
- [IS] LESS [THAN]
- ELSE

*Note:* *IS, TO, and THAN are optional connectors.*

If the specified relationship is true, the system executes the statement or statements after the comparison.

If the relationship is false, processing continues with the first statement after a period or an ELSE verb.

### Relation data

The data used to test for a specific relation may be:

- Field to field comparison
- Field to literal comparison

# Comparing Data IF/ELSE

**Format:**

$$\text{IF } \underline{\text{field - name - 1}} \text{ [IS] } \left\{ \begin{array}{l} \underline{\text{GREATER}} \text{ [THAN]} \\ \underline{\text{LESS}} \text{ [THAN]} \\ \underline{\text{EQUAL}} \text{ [S] [TO]} \end{array} \right\} \left\{ \begin{array}{l} \underline{\text{field - name - 2}} \\ \underline{\text{literal}} \end{array} \right\}$$

*imperative statement...*  
[NEXT SENTENCE]

ELSE

*imperative statement...*  
[NEXT SENTENCE].

**NEXT SENTENCE Example:**

```
IF RELATIONSHIP-CODE IS EQUAL TO '04'  
    NEXT SENTENCE  
ELSE  
    MOVE 'N' TO USER-FIELD-E.
```

---

## NOTES

---

## Comparing data, continued

### **NEXT SENTENCE**

There are times when you might want to execute a series of steps only if a certain condition does not exist.

- **NEXT SENTENCE** will enable you to avoid performing any operation if the condition exists. Execution will continue with the next statement following the **IF/ELSE**.
- If **NEXT SENTENCE** is coded it must be the only imperative statement following the condition.

### **Ending conditional statements**

As indicated, several imperative statements may appear within one conditional. The end of a conditional statement is indicated with a period.

- The placement of periods can affect the logic in an **IF/ELSE** statement.
- If you are using an **ELSE** statement, never place a period before the **ELSE**; code the period at the end of the sentence following all imperative statements that apply to the **ELSE**.
- Indentation is used to make the entire statement easier to read.

# Comparing Data IF/ELSE

## Numeric Comparison

### Example 1:

FIELD A	FIELD B
1   2   3	1   2   3   9   9

```
IF FIELD A GREATER THAN FIELD B
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

FALSE

### Example 2:

FIELD C	FIELD D
1   2   3	0   1   2   3   0

```
IF FIELD C EQUAL TO FIELD D
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

TRUE

## Alphanumeric Comparison

### Example 3:

FIELD 1	FIELD 2
A   B   C	A   B   C   D

```
IF FIELD 1 EQUALS FIELD 2
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

FALSE? / TRUE?

### Example 4:

FIELD 3	FIELD 4
A   B   C   D	A   B   C

```
IF FIELD 3 EQUALS FIELD 4
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

TRUE

## Class Test

### Example 5:

FIELD X
1   5   4   2   5

```
IF FIELD X IS NUMERIC
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

TRUE

### Example 6:

FIELD Y
A   B   2

```
IF FIELD Y IS ALPHABETIC
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
```

### Output:

FALSE

---

## NOTES

---

## Comparing data, continued

### Comparison rules

As with the MOVE operation, numeric and alphanumeric fields/literals are evaluated based on special rules.

### Numeric comparison

Numeric fields/literals are compared algebraically. Even though numbers may not have the same internal configuration, their numeric values are known to be equal. For example, the following are all considered to be equal:

- $012 = 12.00 = 12$

### Alphanumeric comparison

Alphanumeric fields/literals are compared based on the size and content of the field.

- The size of the field/literal is determined by the size of the right-hand value of the IF statement. The right-hand value is the comparison field or literal after the relational condition.
- After the size of the comparison is determined, the content is compared from left to right.

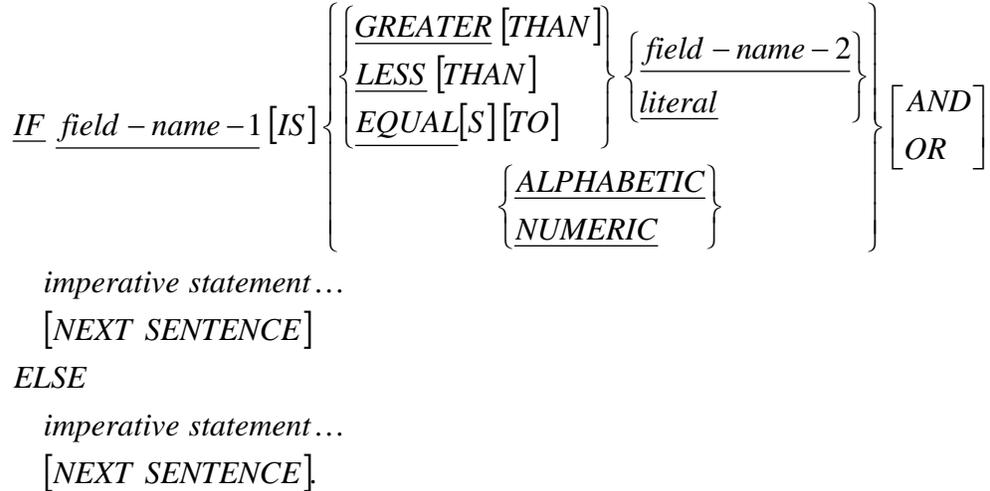
### Class test

In addition to comparing field-to-field and field-to-literal, the type of data in a field can be tested. A class test is a specialized test that can be performed with the IF statement. The class test options allow you to determine the contents of a field by class, either alphabetic or numeric. The CSL words for use with a class test are:

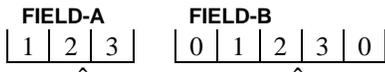
- ALPHABETIC
- NUMERIC

# Comparing Data with Compound Conditionals

**Format:**



## OR Comparison



### Example 1:

```

IF FIELD-A GREATER THAN FIELD-B OR
  FIELD-A EQUALS FIELD-B
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
  
```

### Output:

TRUE



### Example 2:

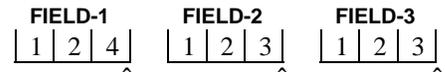
```

IF FIELD-O EQUAL TO 'AA' OR 'BB'
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
  
```

### Output:

TRUE

## AND Comparison



### Example 3:

```

IF FIELD-1 GREATER THAN FIELD-2 AND
  FIELD-2 IS EQUAL TO FIELD-3
  PRINT 'TRUE'
ELSE
  PRINT 'FALSE'.
  
```

### Output:

TRUE

---

## NOTES

---

## Comparing data, continued

### Compound conditional

The conditional statements discussed so far have compared one condition per statement. Compound conditional statements enable a statement to compare many conditions. The two operations that designate a compound condition are:

- **OR**  
tests to see if any one of several conditions exists.
- **AND**  
a test to see if all conditions exist.

### OR in a compound conditional

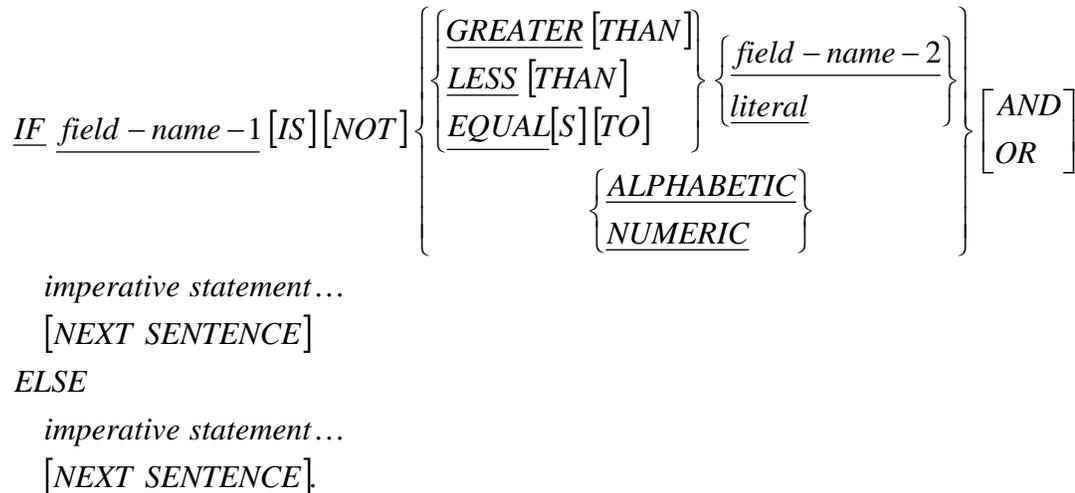
To perform an operation or a series of operations if any one of several conditions exists, the compound conditional with conditions separated by **OR** may be used. This means that if any one of several conditions exists, the imperative statement(s) specified will be executed. The **ELSE** option will be performed if all of the stated conditions are not met.

### AND in a compound conditional

To perform an operation or a series of operations where all of several conditions must be met, the compound conditional with conditions separated by **AND** may be used. This means that if all of several conditions exist, the imperative statement(s) specified will be executed. The **ELSE** option will be performed if any one of the stated conditions is not met.

# Negative Conditionals

**Format:**



**Incorrect Example:**

Print NORTH when FIELDA = 7 or 8, otherwise print SOUTH

```

IF FIELDA IS NOT EQUAL TO '7' OR '8'
  PRINT 'SOUTH'
ELSE
  PRINT 'NORTH'.
  
```

```

FIELDA = 6  Output = SOUTH
FIELDA = 7  Output = SOUTH
  
```

**Correct Example:**

Print NORTH when FIELDA = 7 or 8, otherwise print SOUTH

```

IF FIELDA IS NOT EQUAL TO '7' AND '8'
  PRINT 'SOUTH'
ELSE
  PRINT 'NORTH'.
  
```

```

FIELDA = 6  Output = SOUTH
FIELDA = 7  Output = NORTH
  
```

---

## NOTES

---

## Comparing data, continued

### Negative conditions

There are times when you might want to execute a series of steps only if a certain condition does not exist.

- Use a NEXT SENTENCE to avoid performing any operation if the condition exists and executes only statements after the ELSE.
- Use a NOT preceding the conditional relation to cause the negative effect of the conditional.

### Negative compound conditionals

A common error may occur when you negate compound conditionals. When negating compound conditionals separated by AND or OR, remember that only one condition must be true for the combined condition to be true for an OR statement, and both conditions must be true for an AND statement.

### Example

The routine shown on the opposite page is to print NORTH if FIELD A is not equal to 7 or 8; otherwise it should print SOUTH. An evaluation of the statement on the left produces:

if A is 6; SOUTH will be printed, correct.

if A is 7; SOUTH will be printed, incorrect.

When A=7, the first condition A IS NOT EQUAL TO '7' is FALSE since A is equal to 7. However, the second condition A IS NOT EQUAL TO '8' is TRUE since A is not equal to 8. Therefore, SOUTH is printed since any one of several conditions must exist for the OR condition to be TRUE. In fact, you can now see that SOUTH will be executed regardless of the contents of A. The correct statement is demonstrated on the left.

# IF/ELSE Summary

## Relations

- EQUALS
- [IS] EQUAL[S] [TO]
- [IS] GREATER [THAN]
- [IS] LESS [THAN]

## Class test

- ALPHABETIC
- NUMERIC

## Conditionals

- If the condition exists, all imperative statements to the period or the ELSE will be executed.
- If the condition does not exist, all imperative statements after the ELSE are executed or processing continues to the next sentence.
- Numeric
  - Numeric fields/literals are compared algebraically.
- Alphanumeric
  - Alphanumeric fields/literals are compared based on the size and content of the field.

## Compound conditionals

- AND - When all of several conditions exists, the imperative statement(s) specified will be executed. The ELSE option will be performed if any one of the stated conditions is not met.
- OR - When any one of several conditions exists, the imperative statement(s) specified will be executed. The ELSE option will be performed if all of the stated conditions are not met.

## Negative Conditions

- Any test can be preceded with a NOT to test the negative of a conditional.

---

## NOTES

---

## Comparing data, continued

### **IF/ELSE summary**

In summary, there are several important rules to consider when you use the IF/ELSE statements to compare data.

- The preceding outline summarizes the IF/ELSE rules.

# Calculating

**Format:**

$$\underline{\text{CALCULATE}} \left\{ \begin{array}{l} \textit{literal} \\ \textit{field - name - 1} \end{array} \right\} \left\{ \begin{array}{l} + \\ - \\ * \\ / \end{array} \right\} \left\{ \begin{array}{l} \textit{literal} \\ \textit{field - name - 2} \end{array} \right\} = \textit{field - name - 3}.$$

**Numeric Calculation Example:**

CALCULATE ANNUAL-SALARY \* :1.10 + :100.00 = ANNUAL-SALARY.

**Date Calculation Example:**

CALCULATE CURRENT-DATE-CYYMDD - BIRTH-DATE = AGE.

---

## NOTES

---

## Calculating

### **CALCULATE**

The CALCULATE verb lets you perform arithmetic operations. The arithmetic operations that can be performed include:

- Addition
- Subtraction
- Multiplication
- Division

### **Symbols**

The Symbols that are used to determine the arithmetic operation are:

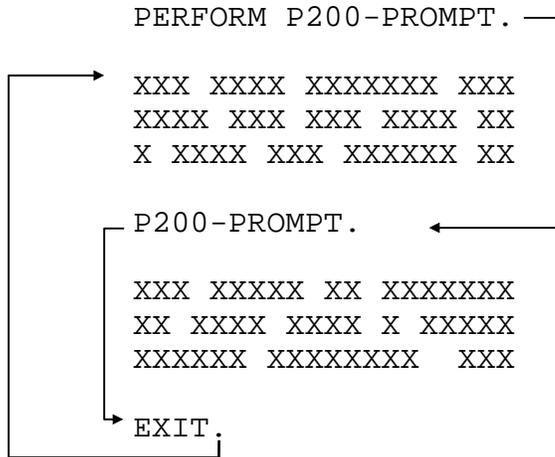
- + Add.
- - Subtract the field that follows the minus sign (-) from the field that precedes it.
- \* Multiply the field that precedes the asterisk (\*) by the field that follows it.
- / Divide the field that precedes the slash (/) by the field that follows it.
- = End the calculation. The value after the equal sign (=) receives the result of the calculation.

### **Usage**

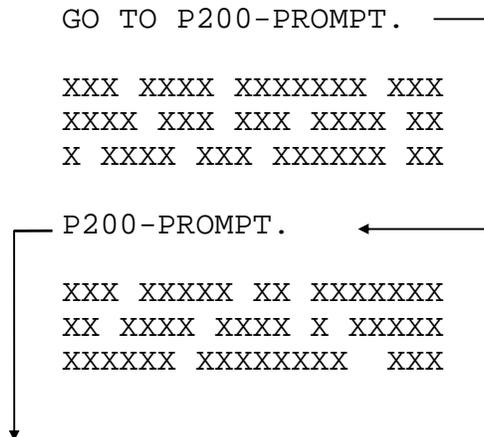
- Fields are processed in order, left to right.
- Only simple operations are allowed. Do not use parentheses or try to perform complex operations with the CALCULATE verb.
- Separate calculation symbols with spaces.
- Use a colon (:) to identify a numeric literal value.

# Transferring Control Within a Program

## PERFORM



## GO TO



---

## NOTES

---

## Transferring control within a program

### Transfer control

There are times when you might want to execute a series of program statements that do not immediately follow the current statement. To accomplish this you must transfer control to a label within the program and re-start execution from that point. There are two verbs in CSL for transferring control within a program:

- **PERFORM**
- **GO TO**

Both of these verbs require a paragraph label to determine where in the current program to be redirected.

### **PERFORM**

The **PERFORM** verb transfers control to a designated paragraph label and then returns control to the originating statement upon completion.

### **GO TO**

The **GO TO** verb transfers control completely to a designated paragraph.

### **Paragraph labels**

Paragraph labels are used to locate a group of statements in a program.

Paragraph labels have a required format 'P999-.' where:

- **P** = literal  
tells the system a paragraph label follows.
- **999**  
a unique three digit number between 100 and 999.
- **- (hyphen)** = literal  
a required part of the paragraph label.
- **.** (period) = literal  
delimits the label.

Between the hyphen and the period, a description may be inserted. (No embedded spaces allowed.)

# PERFORM

**Format:**            *PERFORM Pnnn[label].*

**Single PERFORM Example:**

```
PERFORM P990-CALCULATE.  
:  
:  
P990-CALCULATE.  
  MOVE :125.00 TO AMOUNT-ONE.  
  CALCULATE ANNUAL-SALARY * 1.10 = ANNUAL-SALARY.  
EXIT.
```

**Nested PERFORM Example:**

```
P100-FIRST.  
  PERFORM P200-SECOND.  
  :  
  :  
P200-SECOND.  
  :  
  PERFORM P300-THIRD.  
  :  
EXIT.  
P300-THIRD.  
  :  
  PERFORM P400-FOURTH.  
  :  
EXIT.  
P400-FOURTH.  
  :  
  :  
EXIT.
```

---

## NOTES

---

## Transferring control within a program, continued

### PERFORM/EXIT

The PERFORM verb causes execution of the series of instructions at the named paragraph. When an EXIT is encountered, control is transferred to the statement directly following the PERFORM.

- PERFORM requires a valid paragraph label.
- EXIT is the only delimiter for the PERFORM verb and must be placed at the end of the logic (or paragraph) to be performed.
- You can nest PERFORM statements up to six levels deep.

# GO TO

**Format:**     *GO [UP] TO Pnnn[label].*

**GO TO Example:**

```
GO TO P990-PROMPT.  
  :  
  :  
P990-PROMPT.  
  PRINT 'ENTER: FF-SCR FOR EMPLOYEE NAME & ADDRESS'.  
  SPACE-OVER :07.  
  PRINT 'HH-SCR FOR HOURS, EARNINGS, AND DEDUCTIONS'.
```

---

## NOTES

---

## Transferring control within a program, continued

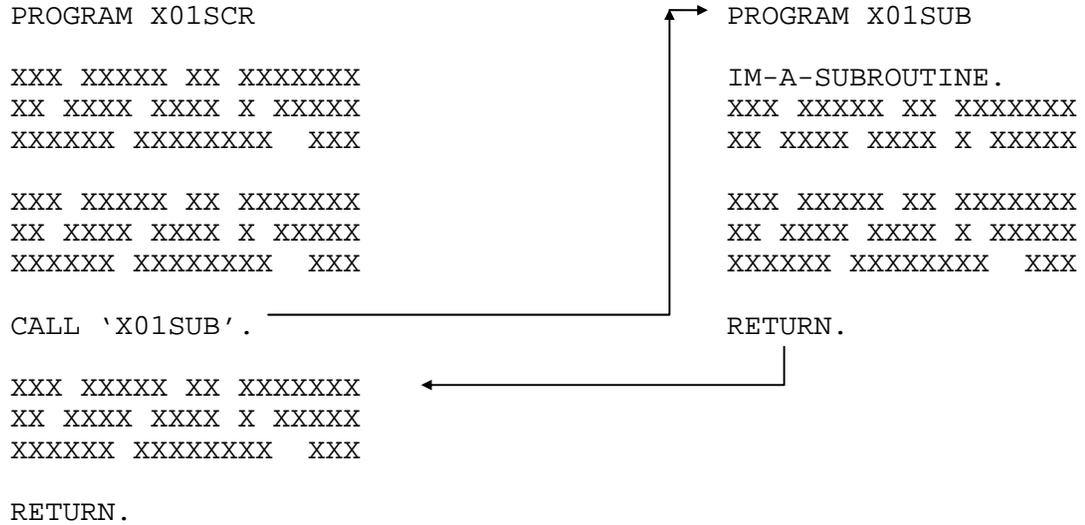
### GO TO

The GO TO verb transfers control to another paragraph in your program. This verb provides an alternative to normal sequential processing where each statement is executed in the order that it appears.

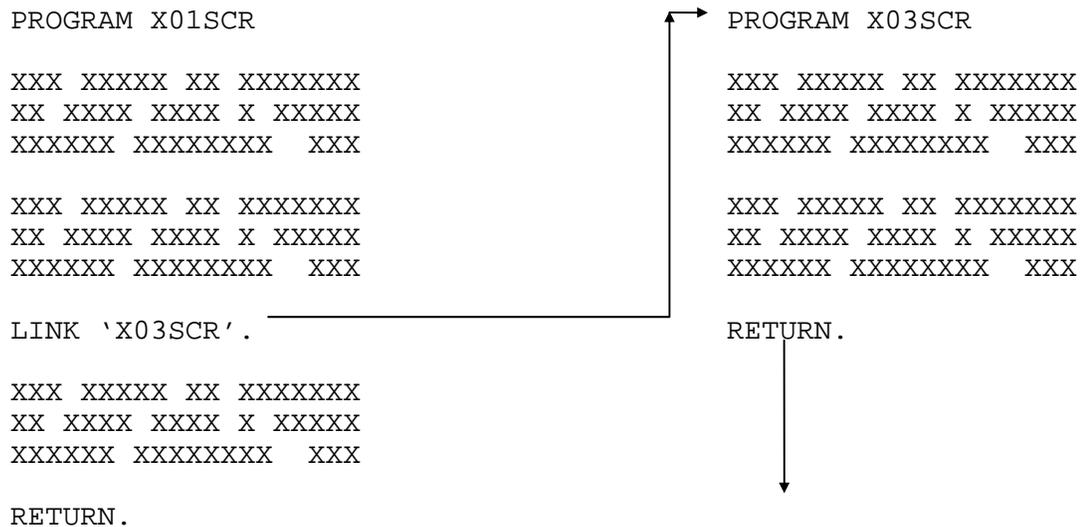
- GO TO is one of the few multi-word verbs that does not use a hyphen. However, TO is required.
- The specified paragraph label can occur before or after the GO TO statement. Control passes to the first matching label that physically follows the GO TO statement. If none exists after the statement, control passes to the first matching paragraph label found by going back through the program from the GO TO statement.
- You can use the statement GO UP TO to avoid any confusion caused by duplicate paragraph numbers. GO UP TO only searches back through the program.
- Processing continues with the statement that follows the paragraph label specified in the GO TO statement.

# Transferring Control to Other Programs

## CALL



## LINK / CHAIN



---

## NOTES

---

## Transferring control within a program, continued

### Transfer control

There are times when you might want to execute a series of program statements that are not in the current program. To accomplish this you must transfer control to another program and re-start execution from that point. There are three verbs in CSL for transferring control to other programs:

- CALL/RETURN
- LINK
- CHAIN

### CALL/RETURN

The CALL verb transfers control to a designated program and then returns control to the originating program at the statement following the CALL. The RETURN verb is used to signal the system when to return to the CALLing program.

### LINK

The LINK verb transfers control completely to a designated program.

### CHAIN

The CHAIN verb transfers control completely to a designated program and changes the content of the command line program field to the program that has been chained.

### IM-A-SUBROUTINE

The IM-A-SUBROUTINE verb may be designated as the first verb of a program to inhibit direct access to a program that is CALLED or LINKed.

# Transferring Control

## CALL

**Format:**     *CALL [literal]*

**Format:**     *IM-A-SUBROUTINE.*

**Example 1:**

```
CALL 'XSUB1 ' .
```

**Example 2:**

```
MOVE 'XSUB1 ' TO W7-06-012.  
CALL .
```

**Example:**

```
IM-A-SUBROUTINE .  
  :  
  :  
RETURN .
```

## Advantages of CALLing Subroutines

**Avoids duplication of effort**

- When specific routines need to be included in more than one program, it is best to write them separately and call them in each program as needed.

**Improves programmer productivity**

- Programmers with advanced skills will typically be responsible for coding complex subroutines that need to be written as efficiently as possible. In an era of programmer shortage, this technique can result in more efficient use of a programmer's time.

---

## NOTES

---

## Transferring control within a program, continued

### CALL

The CALL verb lets you transfer control to another program temporarily. The program that contains the CALL verb is referred to as the ‘calling program’, and the program that receives temporary control is referred to as the ‘called program’.

- A six-character literal name of the called program is required. Use blank spaces to pad out the program name if it is less than six characters long.
- Control returns to the calling program when the called program finds the RETURN verb. Processing continues with the statement after the CALL.
- You can specify multiple CALL statements in a program, but you can nest only six CALLs. However, keep in mind that every program already contains one implicit CALL: made by CYB90–, the root program that controls all processing in the system.
- Remember that RETURN is assumed at the end of all programs, whether explicitly coded or not. To ensure that control is returned to the system, and not to the calling program, you may specify RETURN–Q at the end of the linked program.

*Note:* You may want to specify the IM–A–SUBROUTINE verb as the first statement in a called program to designate that this program cannot be executed independently.

*There is an assumed RETURN as the last statement of all programs.*

# Transfer Control CALL

## FMLEDT Subroutine Example:

```
MOVE EMPLOYEE-NAME TO LAST-FIRST.  
CALL 'FMLEDT'.  
MOVE PRINT-FIELD TO FIRST-LAST.  
PRINT FIRST-LAST.
```

Before: AUSTIN, STEVEN

After: STEVEN AUSTIN

## INCAPS Subroutine Example:

```
MOVE EMPLOYEE-NAME TO W8-30-800.  
CALL 'INCAPS'.  
PRINT W8-30-800.
```

Before: AUSTIN, STEVEN

After: Austin, Steven

## GDTOJD Subroutine Example

```
MOVE BIRTH-DATE TO CENTURY-SAVE-DATE.  
MOVE SAVE-DATE TO GREGORIAN-DATE-1.  
CALL 'GDTOJD'.  
PRINT JULIAN-DAYS-1.
```

---

## NOTES

---

## Transferring Control Within a Program, continued

### Subroutines

Three delivered subroutines can be used to manipulate data:

- **FMLEDT**  
a subroutine that converts a name field to a first–middle–last name format.
- **GDTOJD**  
a subroutine that converts a Gregorian-style date (CCYYMMDD for the US and Canada, CCYYDDMM elsewhere) to a Julian style date. Julian style dates follow this general format: DDD, where DDD is a three-digit value from 001 to 366.
- **INCAPS**  
a subroutine that converts a 30-character string into initial capital format.

# Transferring Control LINK

**Format:**     LINK [*literal*].

**LINK Example 1:**

```
IF CODE-1 EQUALS 'I'  
  LINK 'HHISCR'.
```

**LINK Example 2:**

```
IF CODE-1 EQUALS 'I'  
MOVE 'HHISCR' TO W7-06-012  
LINK.
```

---

## NOTES

---

## Transferring control within a program, continued

### LINK

The LINK verb transfers control to another program permanently. Control is returned to the system when the linked program executes the RETURN verb, whether explicit or assumed.

- A literal value of the six-character program that you want to link is required. Enclose the name of the linked program in single quotation marks ( ' ') and pad the program name with blank spaces if it is less than six characters long.
- If the LINK verb is in a called program, control is not returned to the calling program. Processing continues at the next logical statement after the previous CALL verb.

*Note:* You may want to specify an *IM-A-SUBROUTINE* verb as the first statement in a called program to designate that this program cannot be executed independently.

# Transferring Control CHAIN

**Format:**     CHAIN [*literal*].

**CHAIN Example 1:**

```
CHAIN 'FF-SCR'.
```

---

**NOTES**

---

## Transferring control within a program, continued

### CHAIN

The CHAIN verb transfers control to another program permanently just like the LINK verb. However, CHAIN replaces the PROGRAM field of the command line with the program that is being chained so further online execution will start at that program.

- The CHAIN verb must be ‘hard coded’ into each program. As an alternative, *Solution Series 4.x* users can create checklists. *Solution Series 3.x* users can create P/C records using EDIT and execute them using the SERIES program.
- Checklists or SERIES allows each user to create a unique set of forms to be ‘chained’ together and is not restricted to a ‘hard coded’ condition.
- Because CHAIN puts the name of the CALLED program in the PROGRAM-FIELD of the command line, IM-A-SUBROUTINE cannot be used in programs that receive control via CHAIN.

*Note:* For further information on creating checklists, refer to the *Using The Solution Series: Administrative Solutions* documentation.

## Section Summary

- **Moving data**
- **Comparing data**
- **Calculating**

---

NOTES

---

## Section summary

In this section, you learned additional CSL verbs and techniques. Specifically you learned:

### Moving data

---

---

---

---

---

---

---

### Comparing data

---

---

---

---

---

---

---

### Calculating

---

---

---

---

---

---

---

## Section Summary, continued

- **Transferring control within a program**
- **Transferring control to other programs**

---

### NOTES

---

**Section summary, continued**

**Transferring control within a program**

---

---

---

---

---

---

---

**Transferring control to other programs**

---

---

---

---

---

---

---

# Section 5 Exercise

---

**NOTES**

---

## Section 5 exercise

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in the section. Answer the questions or perform the task below.

Use the field definitions to supply the results to the program statements.

Field Name	Data Type	Storage Length	Content
FIELD-1	Alphanumeric	4	CSSS
FIELD-2	Alphanumeric	3	ABC
FIELD-3	Numeric 4 Decimals	6	050000
FIELD-4	Numeric 2 Decimals	6	400000
FIELD-5	Numeric 2 Decimals	6	025000
FIELD-6	Regular Date	6	900101

1. Display the results for FIELD-5

```
MOVE :0.00 TO FIELD-5.
CALCULATE FIELD-3 * FIELD-4 = FIELD-5.
```

FIELD-5 = \_\_\_\_\_

2. Display the results for FIELD-2

```
MOVE FIELD-1 TO FIELD-2.
```

FIELD-2 = \_\_\_\_\_

3. Display the results for FIELD-3

```
IF FIELD-1 EQUALS 'A' OR 'B' OR 'C' MOVE :10 TO
FIELD-3
ELSE MOVE :20 TO FIELD-3.
```

FIELD-3 = \_\_\_\_\_

## Section 5 Exercise, continued

---

**NOTES**

---

**Section 5 exercise, continued**

4. Create a new program (subroutine) to calculate an employee's age.
  - Add coding to calculate the employee's age using the following formula:  
 $\text{CURRENT-DATE-CYYMDD} - \text{BIRTH-DATE} = \text{AGE}$ .

*Note:* *AGE is not a field that is stored in the employee's record.*

- Add coding to not allow direct access to this program (subroutine).
5. Modify the program you created in Section 3: Cyborg Scripting Language Basics as follows:
    - Add coding to transfer control (CALL) to the age calculation subroutine.
    - Display the result of the calculation.

---

**NOTES**

---

## Section 6: Work Areas and Pointers

---

### Table of Contents

Introduction.....	211
Work areas .....	213
Pointer 44.....	225
Pointer 45.....	227
Pointer manipulation.....	229
Section summary.....	241
Section 6 exercise .....	243

## Objectives

- **Identify and use the available work areas**
- **Identify the pointers/segment relationship for company, tax, and employee records**
- **Use CSL verbs to manipulate pointers**

---

### NOTES

---

## Introduction

### **Purpose**

In this section you will learn the techniques for working with pointers related to work areas, company, tax, and employee records.

### **Objectives**

Upon completion of this section you will be able to:

- Identify and use the available work areas.
- Identify the pointers/segment relationship for company, tax, and employee records.
- Use CSL verbs to manipulate pointers.

## Work Areas

- **Pointer 6**
- **Pointer 7**
- **Pointer 8**
- **Pointer 44**

---

NOTES

---

## Work areas

### Programming with work areas

Several work areas can be used in CSL Programming. These work areas provide a scratch pad where temporary and permanent data can be stored and manipulated. The available work areas are:

- Pointer 6
- Pointer 7
- Pointer 8
- Pointer 44

#### Pointer 6

Pointer 6 contains temporary and permanent numeric fields that can be used to perform arithmetic operations.

#### Pointer 7

Pointer 7 contains constant data and stored information used by the system and programs.

#### Pointer 8

Pointer 8 is the I/O area for forms, reports, and most files.

#### Pointer 44

Pointer 44 addresses ten 1,000-character work areas that may be saved on FILE02 for the duration of the current session.

# Work Areas

**Format:**

W<sub>n</sub>-99-000-TAG

**W** Literal of W for Work.

**n** Pointer number (can be 0, 6, 7 or 8).

**99** Field length.

**000** Field displacement within the pointer, relative to zero.

**TAG** Optional comment tag that is ignored by the system.

**Example:**

```
MOVE BIRTH-DATE TO CENTURY-SAVE-DATE.
PRINT 'Birth Month: ' W8-02-212-BIRTH-MONTH.
```

**Output:**

Birth Month: 03

**Pointer 8:**

		1	9	6	2	0	3	2	5			
0		2	2	2	2	2	2	2	2	2		
0		0	0	1	1	1	1	1	1	1		
0	...	8	9	0	1	2	3	4	5	6		

---

## NOTES

---

## Work areas, continued

### Work area access

Work areas have several predefined fields that may be used to access or manipulate data. However, you may access the work areas in another format. Work areas may be accessed using the following format:

- Wn-99-000-TAG
  - W—Literal of W for Work.
  - n—Pointer number (can be 0, 6, 7 or 8).
  - 99—Field length.
  - 000—Field displacement within the pointer, relative to zero.
  - TAG—Optional comment tag which is ignored by the system.

## Pointer 6—Numeric Fields

### Format

TEMP-01-V0

**TEMP** Literal (can be TEMP or PERM).

**01** Field number (01-06 for TEMP, 01-11 for PERM).

**V0** Number of decimal places (V0 and V2 for TEMP, V0 through V6 for PERM).

### TEMP Example:

```
MOVE :0 TO TEMP-01-V2.  
IF ANNUAL-SALARY IS LESS THAN :30000.00  
    CALCULATE ANNUAL-SALARY * :1.10 = TEMP-01-V2  
ELSE  
    CALCULATE ANNUAL-SALARY * :1.05 = TEMP-01-V2.  
PRINT 'NEW ANNUAL SALARY: ' TEMP-01-V2.
```

### PERM Example:

```
MOVE :0.0000 TO PERM-01-V4.  
IF ANNUAL-SALARY IS LESS THAN :30000.00  
    MOVE :1.1000 TO PERM-01-V4  
ELSE  
    MOVE :1.0500 TO PERM-01-V4.  
CALL 'XCALC1'.  
PRINT 'NEW ANNUAL SALARY: ' PERM-02-V2.
```

---

## NOTES

---

## Work areas, continued

### Pointer 6 numeric fields

Pointer 6 contains two types of numeric fields that may be used in your CSL programs for arithmetic calculations. The fields are:

- TEMP
- PERM

### TEMP

TEMP fields are temporary, meaning they are valid only for your program since the data in the TEMP field is not saved between transmissions.

### PERM

PERM fields are permanent, meaning they are valid across programs since the data in the PERM field is saved between transmissions. PERM fields must be used to pass information from one program to another using the CALL, LINK, or CHAIN verbs.

### Other Pointer 6 areas

In addition to the TEMP and PERM fields, Pointer 6 may be used to store data between executions of a program. Displacements 072–191 are reserved for this purpose.

## Pointer 7—Constants

### **SPACES Incorrect Example:**

```
IF CONTROL-3-CODE EQUALS SPACES  
  MOVE '0000' TO CONTROL-3-CODE.
```

### **QUOTE Incorrect Example:**

```
PRINT 'WHAT'S YOUR NAME?'
```

### **SPACES Correct Example:**

```
IF SPACES EQUALS CONTROL-3-CODE  
  MOVE '0000' TO CONTROL-3-CODE.
```

### **QUOTE Correct Example:**

```
PRINT 'WHAT' QUOTE'S YOUR NAME?
```

---

## NOTES

---

**Work areas, continued****Pointer 7**

Pointer 7 contains constant values and pre-designated fields. Each Pointer 7 field can be accessed by its field name or by the 'a.k.a' (also known as) reference in the W7-99-000 format.

- Documentation for each Pointer 7 field can be found in Appendix C: Pointer 7 Fields.

**Constants**

Constants are fields within Pointer 7 that contain a constant value. Some constants you might want to use are:

- **SPACE or SPACES** (a.k.a. W7-01-176)  
A one position field containing spaces. This field occurs up to a maximum of 60 spaces.
- **HIGH-VALUES** (a.k.a. W7-01-238)  
A one position field containing the highest value for your computer.
- **LOW-VALUES** (a.k.a. W7-01-034)  
A one position field containing the lowest value for your computer.
- **QUOTE** (a.k.a. W7-01-005)  
A one position field containing a single quote.

## Pointer 7—Command Line

### Current Date Example:

```
MOVE BIRTH-DATE TO CENTURY-SAVE-DATE.  
IF W7-02-118-CURR-MONTH EQUALS SAVE-MONTH  
AND W7-02-120-CURR-DAY EQUALS SAVE-DAY  
PRINT 'HAPPY BIRTHDAY ' EMPLOYEE-NAME.
```

### CODE-1 Field Example:

```
IF CODE-1 EQUALS 'I'  
LINK 'XXISCR'.
```

### KEY-FIELD Example:

```
IF KEY-FIELD NOT EQUAL SOCIAL-SECURITY-NBR AND  
W7-01-257 EQUALS '-' AND  
W7-01-260 EQUALS '-'  
MOVE KEY-FIELD TO SOCIAL-SECURITY-NBR.
```

### ADDITIONAL-KEY field example:

```
MOVE W7-06-264 TO SAVE-DATE.  
MOVE SAVE-DATE TO HOLD-DATE.  
PERFORM P300-FIND-SALARY.
```

---

## NOTES

---

**Work areas, continued****Pointer 7 command line**

The command line entries are stored in Pointer 7. You can check the contents of the command line using the field name or check portions of a command line field by manipulating the length and displacement of the work area, a.k.a designation. The command line fields are:

- SESSION-ID (W7-04-304)—Session number value.
- COMM-CANCEL (W7-01-302)—Action value.
- COMPANY-NUMBER (W7-06-240)—Control 1-2 value.
- PROGRAM-FIELD (W7-06-246)—Program field value.
- CODE-1 (W7-01-252)—Code1 field value.
- CODE-2 (W7-01-253)—Code2 field value.
- KEY-FIELD (W7-10-254)—Key field value.
- ADDITIONAL-KEY (W7-15-264)—Additional Key field value.
- CURRENT-TIME (W7-06-122)—System time value.
- CURRENT-DATE (W7-06-116)—System date value.
- OPERATOR-ID (W7-04-280)—Operator identification value.

## Pointer 8—I/O and Work Area

**Example:**

```
MOVE HOME-PHONE TO W8-07-800.  
MOVE SPACES TO W8-14-700.  
MOVE '(' TO W8-01-700.  
MOVE HOME-AREA-CODE TO W8-03-701.  
MOVE ')' ' TO W8-02-704.  
MOVE W8-03-800 TO W8-03-706.  
MOVE '-' TO W8-01-709.  
MOVE W8-04-803 TO W8-04-710.  
PRINT 'HOME TELEPHONE NUMBER: ' W8-14-700.
```

**Output:**

```
HOME TELEPHONE NUMBER: (312) 279-7000
```

---

### NOTES

---

**Work areas, continued****Pointer 8**

Pointer 8 is used for form, file, report and program input/output. It can also be used as a work area to manipulate data.

- Displacements 650 to a maximum of 999 may be used as temporary storage of data fields.

*Warning: Since Pointer 8 is used by various input/output functions you must be careful when coding to avoid an I/O operation from overlaying any temporary field you may be using in Pointer 8. Some of the verbs that use Pointer 8 include:*

- *READ-COMPANY*
- *READ-TAXES*
- *READ-EMPLOYEE*
- *CALL*
- *LINK*
- *CHAIN*

**Pointer 12 PRINT-FIELD**

The Pointer 8 work area is redefined as a pointer area called PRINT-FIELD addressed using Pointer 12. You may position Pointer 12 to specific displacements using the verb SET PRINT-FIELD TO:nnn. All subsequent moves in or out of PRINT-FIELD automatically reposition Pointer 12.

## Pointer 44

**Format:**

SET SCRATCH-AREA TO :n.      CLEAR-SCRATCH. RESTORE-SCRATCH.

SAVE-SCRATCH.

**Example:**

```
SET SCRATCH-AREA TO :5.  
CLEAR-SCRATCH.  
MOVE FLD-A TO W0-60-450.  
MOVE FLD-B TO W0-04-202.  
:  
:  
SAVE-SCRATCH.  
:  
:  
SET SCRATCH-AREA TO :5.  
RESTORE-SCRATCH.
```

---

### NOTES

---

## Pointer 44

Pointer 44 addresses a 10,000-character work area as ten occurrences of 1,000-characters.

### **POINTER 44**

A particular 1,000-character work area (1-10) is selected using the SET SCRATCH-AREA TO :n verb, where n is a value of 1 through 10. Once selected, data may be addressed in the work area by Field Name Table field names for pointer 44 or by using the length/displacement (a.k.a.) naming convention of W0-ll-ddd.

Three verbs exist to clear, save, and restore the selected work area:

### **CLEAR-SCRATCH**

Clears the 1,000-character work area to spaces.

### **SAVE-SCRATCH**

Saves the current 1,000-character work area to FILE02 as a 'ZN' record. Only one copy of each 1000 character work area may be saved per session and program.

### **RESTORE-SCRATCH**

Restores the FILE02 version of the work area. If no 'ZN' record exists, the work area is cleared to spaces.

*Note:*

*Use the SET SCRATCH-AREA TO :n to indicate which occurrence is to be cleared, saved, or restored. All 'ZN' records for a session are deleted when you exit The Solution Series.*

## Pointer 45—SCRATCH

**Format:**

*SET SCRATCH TO :nnnn*

**Example:**

```
SET SCRATCH TO :50.  
MOVE EMPLOYEE-NAME TO SCRATCH.  
MOVE ADDRESS TO SCRATCH.  
MOVE ADDRESS-2 TO SCRATCH.  
MOVE CITY/STATE TO SCRATCH.  
MOVE ZIP-CODE TO SCRATCH.
```

---

### NOTES

---

## Pointer 45

### POINTER 45 SCRATCH

The entire 10,000-character work area is known as a special field, SCRATCH, addressed by pointer 45. You may position pointer 45 to specific displacements from 0 to 9999 using the verb SET SCRATCH TO :nnnn. All subsequent moves in or out of the area using the SCRATCH field name automatically advance your position.

*Note:* When using the MOVE instruction, Pointer 44 and 45 fields follow the same rules as all other data originating in FILE02.

# Segment/Pointer Relationship

## Record/Pointer/Segment

Record	Ptr	Seg	Description
-----			
Company	21	A	Company Name & Address
	22	B	Company Earning & Deductions (HEDs)
	23	C	Other Company detail/user-defined data
	24	D	Report Generator Selection
Tax	25	T1	Tax Specification Information
	26	4	Tax Exemptions and Credits
	27	5	Tax Brackets
Employee	28	Key	Employee key information
	29	E	Basic employee data
	30	F	Employee name and address
	31	G	Labor detail
	32	H	Earnings and Deductions (HEDs)
	33	H	Earnings and Deductions history
	34	J	Employee taxes
	35	J	Employee tax history
	36	L	Human Resources data/user-defined data
	37	P	Period Table data

## Area 2 - Employee Record

		29 ↓		
28→	999999M12345678999		EE 308-82-3775...	
30→	F001AUSTIN, STEVE...		F999AUSTIN, STEVE...	
31→	G0105000CHGOMANU...		G020500NYC SALE...	
32/33→	H0010100...	H0030100...	H5010153...	
34/35→	J101 01...	J102 10...	J103 10...	
36→	LO100204AUSTIN, J...		LO2002ACME MANUF...	
	LO3CHICAGO, IL 606...		LZF213A319I1500...	
	LZF214A319S0700...		LZQ213A31SASMITH...	
	LZR218G029C1600...		LZR218A319C1500...	
37→	P91010...	P91020...	P91013...	P91040...

---

## NOTES

---

## Pointer manipulation

### Pointers

As you may recall from our earlier discussion in Section 2: The Solution Series Architecture, a pointer contains the address of where information is located within working storage.

- When an employee, company, or tax record is read into working storage, each segment is assigned to a specific pointer.
- Pointers are initially set to point to the first occurrence of each segment.
- In order to work with data that resides within a multiple-occurrence segment, you must manipulate the pointer value to point to the appropriate occurrence. There are many techniques to manipulate a pointer value. Our discussion, however, is limited to 2 techniques which will use the following verbs:
  - FIND
  - PROCESS/END-PROCESS

# FIND

**Format:**

$$\underline{FIND} \underline{non-key-field-1} \left[ \begin{array}{l} FROM\ HERE \\ STARTING\ WITH \left\{ \begin{array}{l} literal \\ field-name-2 \end{array} \right\} \end{array} \right].$$

**FIND the First Occurrence Example:**

```
FIND CONTROL-3-CODE.  
IF FOUND  
    PRINT 'CONTROL 3: ' CONTROL-3-CODE  
        ' ' CONTROL-3  
ELSE  
    PRINT '*** NO CONTROL-3 FOUND ***'.
```

**FIND a Specific Occurrence Example:**

```
FIND HED-AMOUNT-YTD STARTING WITH '500'.  
IF FOUND  
    PRINT 'DEDUCTION NBR: ' HED-NUMBER NEXT-LINE  
    PRINT 'DEDUCTION YTD: ' HED-AMOUNT-YTD  
    'DEDUCTION QTD: ' HED-AMOUNT-QTD ' '  
    'DEDUCTION MTD: ' HED-AMOUNT-YTD  
ELSE  
    PRINT '*** NO DEDUCTION HED 500'.
```

---

## NOTES

---

## Pointer manipulation, continued

### **FIND**

The FIND verb enables you to locate a specific occurrence of a segment.

- The FIND verb must be immediately followed by a non-key field from the segment you wish to locate.
- Follow a FIND statement with an IF FOUND or IF NOT-FOUND statement to test the result of the FIND operation.
- Unless you use one of these valid qualifiers, FIND begins at the first segment in the pointer:

### **FROM HERE**

Begins the FIND operation where the pointer is currently positioned. Be sure that you know the position if you use this qualifier.

### **STARTING WITH**

The FIND operation will position the pointer at the first occurrence that contains an equal or greater value in its key of the specified key value.

## MATCH- Verbs

### FIND an Occurrence AS OF a Starting Key Using MATCH-SEGMENT-TYPE:

```
FIND HED-AMOUNT-YTD STARTING WITH '501'.
MATCH-SEGMENT-TYPE.
IF FOUND
  PRINT 'DEDUCTION NBR: ' HED-NUMBER NEXT-LINE
  PRINT 'DEDUCTION YTD: ' HED-AMOUNT-YTD ' '
  PRINT 'DEDUCTION QTD: ' HED-AMOUNT-QTD ' '
  PRINT 'DEDUCTION MTD: ' HED-AMOUNT-MTD.
```

### FIND an Occurrence AS OF The Current Date Using MATCH-SEGMENT-CODE:

```
FIND JOB-CODE STARTING WITH CURRENT-DATE-CYYMDD.
MATCH-SEGMENT-CODE.
IF FOUND
  PRINT 'Current Job Code:' JOB-CODE JOB-CODE-EXTENT.
```

---

## NOTES

---

## Pointer manipulation, continued

### **MATCH–**

MATCH– verbs help you to find the most recent occurrence of a stacked segment based on a limited key. Most multiple (stacked) segments have several key fields, including a date.

Because the FIND verb will only have a FOUND condition if an exact match is made with the STARTING WITH key, we are not able to FIND an ‘AS OF’ occurrence. To accommodate this ‘AS OF’ situation, the MATCH– verbs can be used to change the comparison key being used to determine whether the segment occurrence is FOUND.

The MATCH– statement must immediately precede the IF FOUND or IF NOT FOUND statement. Two of the most commonly used MATCH– verbs are:

- **MATCH–SEGMENT–TYPE**  
The IF FOUND is limited to a match with the segment type.
- **MATCH–SEGMENT–CODE**  
The IF FOUND is limited to a match with the segment type and segment code.

## FIND– Verbs

### FIND an Occurrence As of the Current Date Using a FIND– verb:

#### Example 1:

```
FIND-DEDUCTION.  
IF FOUND  
    PRINT 'DEDUCTION NBR: ' HED-NUMBER NEXT-LINE  
    PRINT 'DEDUCTION YTD: ' HED-AMOUNT-YTD ' '  
    PRINT 'DEDUCTION QTD: ' HED-AMOUNT-QTD ' '  
    PRINT 'DEDUCTION MTD: ' HED-AMOUNT-MTD.
```

#### Example 2:

```
FIND-SALARY.  
IF FOUND  
    PRINT 'CURRENT SALARY: ' ANNUAL-SALARY.
```

#### Example 3:

```
MOVE WORK-DATE TO SALARY-AS-OF-DATE.  
FIND-SALARY-AS-OF.  
IF FOUND  
    PRINT 'Salary:' ANNUAL-SALARY 'As-of: ' SALARY-AS-OF-DATE.
```

---

## NOTES

---

**Pointer manipulation, continued****FIND–**

Special FIND– verbs locate the current occurrence in stacked segments in The Solution Series. These verbs create a segment key that includes the segment, segment code, and often other segment key data.

In addition, the MKVERB facility enables you to create your own FIND– verb to locate the current occurrence of a new user defined segment.

☞ *Refer to the course [Cyborg Scripting Language Customization](#) for additional information about this topic.*

**Example 1**

FIND–DEDUCTION locates the first deduction (HED 501–999) in the employee’s H segments. IF FOUND verifies that pointer 32 is on an H segment.

**Example 2**

FIND–SALARY locates the salary (LZF) segment effective today. It must be followed by an IF FOUND or IF NOT–FOUND to test if the salary segment was found.

**Example 3**

FIND–SALARY–AS–OF locates the salary (LZF) segment effective as of the date in SALARY–AS–OF–DATE. It must be followed by an IF FOUND or IF NOT–FOUND to test if the salary segment was found.

# PROCESS/END-PROCESS

**Format:**

PROCESS non-key-field-1  $\left[ \begin{array}{l} FROM\ HERE \\ STARTING\ WITH \left\{ \begin{array}{l} literal \\ field-2 \end{array} \right\} \\ ENDING\ WITH \left\{ \begin{array}{l} literal \\ field-3 \end{array} \right\} \end{array} \right].$

*imperative statement ...*

END-PROCESS.

**PROCESS All Example:**

```
PROCESS MARITAL-CODE.  
  PRINT 'EMPLOYEE NAME: ' EMPLOYEE-NAME  
    ' VISA: ' VISA-TYPE ' ' VISA  
    ' VISA EXPIRATION: ' VISA-EXPIRE.  
NEXT-LINE.  
END-PROCESS.
```

**PROCESS with Qualifiers Example:**

```
PROCESS METHOD-CODE STARTING WITH '003' ENDING WITH '600'.  
  IF HED-NUMBER IS EQUAL TO '504' BYPASS-ENTRY.  
  PRINT HED-NUMBER ' ' METHOD-CODE.  
NEXT-LINE.  
END-PROCESS.
```

---

## NOTES

---

**Pointer manipulation, continued****PROCESS/END PROCESS**

The PROCESS verb enables you to establish a process loop. A process loop is a series of statements that are executed repetitively. Process loops are an effective way to check multiple-occurrence segments because the PROCESS logic is executed for each occurrence of a particular segment code.

- The PROCESS verb must be immediately followed by a non-key field from the segment you wish to manipulate.
- Unless you use one of these valid qualifiers, PROCESS begins at the first segment in the pointer and ends with the last segment in the pointer.
  - FROM HERE—Begins processing where the pointer is positioned when the program encounters the PROCESS verb.
  - STARTING WITH—Begins processing with the specified or next greater key value.
  - ENDING WITH—Ends processing with the specified key value.
- END-PROCESS is required and marks the end of the process loop.

**BYPASS-ENTRY**

You can use the BYPASS-ENTRY verb in a process loop. This verb causes processing to advance to the next occurrence.

# FIND and PROCESS with Dated Segments

Use these fields to convert a date from regular to complement format or from complement to regular format.

HOLD-DATE	Defines a date in century format (Pointer 8).
SAVE-DATE	Defines a date in regular format (Pointer 8).
SAVE-DATE-2	Defines a date in regular format (Pointer 8).
SAVE-DATE-CYYMDD	Defines a date in century format (Pointer 8).
WORK-DATE	Defines a date in century format (Pointer 8).
WORK-DATE-CYYMDD	Defines a date in century format (Pointer 6).
WORK-DATE-YYMMDD	Defines a date in regular format (Pointer 6).
CENTURY-SAVE-DATE	Defines a date in regular format (Pointer 8).

## FIND with a Century Date Key Example:

```
MOVE '19900101' TO CENTURY-SAVE-DATE .
MOVE CENTURY-SAVE-DATE TO SAVE-DATE-CYYMDD .
FIND CITIZENSHIP-CODE STARTING WITH SAVE-DATE-CYYMDD .
MATCH-SEGMENT-CODE .
IF NOT FOUND
    RETURN .
PRINT MARITAL-CODE CITIZENSHIP-CODE MILITARY-STATUS-CODE .
```

## PROCESS with a Century Date Key Example:

```
MOVE '19900101' TO CENTURY-SAVE-DATE .
MOVE CENTURY-SAVE-DATE TO SAVE-DATE-CYYMDD .
MOVE '19921231' TO CENTURY-SAVE-DATE .
MOVE CENTURY-SAVE-DATE TO HOLD-DATE .
PROCESS CITIZENSHIP-CODE
    STARTING WITH HOLD-DATE
    ENDING WITH SAVE-DATE-CYYMDD .
    PRINT MARITAL-CODE CITIZENSHIP-CODE
        MILITARY-STATUS-CODE .
NEXT-LINE .
END-PROCESS .
```

---

## NOTES

---

## Pointer manipulation, continued

### **FIND and PROCESS using special date fields**

There are four different kinds of dates used in The Solution Series:

- Regular dates in (CC)YYMMDD format
- Time-span dates in (Y)YYMMDD format
- Century complement dates in CYYMDD

This topic deals with how to translate century dates when the key to a segment includes a date, and lists the special date fields that enable you to hold, manipulate, and restore dates.

### **Century dates**

Century dates are stored in complement format that reverses the order in which the data is stored. When you use a century date, you still enter it in regular format and the system translates the date into complement format (CYYMDD), so that the most recent date resides at the top of a stacked segment. Users do not have to change the way that they enter the date because the system automatically translates it.

However, when you use date fields in a CSL program to FIND or PROCESS stacked segments, make sure that the dates have the same format as used in the segment. Special date fields have been defined to assist you to hold, manipulate, and restore dates.

## Section Summary

- **Work areas**
- **Pointer manipulation**

---

NOTES

---

## Section summary

In this section, you learned the techniques of pointer manipulation. Specifically you learned:

### Work areas

---

---

---

---

---

---

---

### Pointer manipulation

---

---

---

---

---

---

---

# Section 6 Exercise

---

**NOTES**

---

## Section 6 exercise

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in section 6. Perform the task below:

1. Create a new program that will:
  - Display the employee's name and address. Use the alternate payroll name/address (NAME-CODE 999) if present. If not, use the legal name/address (NAME-CODE 001).
  - Program Title: **EMPLOYEE JOB HISTORY**  
Place the employee's legal name on the right side of the title line.
  - Display Name and Address fields in the following format:

```

NAME                xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
CITY/STATE          xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
ZIP-CODE            99999

```

### Field Names

```

EMPLOYEE-NAME      (non-key)
CITY/STATE          (non-key)
ZIP-CODE            (non-key)

```

2. Below the name/address information, display the employee's job history since 1975 (JOB-EFFECTIVE from 1975 through current date). Display Job History fields in the following format:

```

JOB EFFECTIVE      CHANGE DESCRIPTION      JOB CODE
MM-DD-YYYY         xxxxxxxxxxxxxxxxxxxxxxxx      xxxxxxx
MM-DD-YYYY         xxxxxxxxxxxxxxxxxxxxxxxx      xxxxxxx
MM-DD-YYYY         xxxxxxxxxxxxxxxxxxxxxxxx      xxxxxxx

```

### Field Names

```

JOB-EFFECTIVE      (key)
CHANGE              (option list description)
JOB-CODE            (non-key)

```

3. Display a prompt at the bottom of the form to identify the Job History date range:

```

JOB HISTORY FROM MM-DD-YYYY TO MM-DD-YYYY

```

---

**NOTES**

---

## Appendix A: Exercise Answers

---

### Table of Contents

Section 2 exercise answers .....	246
Section 3 exercise answers .....	247
Section 4 exercise answers .....	248
Section 5 exercise answers .....	251
Section 6 exercise answers .....	253

## Section 2 exercise answers

Match each term with its proper definition.

- |                             |    |  |
|-----------------------------|----|--|
| <u>h</u> 1. CBSVO           | a. | A unique set of data elements which is used to identify and locate a record. |
| <u>f</u> 2. CBSVBT          | b. | A way to address data in Working Storage.                                    |
| <u>a</u> 3. Record Key      | c. | A piece of data within a record or a records segment.                        |
| <u>e</u> 4. Segment         | d. | 80-byte sequential output file.  |
| <u>b</u> 5. Pointer         | e. | A sub-division of a record which contains data of a particular kind.         |
| <u>c</u> 6. Field           | f. | Batch Cobol program which includes diagnostics for debugging purposes.       |
| <u>g</u> 7. Working Storage | g. | The memory area of The Solution Series software.                             |
|                             | h. | On-line Cobol program that allows real-time, interactive updating.           |
| <u>i</u> 8. Control Record  | i. | The Command Line for batch processing.                                       |
| <u>d</u> 9. FILE10          |    |  |

## Section 3 exercise answers

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in the section. In order to complete this exercise, build a form that will display the following employee information:

- Form Title—EMPLOYEE INFORMATION
- Form Fields:
  - SOCIAL-SECURITY-NBR
  - WORKERS-COMP-CODE
  - BIRTH-DATE
  - EMPLOYMENT-DATE
  - DATE-OF-TERMINATION

Make sure to display the employee name at the top of the form and to the right of the title. Also, make sure to display the name of the field and each field's content.

### Answer

```

00000 SECURITY ' ' . @ TITLE XX
00001 @LAST MODIFIED ON: 08-06-92 BY: USER AUTHOR: USER
00100 READ-EMPLOYEE.
00200 PRINT 'EMPLOYEE INFORMATION'. SPACE-OVER :28. INQUIRY-NAME.
00300 NEXT-LINE. NEXT-LINE.
00400 PRINT 'SOCIAL SECURITY NBR: ' SOCIAL-SECURITY-NBR.
00500 NEXT-LINE.
00600 PRINT 'WORKERS COMP CODE: ' WORKERS-COMP-CODE. NEXT-LINE.
00700 PRINT 'BIRTH DATE: ' BIRTH-DATE. NEXT-LINE.
00800 PRINT 'EMPLOYMENT DATE: ' EMPLOYMENT-DATE. NEXT-LINE.
00900 PRINT 'DATE OF TERMINATION: ' DATE-OF-TERMINATION.

```

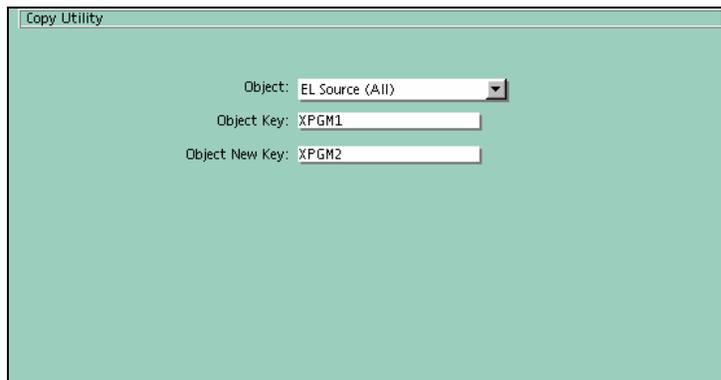
## Section 4 exercise answers

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in section 4. Complete the tasks below:

<b>COMMAND:</b> <b>S</b>
--------------------------

2.     **Copy the program you created in the previous section to a new program name.**



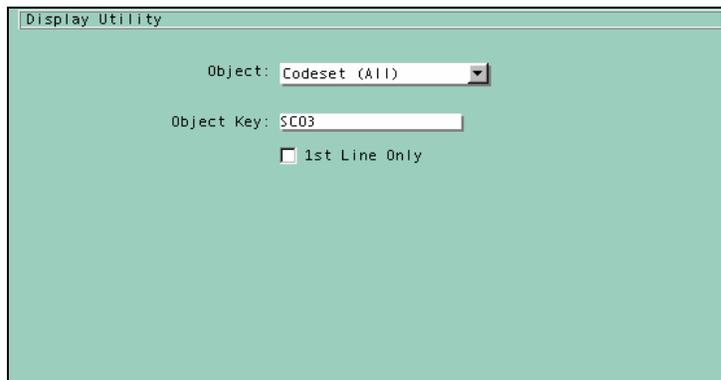
Copy Utility

Object: EL Source (All) ▼

Object Key: XPGM1

Object New Key: XPGM2

3.     **Display the records in the Option List SC03.**



Display Utility

Object: Codeset (All) ▼

Object Key: SC03

1st Line Only

Section 4 exercise answers, continued

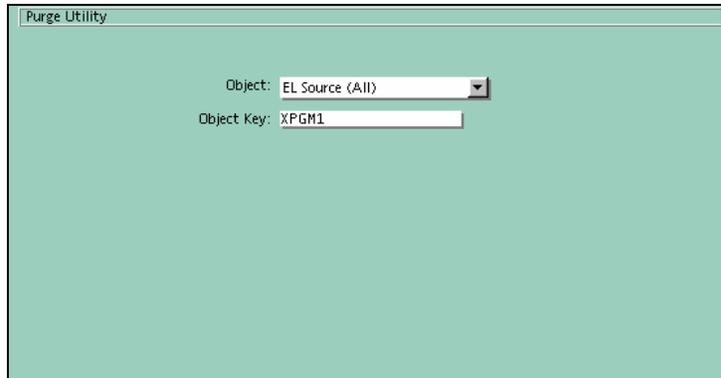
4. Extract the program you created in the previous section to FILE10.

**EXPORT Batch Control Record**

```

      1          2          3          4          5          6          7          8
.....+.....0.....+.....0.....+.....0.....+.....0.....+.....0.....+.....0.....+.....0
COMMENT AREA      999999EXPORT P/ XPGM2
    
```

5. Delete the program you created in the previous section.



**Section 4 exercise answers, continued**

**6. Import the program you extracted to FILE10 in question 3.**

**MAINTI Batch Control Record**

```
      1      1      2      2      3      3      4      4      5      8  
.....5.....0.....5.....0.....5.....0.....5.....0.....5.....0//.0  
COMMENT AREA      MAINTI
```

**7. Use FTLIST or F-MENU to answer the following questions.**

a. What is the Data Type of the BIRTH-DATE field in Pointer 29?

CYYMDD DATE (Century/Complement Date)

b. What is the Segment Type for the 'Basic Employee Data'?

E

c. What is the Option List for Sex Code in the 'Basic Employee Data'?

PP41

## Section 5 exercise answers

### Purpose

The purpose of this exercise is to give you practice using the information you have learned in section 5. Answer the questions or perform the task below.

Use the field definitions to supply the results to the program statements.

Field Name	Data Type	Storage Length	Content
FIELD-1	Alphanumeric	4	CSSS
FIELD-2	Alphanumeric	3	ABC
FIELD-3	Numeric 4 Decimals	6	050000
FIELD-4	Numeric 2 Decimals	6	400000
FIELD-5	Numeric 2 Decimals	6	025000
FIELD-6	Regular Date	6	900101

**1. Display the results for FIELD-5**

```
MOVE :0.00 TO FIELD-5.
CALCULATE FIELD-3 * FIELD-4 = FIELD-5.
```

FIELD-5 = 0000^00

**2. Display the results for FIELD-2**

```
MOVE FIELD-1 TO FIELD-2.
```

FIELD-2 = CSS

**3. Display the results for FIELD-3**

```
IF FIELD-1 EQUALS 'A' OR 'B' OR 'C' MOVE :10 TO FIELD-3
ELSE MOVE :20 TO FIELD-3.
```

FIELD-3 = 10^0000

## Section 5 exercise answers, continued

**4. Create a new program (XSEC5 subroutine) to calculate an employee's age.**

```
00000 SECURITY ' ' . @ TITLE XX
00001 @LAST MODIFIED ON: 06-01-94 BY: USER AUTHOR: USER
00100 IM-A-SUBROUTINE.
00200 CALCULATE CURRENT-DATE-CYYMDD - BIRTH-DATE = AGE.
00300 RETURN.
```

**5. Modify the program you created in Section 3 to transfer control (CALL) the age calculation subroutine and display the result of the calculation.**

```
00000 SECURITY ' ' . @ TITLE XX
00001 @LAST MODIFIED ON: 06-01-94 BY: USER AUTHOR: USER
00100 READ-EMPLOYEE.
00200 PRINT 'EMPLOYEE INFORMATION'. SPACE-OVER :28. INQUIRY-NAME.
00300 NEXT-LINE. NEXT-LINE.
00400 PRINT 'SOCIAL SECURITY NBR: ' SOCIAL-SECURITY-NBR.
00500 NEXT-LINE.
00600 PRINT 'WORKERS COMP CODE: ' WORKERS-COMP-CODE. NEXT-LINE.
00700 PRINT 'BIRTH DATE: ' BIRTH-DATE. NEXT-LINE.
00800 PRINT 'EMPLOYMENT DATE: ' EMPLOYMENT-DATE. NEXT-LINE.
00900 PRINT 'DATE OF TERMINATION: ' DATE-OF-TERMINATION.
01000 NEXT-LINE.
01100 CALL 'XSEC5 ' .
01200 PRINT 'AGE: ' AGE. NEXT-LINE.
```

## Section 6 exercise answers

### Purpose

Create a program to display the employee's alternate payroll name/address, if present, otherwise use the legal name/address, and display their job history from 1975 to the current date.

### Answer

```

P X4-SCR 00000 SECURITY ' '. @ EMPLOYEE JOB HISTORY MIHRUSER
P X4-SCR 00001 @LAST MODIFIED ON: 09-08-98 BY: USER AUTHOR: USER USER
P X4-SCR 00003 @This program displays the employee's payroll Name/Address USER
P X4-SCR 00004 @or legal Name/Address and Job History from 1975 to the USER
P X4-SCR 00005 @present. USER
P X4-SCR 00100 P100-START. USER
P X4-SCR 00200 READ-EMPLOYEE. USER
P X4-SCR 00300 PRINT ' EMPLOYEE JOB HISTORY'. SPACE-OVER :28. USER
P X4-SCR 00400 INQUIRY-NAME. NEXT-LINE. NEXT-LINE. USER
P X4-SCR 00500 P200-NAME-ADDRESS. USER
P X4-SCR 00600 FIND CITY/STATE STARTING WITH '999'. USER
P X4-SCR 00700 IF NOT FOUND USER
P X4-SCR 00800 FIND CITY/STATE STARTING WITH '001' USER
P X4-SCR 00900 IF NOT FOUND RETURN. USER
P X4-SCR 01000 SPACE-OVER :10. USER
P X4-SCR 01100 PRINT 'NAME ' EMPLOYEE-NAME. NEXT-LINE. USER
P X4-SCR 01200 SPACE-OVER :10. USER
P X4-SCR 01300 PRINT 'CITY/STATE ' CITY/STATE. NEXT-LINE. USER
P X4-SCR 01400 SPACE-OVER :10. USER
P X4-SCR 01500 PRINT 'ZIP CODE ' ZIP-CODE. NEXT-LINE. USER
P X4-SCR 01600 NEXT-LINE. USER
P X4-SCR 01700 P300-PROCESS-JOB-HISTORY. USER
P X4-SCR 01800 SPACE-OVER :06. USER
P X4-SCR 01900 PRINT 'Job Effective Change Description Job Code'. USER
P X4-SCR 02000 NEXT-LINE. NEXT-LINE. USER
P X4-SCR 02100 MOVE '19750101' TO CENTURY-SAVE-DATE. USER
P X4-SCR 02200 MOVE CENTURY-SAVE-DATE TO SAVE-DATE-CYYMDD. USER
P X4-SCR 02300 PROCESS JOB-CODE STARTING WITH CURRENT-DATE-CYYMDD USER
P X4-SCR 02400 ENDING WITH SAVE-DATE-CYYMDD. USER
P X4-SCR 02500 SPACE-OVER :07. USER
P X4-SCR 02600 PRINT JOB-EFFECTIVE ' ' CHANGE USER
P X4-SCR 02700 ' ' JOB-CODE. USER
P X4-SCR 02800 NEXT-LINE. USER
P X4-SCR 02900 MOVE JOB-EFFECTIVE TO CENTURY-SAVE-DATE. USER
P X4-SCR 03000 END-PROCESS. USER
P X4-SCR 03100 P400-PROMPT. USER
P X4-SCR 03200 NEXT-LINE. USER
P X4-SCR 03300 SPACE-OVER :06. USER
P X4-SCR 03400 PRINT 'JOB HISTORY FROM ' CENTURY-SAVE-DATE ' TO ' USER
P X4-SCR 03500 CURRENT-DATE-CYYMDD. USER
P X4-SCR 03600 RETURN. USER

```

---

**NOTES**

---

---

## Appendix B: Extra for Experts

---

### Table of Contents

Introduction.....	256
Hypertext development process.....	257
Scoping the extent of work required.....	257
Windows help .....	258
RTF footnotes .....	260
Windows Help compilation .....	263
Integration—the real scope of customization .....	263
How the Module Code Reference File is used to access context-sensitive help .....	265
How the cyborg.cnt file integrates custom help .....	266
How the A.cnt links the field level help topics.....	267
What to do with *.GID files.....	268
ASCII conversion chart .....	269
Detailed Directions .....	272
Task 1: Designing your custom help .....	273
Task 2: Creating your form level context-sensitive help.....	274
Task 3: Creating your field level context-sensitive help .....	276
Task 4: Place your files in the Support subdirectory .....	278
Task 5: Configuring context-sensitive access to your form help.....	278
Task 6: Integrating your custom program help with the modular help system .....	280
Task 7: Configuring context-sensitive access to your field help.....	281
Task 8: Integrating your custom field topics with field help.....	282

## Introduction

This appendix will help you identify what you must do to create custom help file(s) that can be integrated into The Solution Series Support System.

After you have created a custom form help program and a custom field help program, you must integrate them into The Solution Series Support System so your users can access your help.

This appendix also provides a quick reference to assist you in determining the appropriate Topic ID for a custom form topic.

## Hypertext development process

The Solution Series products introduce a comprehensive electronic support system. The system is complex because there are so many individual components, yet provides easy access to information. Elements of the Support System may be customized in order to introduce and integrate custom help into the system for context-sensitive access.

### Help development tools

If you are an experienced Help developer, you can program the RTF files directly in Word and use the Windows Help compiler, so long as you adhere to the topic ID rules detailed later in this appendix. If you are not an experienced developer, it is best to employ a development tool, such as RoboHelp. This tool provides an interface that walks you through the formatting process and help design. It also allows you the control necessary to override automation and set up the topic IDs so they adhere to our context sensitivity rules.

*Note:* *In the event you are an experienced help developer and do not wish to use a help development tool, we provide you with the Help Compiler Workshop (HCW.EXE), the Help Compiler program (HCRTF.EXE), and the Help Author's Guide help (HCW.HLP). We also provide you with a sample Help project (referenced in this appendix as WILSON help) on the Customer Center Documentation page.*

## Scoping the extent of work required

A help topic is a logical 'chunk' of information. In the source document, one topic is separated from another by a manual page break between the two. When compiled, each chunk of information is separated into uniquely identified 'topics' in the help.

There are only two kinds of help topics you must create in order to provide context-sensitive help topics to your users: form/program level topics and field level topics.

### ■ Field level context-sensitive topic

Description of the field; may include entry details.

### ■ Form level context-sensitive topic

Description of the form, its purpose, any special notes, and hypertext links to any related topics. You can include the following kinds of information in your form topic, as well:

- Task information (optional)  
Brief 'How To' instructions and steps for completing the form to perform a task, and hypertext links to any related topics.
- Concept information (optional)  
Detailed discussion of a subject that may help the user completing the task to understand why they are doing something in a particular way; alternatively, a factual bit of information that will be useful to the user when

completing the form.

This information can be specific to your form, or it may be related to more than one form—for example, it may be a high level discussion about a process involving many activities. If this is the case, you can make it a separate topic and then link to it from your form level topics (and link to your form level topics from this topic).

You can create as many or as few help topics as you wish, then insert hypertext jumps to link them. Or you may opt for a ‘One topic per form’ design option (recommended) that includes a description of the form and detailed tasks and steps for performing an operation with that form.

## Windows help

Compiled Windows help programs can be simple or complex. The Solution Series Support System is a very complex integration of over one hundred help files, as well as multimedia programs and ancillary files. For the purposes of creating custom help, you need only create two help programs: one for field descriptions and one for forms or other programs.

### Help file constants

When you compile a Windows Help program, you need source, instructions, and an interface in which to display the resulting compiled executable. These are constants.

Whether you are creating a help program for your custom fields or your custom programs and forms, the following information applies.

### Source files

The source material for a Windows help program is a Microsoft Word file that has been saved in rich text format (\*.rtf). This is called an ‘RTF’ file. You can have one or several RTF files compiled into a single help program.

There are three format rules for RTF files that are to be compiled into Help:

Every topic begins and ends with a manual page break (Control+Enter). This is what distinguishes one topic from another when the Windows compiler begins its work.

Every topic must have three footnotes with unique entries—Topic ID (identified by the hash symbol ‘#’), Topic Title (identified by the dollar symbol ‘\$’), and the Keyword (identified by the symbol ‘K’).

A Word template must be employed to format the text. This means applying styles, as opposed to assigning font and paragraph characteristics to each block of text. NEVER use spaces to align text.

*Note:* Help development tools automate topic creation, so you are not expected to edit the RTF coding directly.

### Why Microsoft Word?

Any word processor that supports rich text format (RTF) files can be used to create help topics.

Most Help development tools (and help developers) use Microsoft Word because some of the formatting and coding used to create source files (such as custom footnote markers) are more easily accessible in Word than in other word processors.

### Help project file

The Help Project file is called the 'HPJ' file. It contains the instructions to the Windows compiler. The information in this file is used to tell the compiler what the title of the help file will be, what the name of the compiled help file will be, what files will be read in as source and their locations, what the interface window will look like when it displays the help topics, and so forth.

*Note:* Help development tools provide an interface to the HPJ file, so you will not need to edit it directly. It will be created as part of the Help tool's development workflow.

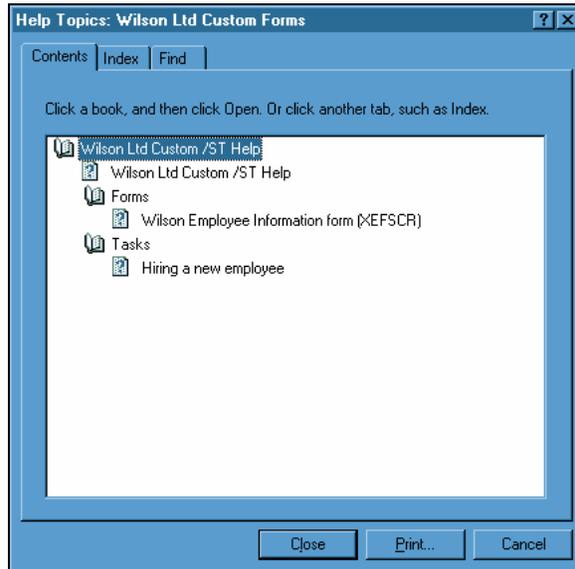
### Help Contents file

The Help Contents file is called the 'CNT' file. It is a text-based file that tells Windows how to display the menu for the compiled help. You can have a single CNT file for multiple compiled help files, thus creating your own modular custom help.

The following example CNT file (as displayed in Notepad)...

```
:Base wilsoncs.hlp>Main
1 Wilson Ltd Custom Help
3 Wilson Ltd Custom Help=WilsonLtdCustom/Help@wilsoncs.hlp>Main
2 Forms
3 Wilson Employee Information form (XEFSCR)= 088 069 079 083 067 082@wilsoncs.hlp>Main
2 Tasks
3 Hiring a new employee=Hiringanewemployee@wilsoncs.hlp>Main
```

...looks like this when you launch the help file (once you have clicked to expand the 'books'):



## RTF footnotes

As noted earlier, the footnote facility in Word is the means the Help development tool employs to apply search logic to help topics. There are three required footnotes for each of your topics, whether they are for programs or fields:

- Topic title
- Topic ID
- Keyword ID

### Topic Title (\$)

The topic title is the title that displays at the top of every topic. This will also display in the list of topics when you search for information on a subject.

Topic type	Suggested topic title format
field description	X-WILSON-DATE-12
form description	Wilson Ltd Employee Information (XEFSCR)
report description	Wilson Ltd Employee Information report (XEFRPT)
program description	Wilson Ltd Compression Program (WLCMP)
task/steps activity	Adding a Wilson employee to the system

**Topic ID (#)**

The Topic ID is a unique internal identifier, specially formatted to work with The Solution Series GUI and the help program. This is the identifier the GUI will search for when it launches the help file.

The reason the GUI searches for something other than the topic title is that both the GUI and the WinHelp compiler have certain limitations related to spaces and non-standard characters such as dashes and @ symbols. A general rule to follow when creating a Topic ID is not to use any spaces in your entry (this is done automatically by most help development tools).

The GUI has been designed to apply differing logic, depending on the type of access requested and the type of topic to be displayed.

*Note: Help development tools generally create topic IDs in the background, so you must override the topic ID automatically assigned and enter a topic ID in a format the GUI will expect.*

**Field description topic IDs**

Fields generally contain dashes in their titles. The GUI is programmed to replace dashes with the character ‘X’ when it searches the help program for the topic ID for a field. This means, the GUI is not looking for ‘X-WILSON-DATE-12’, but for ‘XXWILSONXDATEX12’. Your topic ID for the ‘X-WILSON-DATE-12’ field must be ‘XXWILSONXDATEX12’ or the GUI will be unable to locate the help topic.

Field name	GUI searches for	Help topic ID must be
X-WILSON-DATE-12	XXWILSONXDATEX12	XXWILSONXDATEX12

*Note: The field name is from the F-NAME table.*

**Form/Program description topic IDs**

The GUI identifies forms in the system by their program IDs. For example, the Employee Information form is identified by the GUI as ‘EF-SCR’.

You can name a form program anything, with any characters (though we strongly suggest you always follow Solution Series naming conventions). Variant forms generally contain an @ symbol, which is not an acceptable character for WinHelp. Because non-standard characters may be automatically replaced with underscores (‘\_’), duplication of topic IDs is a potential problem.

Rather than employ the ‘X’ replacement technique used for hyphens in field names (again, there would be a potential for duplication), the GUI converts each character in the program ID to its three-position ASCII equivalent. This is what it will search for, and your help topic ID must be what the GUI expects or the GUI will not locate the help topic.

Program ID	GUI searches for	Help topic ID must be
EF-SCR	069 070 045 083 067 082	069 070 045 083 067 082

*Note:* Do not place spaces between the three-position ASCII equivalents—they are simply displayed here for clarity.

**Keyword ID (K)**

Topic keywords are words (or phrases) you associate with a topic so users who use the Index facility in the help program can find the topics that answer their questions.

Usability studies show that the keyword list is the first place most users turn to for a quick answer to a specific problem. There are ways of integrating your indexes into The Solution Series Support System and the Support System’s indexes into your help, but this is a complex process. Since the goal in this appendix to ensure context-sensitive access, we will assume that you want your users to use your index on a stand-alone basis.

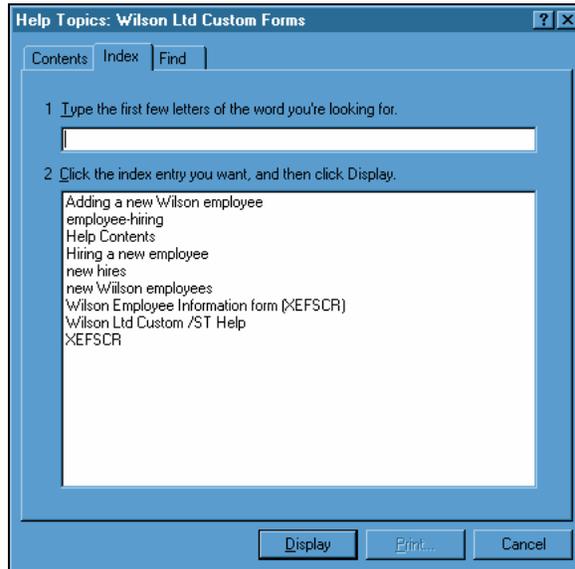
*Note:* Help tools generally provide a means of automatically entering and formatting index (keyword) entries.

In our Wilson Ltd example, we have a topic with the following Keyword footnote entry:

```
x Hiring a new employee;employee-hiring;new hires;new Wilson employees;Adding a new Wilson employee
```

Each key word or phrase is separated by a semicolon (;) with no spaces before or after it.

When you launch the Wilson help file and select the Index tab, you can see our entries have been inserted correctly:



## Windows Help compilation

To compile Windows help:

1. Create source and ancillary project files to instruct the compiler on what to compile and how it should display.
2. Invoke a help compiler.

Whether you are programming the help from scratch or if you use a help development tool, the same help compiler is invoked. In the case of a development tool, the compilation is behind the scenes—you just click a button and the help is automatically compiled via the Microsoft Windows Help Compiler (HCRTF.EXE).

## Integration—the real scope of customization

Proper integration will ensure your custom support displays when a user accesses the Support System in a context sensitive fashion.

All the components must work together. This is achieved by hypertext links between help files and by three text files that contain configuration information:

- modcode.ini
- cyborg.cnt
- a.cnt

These text files are referenced by The Solution Series GUI when it responds to a user request for help.

### Module Code Reference File

The Module Code Reference (modcode.ini) file, located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory, is the most important file in the system for accessing context-sensitive help. The GUI references it to find out which help file it should look in to display context-sensitive help for a particular form or field.

Most of the associations for form level help are done at the module code level. For example, if the form in question has a module code of PM, the GUI assumes the help file to open is 'posmancs.hlp', unless there is an entry in the EXCEPTIONS section that directs the GUI elsewhere for the topic.

Entries in the EXCEPTIONS section of this file direct (or redirect) the GUI to a different help file for a specific form in the system.

This file also acts as a lookup for field level help exceptions. The GUI locates the appropriate help file by identifying it according to the first letter of the field name.

The file contains four sections:

- **Module Codes**—A list of all module codes and the compiled One-Stop document to which they map. If your custom forms and programs within The Solution Series have a unique module code assigned to them, then you can map all such programs to your own custom help with an entry in this section for that module code.
- **Field Code Exceptions**—A list of all the fields that do not adhere to the logic assumed by the GUI (that is, a field description topic that does not reside in the help file provided for fields beginning with the letter with which this field begins). For example, if the field name begins with the letter 'X', but the field description is actually located in Customfields.hlp). This is recommended for custom work only, so you can maintain your custom field-level help separately, while still integrating it in with the delivered Support System.
- **Exceptions**—A list of exceptions to Module Codes section. Where a form or program does not map to the same One-Stop document as the rest of the forms for that module code, it will be listed here. If you have your own help topics that you want to display context-sensitively for a particular form, you make an entry here to redirect the GUI to your help file.
- **Menu Titles**—A list of all One-Stop documents and the text that should appear on the Help on the *One-Stop document* menu item on the Help menu. You can make an entry here to add your own help file to the list of those in The Solution Series Support System.

### Modular Support System Help Contents File

The modular help system in The Solution Series Support System is controlled by one Help Contents file (cyborg.cnt).

The Solution Series Help Contents (cyborg.cnt) file, located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory, may be edited so it displays and launches your custom help file.

### **Field Level Help Contents File**

The Field Level Help Contents (a.cnt) file, located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory, may be edited to access your custom field descriptions when a user browses through the field level help. This is the master file of all the field level help. Every field in the system is listed in this text file.

## **How the Module Code Reference File is used to access context-sensitive help**

The Solution Series references the Module Code Reference file (modcode.ini) to locate topics in your help files in a context-sensitive fashion.

### **Form level context-sensitive help**

Form or program-level help is activated by the GUI when:

- A user clicks on the Form Help button on the Help Toolbar.
- A user presses the F1 key while displaying a form.
- A user selects the Help on this Form menu item from the Help menu.

The GUI employs the following method to identify and launch the correct help topic in the correct help file:

1. Locate the module code for this component. The module code is contained in line 0 of the program code in a fixed position.
2. Locate the form name for this form. The form name is contained in line 0 of the program code in a fixed position.
3. Open and read the Module Code Reference File.
4. Search for the form name in the Exceptions section of the file. If the form name is present there, then retrieve the help file name assigned to it.
5. If the form name is not in the Exceptions section of the file, then search for the module code in the Module Codes section of the file. If the module code is present there, then retrieve the help file name assigned to it.
6. The final part of the file name for the help file location will be 'CS.HLP'—for example, if the form is a custom program for Wilson Ltd and a code WILSON has been set up for the Wilsoncs.hlp file, the file to be opened would be 'WILSONCS.HLP'.
7. If no module code is available or if the module code cannot be located in the Module Code Reference File, an error message displays and allows the user to

browse through the top level of The Solution Series Support System to locate the information.

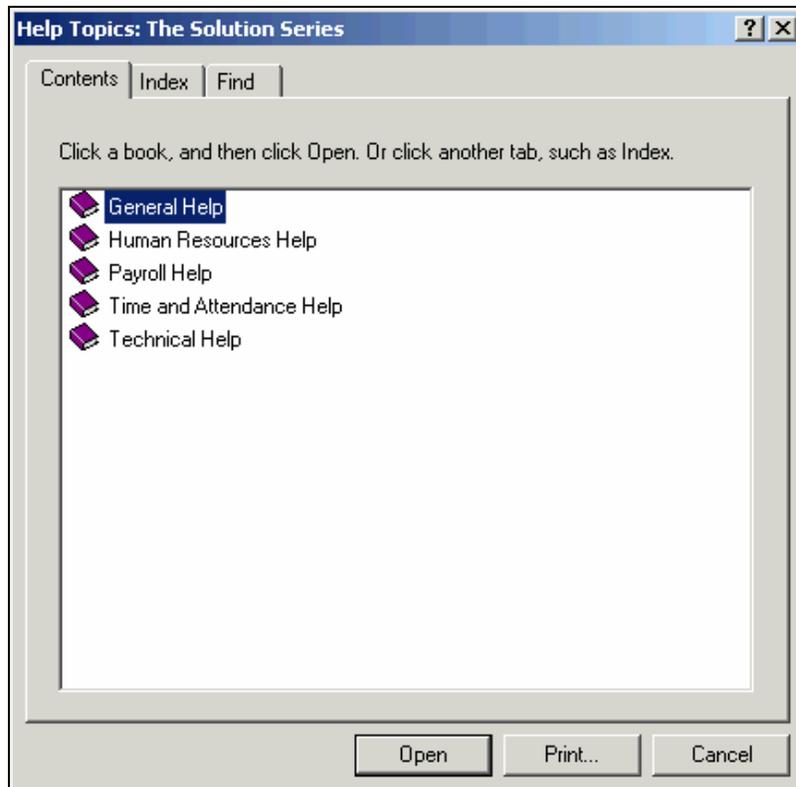
By adding in a reference to your help in the Menu Titles and a reference to your custom form in the Exceptions section, then equating it to your help file, you ensure that when a user accesses your form and requests help on the form, your help file will display.

### Field level context-sensitive help

Field topics are located according to the first letter of the name. For example, if a field name begins with an 'X', the system first looks in this exception section for the field name, then looks in the logical help file (X.hlp). If a field name begins with a numeral or 'A', the system first looks in this exception section for the field name, then looks in the logical help file (A.hlp). Each field will link to the correct field topic in the correct context-sensitive help file, via the What's This icon.

## How the cyborg.cnt file integrates custom help

When a user selects the 'Help on all of The Solution Series' option from the Help menu, a help file launches:



This is a list of all of the installed help components. If the user opens one of the books and clicks the topic below it, one of the help files is launched.

By editing the cyborg.cnt file in a text editor to add references to your help file, the custom help file is added to this list.

## How the A.cnt links the field level help topics

Field level help is activated by the GUI when:

- Users press the Shift and F1 keys while the cursor is in a field on a form.
- Users click the What's This icon on the Help Toolbar and then click on a field on a form.
- Users select the What's This? Menu item from the Help Menu and then click on a field on a form.

When the user requests field level help:

1. The GUI finds the name of this field in the form's Form Appearance Table (SAT file).
2. If this form has a SAT file, the GUI locates the first character of this field name. If the first character of the field name is a number, the letter 'A' is added to the start of the field name (for example, '12CODE' becomes 'A12CODE').

*Note:*

*The reason for these conditions is that our field names must be used as context-IDs to allow us to search for them, but the Winhelp engine does not support context-IDs that start with a number or contain non-alphanumeric characters.*

3. The GUI opens and reads the Module Code Reference File and searches for the field name in the Field Exceptions section of the file. If the field name is present there, then the GUI retrieves the help file name assigned to it.
4. If the field name is not in the Field Exceptions section of the file, then the GUI retrieves the help file name assigned to the first letter of the field name.

There are 26 unique, field level help files delivered, each with an associated Help Contents file:

- A.HLP – all fields and option list names beginning with the letter 'A' or a numeral
  - B.HLP – all fields or option lists beginning with the letter B
  - C.HLP – all fields or option lists beginning with the letter C
- and so forth.

Although there are also 26 Help Contents files, only one contains the master list of all the fields in the help—A.cnt. The other CNT files reference the A.cnt file.

This entry in the A.cnt file...

```

:
2 TYPE-DATE-4X=TYPEXDATEX4X@T.hlp>Concept
:
    
```

is interpreted like this...

Example	Function
2[space]	Hierarchy setting. Either a literal '1' or '2', followed by a space. Indicates if the entry is a heading ('1') or a field ('2').
TYPE-DATE-4X	Topic Title. The name of the topic to be displayed when the Help file is launched and the topic is displayed.
=	Literal '='. Links the Topic Title with the Topic ID.
TYPEXDATEX4X	Topic ID. Used by the WinHelp compiler (this is what the GUI searches for when it opens the help file and searches for a topic, so it must match exactly with what is in your RTF footnote).
@	Literal '@'. Indicates that the name of the help file where the field topic is located follows.
T.hlp	Identifies the help file where the field description topic is located.
>	Literal '>'. Indicates the proper window type in which to display the help topic follows.
Concept	Identifies to the WinHelp executable which type of predefined window in which to display the topic.

## What to do with \*.GID files

GID files include the latest information on the help file, but they are not rebuilt when you edit an existing CNT file or when you replace an existing help file that had been launched at some point previously. Therefore, it is important that you delete GID files from your system on a regular basis. If you update a help file, but do not delete the associated GID file, your changes will not display.

*Note:* When speaking, pronounce 'GID' as a single word; the 'G' is hard.

## ASCII conversion chart

Character	3-position code	Comment
NUL	000	(null)
SOH	001	(start of heading)
STX	002	(start of text)
ETX	003	(end of text)
EOT	004	(end of transmission)
ENQ	005	(enquiry)
ACK	006	(acknowledge)
BEL	007	(bell)
BS	008	(backspace)
TAB	009	(horizontal tab)
LF	010	(NL line feed, new line)
VT	011	(vertical tab)
FF	012	(NP form feed, new page)
CR	013	(carriage return)
SO	014	(shift out)
SI	015	(shift in)
DLE	016	(data link escape)
DC1	017	(device control 1)
DC2	018	(device control 2)
DC3	019	(device control 3)
DC4	020	(device control 4)
NAK	021	(negative acknowledge)
SYN	022	(synchronous idle)
ETB	023	(end of trans. block)
CAN	024	(cancel)
EM	025	(end of medium)
SUB	026	(substitute)
ESC	027	(escape)
FS	028	(file separator)
GS	029	(group separator)
RS	030	(record separator)
US	031	(unit separator)
SPACE	032	
!	033	
"	034	
#	035	
\$	036	
%	037	
&	038	
'	039	
(	040	
)	041	
*	042	
+	043	
,	044	

Character	3-position code	Comment
-	045	
.	046	
/	047	
0	048	
1	049	
2	050	
3	051	
4	052	
5	053	
6	054	
7	055	
8	056	
9	057	
:	058	
;	059	
<	060	
=	061	
>	062	
?	063	
@	064	
A	065	
B	066	
C	067	
D	068	
E	069	
F	070	
G	071	
H	072	
I	073	
J	074	
K	075	
L	076	
M	077	
N	078	
O	079	
P	080	
Q	081	
R	082	
S	083	
T	084	
U	085	
V	086	
W	087	
X	088	
Y	089	
Z	090	
[	091	

Character	3-position code	Comment
\	092	
]	093	
^	094	
_	095	
`	096	
a	097	
b	098	
c	099	
d	100	
e	101	
f	102	
g	103	
h	104	
i	105	
j	106	
k	107	
l	108	
m	109	
n	110	
o	111	
p	112	
q	113	
r	114	
s	115	
t	116	
u	117	
v	118	
w	119	
x	120	
y	121	
z	122	
{	123	
	124	
}	125	
~	126	
DEL	127	

## Detailed Directions

This section provides detailed instructions for the tasks discussed in this appendix.

<b>Task</b>	<b>Page</b>
Task 1: Designing your custom help .....	273
Task 2: Creating your form level context-sensitive help .....	274
Task 3: Creating your field level context-sensitive help .....	276
Task 4: Place your files in the Support subdirectory .....	278
Task 5: Configuring context-sensitive access to your form help .....	278
Task 6: Integrating your custom program help with the modular help system.....	280
Task 7: Configuring context-sensitive access to your field help.....	281
Task 8: Integrating your custom field topics with field help....	282

### Scenario

Your organization, Montolla Electric, has added a custom form to the delivered system:

Program Name	Program Title	Description of Custom Work
XEFSR	Montolla Employee Information Form	This program is based on the delivered Employee Information form (EF-SCR). The copy was altered to rename and enhance the EMP-NAME-CODE field and lengthen the EMP-LAST-NAME field to 25 positions.

*Note: These are just examples of the kinds of customizations you may have made; they are not changes we would suggest you make.*

## Task 1: Designing your custom help

There are a series of decisions to make before creating the help for the custom programs and fields.

### 1. Determine how many help files you want to create and their names

If you have any custom fields, you must create a separate help file for them. The instructions in this appendix assume you will create at least two help files—one for custom fields and one for custom forms/programs.

*Note:* Because the GUI assumes a 'cs.hlp' is appended to a core help file name when it opens a help file to find the form level context-sensitive topic for a displayed form, always end your form help file name with 'cs.hlp'.



*For practice, plan to create two help files. The form help file will be called Montollacs.hlp and the field help will be called Montollafields.hlp. Each of them is constructed in the same way (but with different names and different rules for creating topic IDs).*

### 2. Determine how many and the types of topics you must create

Review the help available for the original form that the new form is based upon.

Since there are two fields affected—one new and one altered—there will be two field level context-sensitive topics.

A review of the help for the Employee Information form (EF-SCR) reveals that there is one form level context-sensitive topic, with a link to a single task topic. The task topic has a single link to a concept topic.

You plan to create just one topic for your form and are satisfied that the documentation record you created is correct and complete.



*For practice, plan to create one topic in Montollacs.hlp and two topics in Montollafields.hlp.*

### 3. Decide if you want to create an overview topic

An overview topic can help users navigate through the form help if they open it and browse through it in a non context-sensitive manner.



*For practice, plan to create an overview topic.*

If you completed the Guided Practice, the scope of the work to be done to provide help for your organization is as follows:

Help File Name	Content
Montollacs.hlp	1. One new form-level context-sensitive topic
Montollafields.hlp	2. Two new field-level context sensitive topics

**4. Decide if you want to use a Help development tool**

If you elect to use a help development tool, you must still adhere to the basic design and adhere to the Topic ID rules described earlier in this appendix.



*For practice, plan to create your help files with the use of a Help development tool.*

**Task 2: Creating your form level context-sensitive help**

There are at least two ways to create content for your topic. One way is to write the content from scratch; the other way is to copy what was delivered and edit it for your purposes.

In this task, we will assume you are copying content and editing it for your custom programs.

**1. Identify the help delivered for the form your custom form is based upon**

When you are looking for the help related to a form, it is best to access the Support System in a context-sensitive fashion to identify the form level help.

To do this, launch The Solution Series, open the form your custom form was based upon, and click on the Question Mark Help on this Form help icon.



The help topic for that form displays. You can then click on any hypertext links in that topic to identify a task topic. Once the task topic is displayed, you can click on any See also hypertext links at the end of it to identify related concept topics.



*For practice, launch The Solution Series and navigate to the Employee Information form (EF-SCR) and follow the help links.*

**2. Copy and paste content**

Either copy the topic content from your custom documentation file and paste it into your topic or copy the content from the displayed, delivered help and paste it into your topic.

*Note: The content may extend over a single page. As long as there is no manual page break, the help compiler will recognize more than one page of content as belonging to the same topic.*

 *For practice, copy the content from the delivered Employee Information form (EF-SCR) into your Montolla Employee Information form (XEFSCR) topic.*

**3. Style the copied text**

Some styles do not translate well from the compiled help to a Word file, so the paragraphs must be restyled. Remember, it is important to always apply a style instead of formatting text manually.

 *For practice, restyle the content you copied from the help file. Do not use spaces to align text.*

**4. Edit the content with your additions/revisions**

It may be necessary to remove a step or to add information. Edit the content of the topic to reflect your requirements.

 *For practice, add a step in the topic to reflect the additional functionality of the new EMP-NAME-CODE field (it has five positions now, instead of three, and the label on the form is called 'ME Name Code').*

Once restyled, the content looks like this:

<b>3. → Enter the Name Code¶</b>		
Enter a unique, three-digit employee name value. Valid entries are:¶		
Name Code	→	Description¶
001	→	→ Legal name and address¶
002--998	→	→ User-defined names and addresses¶
999	→	→ Payment mailing address (if different from legal address)¶

If you followed the guided practice, it will look like this:

<b>3. → Enter the ME Name Code¶</b>		
Enter a unique, five-digit Montolla Electric employee name value. Valid entries are:¶		
Name Code	→	Description¶
00001	→	→ Legal name and address¶
00002--00998	→	→ User-defined names and addresses¶
00999	→	→ Payment mailing address (if different from legal address)¶

**5. Create hypertext links to other topics (optional)**

Follow the instructions for your Help development tool to create hypertext links to other topics in your help program.

 *For practice, do not create any hypertext links.*

**6. Save your restyled topic file**



*For practice, Save, using your Help development tool menu.*

**7. Create and edit the Help Contents file**

In order to display the compiled help so the user can browse and navigate through it, you must create a file called a Help Contents file. It is a listing of the topics in your custom help, displayed in a hierarchical structure. This may be performed automatically using your Help development tool.

**8. Create and edit the Help Project file**

The Help Project File determines what your compiled help will look like. Use your Help development tool to assign attributes such as Window size and color. We deliver a sample Help Project File as a reference if you wish to create your own custom HPJ file.

**9. Compile your help topics into a help file**

In order to make your source into a help file, you must use the Windows Help Compiler program (HCRTF.EXE). There are many methods for invoking the compiler, including invoking it from a DOS command line. Use your Help development tool to compile your help.



*For practice, compile your help program.*

**Task 3: Creating your field level context-sensitive help**

There are at least two ways to create content for your topic. One way is to write the content from scratch; the other way is to copy what was delivered and edit it for your purposes.

In this task, we will assume you are copying content and editing it for your custom fields.

**1. Identify the help delivered for the field on which your custom field is based**

When you are looking for the help related to a field, it is best to access the Support System in a navigation fashion to identify the field level help.

To do this, launch The Solution Series, open the form your custom form was based upon, and click on the Arrow Question Mark What's This? Help icon.



Click the field on which your field was based. The help topic for that field displays.

 *For practice, launch The Solution Series and navigate to the Employee Information form (EF-SCR) and display the help for EMP-NAME-CODE (the field label is 'Name Code'). This is the field that was enhanced in our scenario—your field name is 'XME-NAME-CODE'.*

## 2. Copy and paste content

Either copy the topic content from your custom documentation file and paste it into your topic or copy the content from the displayed, delivered help and paste it into your topic.

*Note:* *The content may extend over a single page. As long as there is no manual page break, the help compiler will recognize more than one page of content as belonging to the same topic.*

 *For practice, copy the content from your custom documentation file and paste it into your new topic.*

## 3. Style the copied text

Some styles do not translate well from the compiled help to a Word file, so the paragraphs must be restyled. Remember, it is important to always apply a style instead of formatting text manually.

 *For practice, restyle the content you copied from the help file. Do not use spaces to align text.*

## 4. Edit the content with your additions/revisions

Edit the content of the topic to reflect your requirements.

## 5. Save your restyled topic file

 *For practice, Save, using your Help development tool menus.*

## 6. Create and edit the Help Contents file

Use the delivered sample Help Contents file for the Wilson help project. Copy it and rename it. Open your new Help Contents file in Notepad. It looks like this:

```
:Base Wilsonfields.hlp>Concept
:Include A.cnt
```

Change the first line in the file so it reflects your custom help, for example:

```
:Base Montolla.hlp
:Include A.cnt
```

The ‘:Include A.cnt’ statement refers to the fact that this help file will use the Support System field help master Help Contents file. You will integrate your field help into the Support System in the following tasks.

Close and save the CNT file.

**7. Create and edit the Help Project file**

The Help Project File determines what your compiled help will look like. Use your Help development tool to assign attributes such as Window size and color. We deliver a sample Help Project File as a reference if you wish to create your own custom HPJ file.

**8. Compile your help topics into a help file**

In order to make your source into a help file, you must use the Windows Help Compiler program (HCRTF.EXE). There are many methods for invoking the compiler, including invoking it from a DOS command line. Use your Help development tool to compile your help.



*For practice, compile your help program.*

**Task 4: Place your files in the Support subdirectory**

Copy all your custom \*.HLP and \*.CNT files in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory.

**Task 5: Configuring context-sensitive access to your form help**

To configure context sensitive access to your form level topics, follow these steps:

**1. Open the modcode.ini file in Notepad**

The modcode.ini file is located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory. You may use a different text editor, but a simple text editor works better than a word processor, such as Word, because no format characteristics will be automatically applied to the text in the file.



*For practice, open the modcode.ini file from Notepad.*

- 2. Add a reference to a new module code in the [Module Codes] section**  
You must instruct The Solution Series to launch your own help file, instead of the delivered help file. To do this, you must add a line to the [Module Codes] section of the modcode.ini file.



*For practice, add a module code reference of 'ME=Montolla Electric' in the [Module Codes] section of the modcode.ini file.*

- 3. Add a reference to your form in the [Exceptions] section**

Direct The Solution Series to your help file by entering an exception. When a user requests help for this form, the system will read the modcode.ini [Exceptions] section to find out which help file it should launch to display the correct topic.



*For practice, create an exception of 'XEFSCR=MONTOLLA' in the [Exceptions] section of the modcode.ini file.*

- 4. Add a reference to your help file in the [Menu Titles] section**

Direct The Solution Series to display the title of your help file when it launches a context-sensitive topic by providing the title for your help file in the [Menu Titles] section of the modcode.ini file.



*For practice, create a menu title of 'MONTOLLA=Montolla Electric' in the [Menu Titles] section of the modcode.ini file.*

If you completed the Guided Practice, the entries should look similar to the example that follows:

```
[Module Codes]
HR=HUMRES
BA=BENADM
DS+DISTRB
:
ME=Montolla
:
[Exceptions]
EF-SCR=PAYROL
43-SCR=SALAD
50-SCR=SALAD
:
XEFSCR=MONTOLLA
:
[Menu Titles]
```

```
BENADMN=Benefits Administration
DISTRB=Distributed Administration
ELPROG=CSL Programming
HUMRES=Human Resources
:
MONTOLLA=Montolla Electric
:
```

**5. Close and save the Modcode.ini file**

 *For practice, save and close the modcode.ini file.*

**Task 6: Integrating your custom program help with the modular help system**

To integrate your help into the Support System modular help, follow these steps:

**1. Open the cyborg.cnt file in Notepad**

The cyborg.cnt file is located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory. You may use a different text editor, but a simple text editor works better than a word processor, such as Word, because no format characteristics will be automatically applied to the text in the file.

 *For practice, open the cyborg.cnt file from Notepad.*

**2. Create both a book and an 'include' entry**

The lines starting with a '1' are books in the help file, and the lines starting with ':include' are actual help files.

 *For practice, create both a book ('1 Montolla Electric Custom Help') and an 'include' entry (':include montollacs.hlp') for Montolla Electric help.*

**3. Create an index entry**

Optionally, you can add an index entry for the new help file in the next section. The new help file's index will be added to the overall Solution Series help index.

 *For practice, add an index entry for montollacs.hlp (':index Montolla Electric Custom Help=montollacs.hlp').*

If you completed the Guided Practice, the resulting form should look similar to the example that follows:

```

:
:
1 Technical Help
:include cybmk02.cnt
:include cybmk03.cnt
:include cybmk06.cnt
:include cybmk09.cnt
:include cybmk11.cnt

1 Montolla Electric Custom Help
:include montollacs.hlp
:
:
:index Using Training Administration=tranad01.hlp
:index Using Training Administration=tranad02.hlp
:index Using Training Administration=tranad03.hlp
:index Using Training Administration=tranad04.hlp
:index Using Training Administration=tranadcs.hlp
:index Montolla Electric Custom Help=montollacs.hlp

```

#### 4. Close and save the cyborg.cnt file



*For practice, save and close the cyborg.cnt file.*

#### 5. Delete the cyborg.gid file



*For practice, delete the cyborg.gid file.*

### Task 7: Configuring context-sensitive access to your field help

To configure context sensitive access to your field level topics, follow these steps:

#### 1. Open the modcode.ini file in Notepad

The modcode.ini file is located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory. You may use a different text editor, but a simple text editor works better than a word processor, such as Word, because no format characteristics will be automatically applied to the text in the file.



*For practice, open the modcode.ini file from Notepad.*

**2. Add a reference to your field in the [Field Exceptions] section**

Direct The Solution Series to your help file by entering an exception. When a user requests help for this field, the system will read the modcode.ini [Field Exceptions] section to find out which help file it should launch to display the correct topic.



*For practice, add an exception for your XME-NAME-CODE field of 'XME-NAME-CODE=MONTOLLAFIELDS.HLP>Concept' in the [Field Exceptions] section of the modcode.ini file.*

**3. Close and save the Modcode.ini file**



*For practice, save and close the modcode.ini file.*

**Task 8: Integrating your custom field topics with field help**

To integrate your field help into the Support System field help, follow these steps:

**1. Open the A.cnt file in Wordpad**

The A.cnt file is located in the [drive letter]:\Program Files\Cyborg Systems\Client50\Support\Help subdirectory. You may use a different text editor, but a simple text editor works better than a word processor, such as Word, because no format characteristics will be automatically applied to the text in the file. The A.cnt file is too large to be edited in Notepad.



*For practice, open the A.cnt file from Wordpad.*

**2. Create an index entry**

Optionally, you can add an index entry for the new help file. The new help file's index will be added to the overall *field level help* index.



*For practice, add an index entry for montollafields.hlp ('index Montolla Custom Field Help=montollafields.hlp').*

**3. Add a reference into the A.cnt for your help topic**



*For practice, enter a reference for your XME-NAME-CODE field ('2 XME-NAME-CODE=MONTOLLAFIELDS.hlp>Concept').*

If you completed the Guided Practice, the result should look similar to the example that follows:

```
:  
  
:Index Cyborg Systems Field Level Help=W.hlp  
  
:Index Cyborg Systems Field Level Help=X.hlp  
  
:Index Cyborg Systems Field Level Help=Y.hlp  
  
:Index Cyborg Systems Field Level Help=Z.hlp  
  
:Index Montolla Custom Field Help=MONTOLLAFIELDS.hlp  
  
:  
:  
  
2 XMCC-HED-HFCDR-FLAG=XMCC-HED-HFCDR-FLAG  
  
2 XMC-KEY=M.hlp>Concept  
  
2 XMC-LIMIT-KEY=M.hlp>Concept  
  
2 XME-NAME-CODE=MONTOLLAFIELDS.hlp>Concept  
  
2 XMEDICAL-COVERAGE-CD=M.hlp>Concept  
  
2 XMEDICAL-COVER-CD=M.hlp>Concept  
  
:
```

#### 4. Close and save the A.cnt file



*For practice, save and close the A.cnt file.*

#### 5. Delete the A.GID file



*For practice, delete the A.GID file.*

---

**NOTES**

---

## Appendix C: Pointer 7 Fields

---

### Table of Contents

Pointer 7 fields .....	286
Pointer 7 fields in pointer order .....	287
Pointer 7 fields in alphabetical order .....	288
Pointer 7 field definitions .....	289

## Pointer 7 fields

### Description

Pointer 7 is a work area used by the CBSV programs that can also be used by other programs. It consists of pre-designated fields and constants.

### Naming conventions

Work areas are defined as follows:

- W7-99-000-STAT
  - W—literal of W for work
  - 7—pointer number
  - 99—field length
  - 000—displacement within the pointer (relative to 0)
  - STAT—comments tag (ignored by the system)

## Pointer 7 fields in pointer order

Appendix W7-02-000	W7-01-003	W7-01-301	COMM-ACTION
TRACE-CONTROL		W7-01-302	COMM-CANCEL
W7-01-004	TRACE-READY	W7-01-303	COMM-INQUIRY
W7-01-005	QUOTE	W7-04-304	SESSION-ID
W7-02-006	LINE-ADVANCE	W7-06-316	QUERY-PROGRAM
W7-02-008	KEY01-SIZE	W7-02-322	QUERY-ALT-KEY
W7-24-010	KEY01-AREA	W7-19-324	QUERY-FROM-VALUE
W7-01-034	LOW-VALUE, LOW-VALUES	W7-02-343	QUERY-FROM-DUP
W7-09-035	MORE-LOW-VALUES	W7-19-345	QUERY-TO-VALUE
W7-02-044	KEY02-SIZE	W7-02-364	QUERY-TO-DUP
W7-32-046	KEY02-AREA	W7-01-366	QUERY-SWITCH
W7-14-078	SAVE-SEGMENT-KEY	W7-02-368	PAGE-LINES
W7-01-093	SCREEN-ERROR	W7-02-370	PAGE-LENGTH
W7-02-094	RECORD-LOCKED	W7-06-372	MENU-STOP
W7-01-096	SCREEN-WARNING	W7-13-378	TRACE-HOLD
W7-01-097	RECORD-UPDATED	W7-06-391	CURRENT-DATE-CYYMDD
W7-04-112	TRANS-NUMBER	W7-01-397	TRACE-SWITCH
W7-06-116	CURRENT-DATE	W7-01-398	AUTOHEADERS-SWITCH
W7-06-122	CURRENT-TIME	W7-01-399	HORIZONTAL
W7-48-128	CALL-STACK	W7-03-400	OPERATING-SYS-CODE
W7-01-176	SPACE, SPACES	W7-01-403	MACHINE-SWITCH
W7-61-177	MORE-SPACES	W7-01-428	SESSION-STATUS
W7-01-238	HIGH-VALUE, HIGH-VALUES	W7-01-429	CONTROL-COUNTRY
W7-01-239	MORE-HIGH-VALUE	W7-04-430	BALNCE-NUMBER
W7-06-240	COMPANY-NUMBER	W7-06-434	BACKREF-HOLD
W7-06-246	PROGRAM-FIELD	W7-01-440	PRODUCTION-VERSION
W7-01-252	CODE-1	W7-01-441	COLOR-INDICATOR
W7-01-253	CODE-2	W7-01-442	QM-PHASE
W7-10-254	KEY-FIELD	W7-01-443	DATE-FORMAT
W7-15-264	ADDITIONAL-KEY	W7-01-444	APPL-INDICATOR
W7-04-280	OPERATOR-ID	W7-03-445	MENU-PAGE
W7-01-298	COMM-CHAR	W7-01-448	AUTO-KEY-SWITCH
W7-01-299	COMM-MODE	W7-01-449	CODE-SET-SWITCH
W7-01-300	REPEAT-COUNTER		

## Pointer 7 fields in alphabetical order

W7-15-264	ADDITIONAL-KEY	W7-01-239	MORE-HIGH-VALUE
W7-01-444	APPL-INDICATOR	W7-09-035	MORE-LOW-VALUES
W7-01-448	AUTO-KEY-SWITCH	W7-61-177	MORE-SPACES
W7-01-398	AUTOHEADERS-SWITCH	W7-03-400	OPERATING-SYS-CODE
W7-06-434	BACKREF-HOLD	W7-04-280	OPERATOR-ID
W7-04-430	BALNCE-NUMBER	W7-02-370	PAGE-LENGTH
W7-48-128	CALL-STACK	W7-02-368	PAGE-LINES
W7-01-252	CODE-1	W7-01-440	PRODUCTION-VERSION
W7-01-253	CODE-2	W7-06-246	PROGRAM-FIELD
W7-01-449	CODE-SET-SWITCH	W7-01-442	QM-PHASE
W7-01-441	COLOR-INDICATOR	W7-02-322	QUERY-ALT-KEY
W7-01-301	COMM-ACTION	W7-02-343	QUERY-FROM-DUP
W7-01-302	COMM-CANCEL	W7-19-324	QUERY-FROM-VALUE
W7-01-298	COMM-CHAR	W7-06-316	QUERY-PROGRAM
W7-01-303	COMM-INQUIRY	W7-01-366	QUERY-SWITCH
W7-01-299	COMM-MODE	W7-02-364	QUERY-TO-DUP
W7-06-240	COMPANY-NUMBER	W7-19-345	QUERY-TO-VALUE
W7-01-429	CONTROL-COUNTRY	W7-01-005	QUOTE
W7-06-116	CURRENT-DATE	W7-02-094	RECORD-LOCKED
W7-06-391	CURRENT-DATE-CYYMDD	W7-01-097	RECORD-UPDATED
W7-06-122	CURRENT-TIME	W7-01-300	REPEAT-COUNTER
W7-01-443	DATE-FORMAT	W7-14-078	SAVE-SEGMENT-KEY
W7-01-238	HIGH-VALUE, HIGH-VALUES	W7-01-093	SCREEN-ERROR
W7-01-399	HORIZONTAL	W7-01-096	SCREEN-WARNING
W7-10-254	KEY-FIELD	W7-04-304	SESSION-ID
W7-24-010	KEY01-AREA	W7-01-428	SESSION-STATUS
W7-02-008	KEY01-SIZE	W7-01-176	SPACE, SPACES
W7-32-046	KEY02-AREA	W7-02-000	STAT-KEY
W7-02-044	KEY02-SIZE	W7-01-003	TRACE-CONTROL
W7-02-006	LINE-ADVANCE	W7-13-378	TRACE-HOLD
W7-01-034	LOW-VALUE, LOW-VALUES	W7-01-004	TRACE-READY
W7-01-403	MACHINE-SWITCH	W7-01-397	TRACE-SWITCH
W7-03-445	MENU-PAGE	W7-04-112	TRANS-NUMBER
W7-06-372	MENU-STOP		

## Pointer 7 field definitions

### **ADDITIONAL-KEY**

W7-15-264

This area contains the Additional Key field value from the Command Line. See also the SCNT11 text file.

### **ADDL-KEY-PROGRAM**

W7-07-272

This area is used by the MENU and DOCPRn programs to hold the program named in the Additional Key field.

### **APPL-INDICATOR**

W7-01-444

This area contains a code used to indicate which application is being executed. It is set when you log on to the system. Values are: H = Payroll/HR Solutions, T = Time and Attendance stand alone.

### **AUTO-KEY-SWITCH**

W7-01-448

This area contains a switch set by the REPEAT verb and reset by CYB90.

### **AUTOHEADERS-SWITCH**

W7-01-398

This area contains a switch used by the AUTO-HEADERS verb.

### **BACKREF-HOLD**

W7-06-434

This area is used by the MENU and DOCPRn programs to hold a back reference menu name.

### **BALNCE-NUMBER**

W7-04-430

This area is used by the BA-SCR program. When a value other than '0000' is present, an Inquiry form is returned.

## **CALL-STACK**

W7-48-128

This area contains a CALL stack. It can hold up to six entries. Each entry contains a program name followed by a two-byte computational field. The computational field contains the displacement of the next statement within the calling module when control is returned to it. **DO NOT USE THIS FIELD NAME IN A PROGRAM.** This field name has been provided for documentation purposes only.

## **CODE-1**

W7-01-252

This area contains the Code1 field value from the Command Line. This field receives an 'I' when an Inquiry Only Password is used to access the system. When this is the case, the I writes over any other option placed in the Code1 field on the Command Line. If special form processing uses the Code1 field, the program code must check the SCREEN-CODE1 field instead of the CODE-1 field for the option.

## **CODE-2**

W7-01-253

This area contains the Code2 field value from the Command Line.

## **CODE**

W7-02-252

This area is used to hold the CODE-1 and CODE-2 field values.

## **CODE-SET-SWITCH**

W7-01-449

This area contains a switch indicating how Option lists are treated. It is set by the ESTABLISH-QUOTE verb.

Values are:

Y—Treat Code Set type '9' as 'Y'.

N—Treat Code Set normally.

## **COLOR-INDICATOR**

W7-01-441

This field contains a monitor color indicator. A value of 'C' indicates color monitor. A value of blank indicates a monochrome monitor.

**COMM-CHAR**

W7-01-298

This field contains the value of the Code1 field from the Command Line. This field is then used by the COMM-MODE field when building form output lines. See also COMM-MODE.

**COMM-MODE**

W7-01-299

This field is used by the START-LINE and REPEAT verb loop to create either an entry box line or inquiry data line when building a form. This field acts as a switch to determine which type of output line is being built. To do this, the START-LINE verb checks to see what mode the screen is in by looking at the COMM-CHAR field. Based on the entry in COMM-CHAR, either one or two passes are performed to build the form line.

COMM-CHAR	COMM-MODE	LINE BUILT
E	E	entry boxes-one pass
I	I	inquiry data-one pass
blank	E	entry boxes-1st pass
I		inquiry data-2nd pass

Output lines are built field by field.

**COMPANY-NUMBER**

W7-06-240

This area contains the Control 1-2 value from the Command Line. See also the SCNT07 text file.

**CONTROL-COUNTRY**

W7-01-429

This field contains the country code from the AF transaction in the company header record. It is set when you log on to the system.

**CURRENT-DATE**

W7-06-116

This area contains today's date as YYMMDD.

**CURRENT-DATE-CYYMDD**

W7-06-391

This field contains today's date in the complement form.

### **CURRENT-TIME**

W7-06-122

This area contains the current time as HHMMSS. It is updated whenever a field with an edit pattern of '35' is PRINTed.

### **DATE-FORMAT**

W7-01-443

This area contains a code that determines the format of an edited date. It is set by CBSVO. Values are:

0—MM-DD-CCYY

1—DD-MM-CCYY

### **HIGH-VALUE**

W7-01-238

This area contains a literal of a high value. See also W7-01-239.

### **HORIZONTAL**

W7-01-399

This field contains an 'H' if horizontal entry verbs are in use.

### **KEY-FIELD**

W7-10-254

This area contains the Key field value from the Command Line. See also SCNT10 text file.

### **KEY-WORD**

W7-06-254

This area is used by the MENU and DOCPRn programs to hold the Key-word menu name entered in the Key field.

### **KEY01-AREA**

W7-24-010

This area is used to supply the program with a key for FILE01 prior to a READ-UNIQUE operation. See the ELKEY menu for more information.

### **KEY01-SIZE**

W7-02-008

This area is used to specify the length of the FILE01 key in the field W7-24-010 (KEY01-AREA). See the ELKEY menu for more information.

**KEY02–AREA**

W7–32–046

This area is used to supply the program with a key for FILE02 prior to an I/O operation. See also W7–02–044 (KEY02–SIZE).

**KEY02–SIZE**

W7–02–044

This area is used to specify the length of the FILE02 key in field W7–32–046 (KEY02–AREA).

**LINE–ADVANCE**

The LINE-ADVANCE command is used in –RP programs to define the number of blank lines you want to print following an output print line (carriage Control). There are two special LINE-ADVANCE values:

99—suppresses the output print line

00—forces a top of page prior to printing the output print line. Headings are NOT repeated.

If this command is used, you must also account for the lines added by calculating LINE-COUNT. The special 99 value can be used to eliminate duplicate total lines for an employee.

**LOW–VALUE**

W7–01–034

This area contains a literal of low values.

**MACHINE–SWITCH**

W7–01–403

This field contains a space or an 'S', depending on the machine type.

**MENU–PAGE**

W7–03–445

This area is used by the MENU program to retain the first three digits of the sequence field of the last menu used.

**MENU–STOP**

W7–06–372

This field is used by MENU and the DOCPRn programs to hold the name of a menu that when found in the back reference line causes the program(s) to stop.

**MORE-HIGH-VALUE**

W7-01-239

This area contains another literal of a high value. See also W7-01-238 (HIGH VALUE or HIGH VALUES).

**MORE-LOW-VALUES**

W7-09-035

This area contains nine bytes of low values. See also LOW-VALUE and LOW-VALUES

**MORE-SPACES**

W7-61-177

This area contains sixty-one spaces. See also W7-01-176 (SPACE or SPACES).

**OPERATING-SYS-CODE**

W7-03-400

This field contains the operating system code that identifies the computer type in use. See the text file for NEWJCL for a list of valid codes.

**OPERATOR-ID**

W7-04-280

This area contains the Operator-ID that appears on the right side of the Command Line. See also menu USER.

**OPTION-NUMBER**

W7-02-254

This area is used by the MENU and DOCPRn programs to hold the option number entered in the Key field.

**PAGE-LENGTH**

W7-02-370

This field is a computational field used in batch programs. It contains the maximum number of lines to be printed on a page.

**PAGE-LINES**

W7-02-368

This field is a computational field used in batch programs to hold a line count.

**PRODUCTION-VERSION**

A Y entered in this field defines this version of the Control File as the production file. The Y causes certain programs to behave differently than if this field was left blank. The Y directs FILE02 to produce a normal Audit Trail report.

**PROGRAM-FIELD**

W7-06-246

This area contains the Program name value from the Command Line. See also the SCNT08 text file.

**QM-PHASE**

W7-01-442

This area is used by the Query Maintenance Facility to keep track of what phase it is in.

**QUERY-ALT-KEY**

W7-02-322

This area contains the QUERY Alternate Key value.

**QUERY-FROM-DUP**

W7-02-343

This area contains the QUERY From Duplicate Key value.

**QUERY-FROM-VALUE**

W7-19-324

This area contains the QUERY From key value.

**QUERY-PROGRAM**

W7-06-316

This area contains the QUERY Program name.

**QUERY-TO-DUP**

W7-02-364

This area contains the QUERY To Duplicate Key value.

**QUERY-TO-VALUE**

W7-19-345

This area contains the QUERY To key value.

### **QUERY-SWITCH**

W7-01-366

This area contains a switch used by QUERY.

### **QUOTE**

W7-01-005

This field contains a literal of a quote. The quote is placed in this field by the ESTABLISH-QUOTE verb in the CYB88 programs. This field can be used in MOVE statements to place a quote on a form or report. See also the word description for ESTABLISH-QUOTE.

### **RECORD-LOCKED**

W7-02-094

This area is used to indicate the first pointer number associated with a record that is to be written or read. For example, the UPDATE-EMPLOYEE verb sets RECORD-LOCKED to '28' prior to calling CYBGET. Also, if RECORD-LOCKED is set to '40' in a form, the data in the SCREEN area is written to FILE02 as a 'ZZ' (or Audit Trail) record.

### **RECORD-UPDATED**

W7-01-097

This field is initially set up to a space when a form is entered. If any data is detected by an ENTRY verb, it is set to a 'Y'.

### **SAVE-SEGMENT-KEY**

W7-14-078

This area is used by the ENTRY verb to save a segment key.

### **SCREEN-ERROR**

W7-01-093

This area is initially set to an 'F' by the system before calling a form. If there is an error found on the form, the value is changed to an @ (at) sign and a reject message (\*\*REJECT\*\*) is returned. Once the error is corrected, this field is again set to an 'F'. See also the text file SCSP11.

### **SCREEN-WARNING**

W7-01-096

This area is initially set to a 'W' by the system before calling a form. Other values are:

@ If an ENTRY verb detects an error, this area is set to an '@', and a warning message (\*\*WARNING\*\*) is returned.

**SESSION-ID**

This field contains the current session number.

**SESSION-STATUS**

W7-01-428

This area contains the current session's status.

**SPACE**

W7-01-176

This area contains a literal of a space. When used in a compare, each position of the compare field is checked for spaces. The compare field must be the second field. For example,

IF SPACE EQUALS KEY-FIELD MOVE 'MAIN M' TO KEY-FIELD.

In this example, each position of the Key field is checked for spaces. See also W7-61-177 (MORE-SPACES).

**STAT-KEY**

W7-02-000

The STAT-KEY is set by CBSV COBOL programs after every I/O operation. The meaning of the values returned are:

00—Good I/O

01—Record read has greater key

10—EOF

22—Write failed due to duplicate record (1 or greater than 23)

23—Record not found (START, DELETE or REWRITE)

24—Space exhausted

90—Invalid request

91—Invalid file number (less than 23)

93—Invalid file number (8 or 9)

95—Open failed

99—Invalid key

**TRACE-HOLD**

W7-13-378

This area is used by the TRACE facility to hold

# the name of the program

# a switch indicating whether the trace is on or off

# a starting paragraph number

**TRACE-READY**

W7-01-004

This is an internal switch used by the TRACE program.

**TRACE-SWITCH**

W7-01-397

This area contains a switch used by the TRACE program. A value of T means the CBSVOT or CBSVBT program is executing.

**TRANS-NUMBER**

W7-04-112

This area contains the current transaction number.

## Appendix D: Quick Solution—Technical Reference Guide

---

### Table of Contents

System Control Repository File object codes .....	300
Operating system codes .....	301
Edit command list .....	302

## System Control Repository File object codes

System Control Repository File object codes are used to select specific System Control Repository (FILE01) records for processing.

<u>Object Description</u>	<u>Object</u>	<u>Object Description</u>	<u>Object</u>
A Records	A	Locked Records	ZL
Alt Lang Doc	P/N	Menu	P/L
Alt Lang Error Msgs	P/Q	Obj/Act Security	PXS
Alt Lang Menu	P/K	Object Code	P/X
Alternate Key Records	Q	Objects	PX
Assembler Source	P/W	P Records (All)	P
B Records	B	P Records (No Obj)	P-X
Option List (All)	C	PAYXTR/PAYMRG Select	EXT
Option List Doc	C/D	PTF History	P/H
Option List EL Calc	C/M	Pos Mgt Navigation	EPM
Option List EL Edit	C/R	R Records	R
Option List Value	C/V	RDBMS View Names	ETL
Option List W/Country cd	C/C	Reload Messages	P/E
Consultant Tables	W	Rpt C1-2 Schedule	PD
Context Menus	ECM	Rpt C1-2 Validation	PC
Distr Sol Tables	U	Rpt Print Pos Recrds	RT
Documentation	P/M	Rpt Schedule	PE
Edit Heading	PXH	Sample JCL	P/J
EL Rpt Components	RPT	Screen Chain	P/C
EL Rpt Source (All)	RS	Screen Items Table	P/S
EL Source	P/	Security Records	Y
EL Source (All)	SRC	Security Violations	Y/V
EL Source-Generated	P/G	Sol View Qry Specs	RQM
Error Messages	P/R	Sol View Qry Src	QRY
Field Documentation	F/D	Sol View Rpt Specs	RRM
Field Entry	F1	Sol View Scr Specs	RSM
Field Name Tbl Menu	FTM	Sol View Screen Src	SCR
Field Name Tbl Xref	FTX	Sol View Xref Recrds	XRF
Field Security	FS	Sol View Xtr Specs	RXM
Field Table (All)	F	Sol View Xtr Src	XTR
FILECL Updates	MCL	Table	T
GUI Menu	MMN	Transaction Data	P/T
HRMS L Defaults	D	User Tables	X
Hypertext Titles	P/D	Verbs	F/V
		Virtual Pgm Name	PI

## Operating system codes

Operating System Codes are used to distinguish unique platform characteristics.

<u>Manufacturer</u>	<u>Code</u>	<u>Environment</u>	<u>Manufacturer</u>	<u>Code</u>	<u>Environment</u>
Bull	HW7	DPS 7 (5)	Prime	PR1	PRIME (VT100) (5)
	HW8	DPS 8 (5)		PR2	PRIME COBOL 2 (5)
Data General	DG	DATA GENERAL (4)		PRM	PRIME (PT200) (5)
Digital Equipment	VAX	DEC/VAX (5)	Unisys	U11	UNIVAC 1100/TIP (4)
	VAR	DEC/VAX RDB (5)		U1M	UNIVAC 1100/MCB (4)
Hewlett Packard	HP3	H/P 3000 (4)		U1S	UNIVAC 1100/SFS (4)
	HPS	H/P SPECTRUM (4)		U31	UNIVAC OS3 (4)
IBM	AS4	IBM AS/400 (5)		U30	UNIVAC 9030 (4)
	ASR	IBM DB2/400		BME	BURROUGHS Med 5
	CM2	IBM VM/CMS COBOL 2 (4)			MEDIUM (5)
	DO2	IBM DOS COBOL2 (4)		BLR	BURROUGHS LARGE (6)
	DO2	IBM VSE/ESA COBOL 2 (4)		BLC	BURROUGHS COMS (6)
Unisys	DOS	IBM DOS (4)	Wang	WNG	WANG (5)
	IDM	IBM IDMS (4)	Various - UNIX	MF2	AIX, GPUX, ULTRIX,
	ID2	IBM IDMS COBOL 2 (4)			XENIX (5)
	IMS	IBM IMS (4)	Various - PC	PC	IBM Compatible PC (4)
	IM2	IBM IMS COBOL 2 (4)		MFJ	NT SQL Server
	OS	IBM OS/MVS (4)		MFT	NT Indexed
	OS2	IBM OS/MVS COBOL 2 (4)			
	OSR	IBM OS/MVS DB2 (4)			
	S38	IBM System 38 (5)			
NCR	NCB	NCR (BOSS 3) (5)			
	NCM	NCR (MEGAPATH) (5)			
	NCR	NCR (NO MONITOR) (5)			
	NCT	NCR (TRANPRO) (5)			

## Edit command list

The EDIT form contains a secondary command line with two fields. These EDIT command fields allow you to perform various editing functions on the current object.

The first field, referred to as the command field, requires a one-character command code to perform actions within the EDIT program. The second field, referred to as the Parameter field, is used to specify parameters related to certain command codes.

*Note:* Some codes are valid for specific objects.

\* Denotes commands valid for all P records

\*\*Denotes commands valid for OBJECT = P/

### **A—AUTO ADD MODE\*\***

The A command automatically adds sequence line numbers to the end of your file in increments of 100. It places an A (Add) in the first column of each added line, making them ready for entry. Auto Add Command:

(A)( )

### **B—BACKUP 19 LINES\***

The B Command allows you to display the previous page of your file.

Backup 19 Lines Command:

(B)( )

### **C—CHANGE STRING**

The C command works like a global change. You can request that the system display all lines containing the specified word or string together or one line at a time. You must use literals, that is, the exact characters as they appear in the string.

Change all Command:

(C)/READ-EMPLOYEE/UPDATE-EMPLOYEE/ )

Change One Command:

(C)(READ-EMPLOYEE/UPDATE-EMPLOYEE/1 )

**D—DELETE RANGE\***

The D command allows you to delete lines.

Line delete Command:

(D)(00220, 00240, 00530 )

Range delete Command:

(D)(00220/00350 )

**E—END AUTO ADD MODE\*\***

The E command ends the Auto Add Mode and returns the form to manual operation.

End Auto Add Command:

(E)( )

**F—FIND STRING**

The F command will display all lines containing the specified word or string together or one line at a time. You must use literals, that is, the exact characters as they appear in the string. You can also use an equal sign (=) as a wild card in all but the first position of the search argument.

Find all Command:

(F)(/W8-01-999/ )

Find one Command:

(F)(/CONTROL-1-2/1 )

Find Wildcard Command:

(F)(CONTROL - = CODE/ )

**G—GO TO SEQUENCE NUMBER**

The G command can select a line to appear as the first line on your form.

Go To Command:

(G)(00312 )

**H—HOLD RANGE\*\***

The H command is used to hold a range of lines for the purpose of inserting (I) or Transferring (T).

Hold Range Command:

(H)(00100/00200 )

### **I—INSERT HELD RANGE\*\***

The I command is used with the H (HOLD) command to copy text or code within the same file. There must be room between the existing lines to insert the copied lines. The inserted line numbers must be currently unused and not duplicate existing line numbers. The inserted lines are renumbered in increments of 10 unless otherwise stated.

Insert Command:

(I)(00205 )

Insert with increments of 5 lines Command:

(I)(00205/005 )

### **K—KEY CHANGE**

The K command is used to quickly change the EDIT form from the file you are editing to a different file within the same record type (OBJECT code). If you are editing a documentation file (P/M) and wish to access another P/M file, you should use this quick method. You can also use the K command to create a new file.

Change file command:

(K)(MYFILE )

### **L—LOGGING ON/OFF\*\***

The L command is used to activate LOGGING. Logging saves the original lines of code, as they were prior to any editing, and places them in FILE01. They are saved under your Operator ID and are remembered for as long as LOGGING stays on.

Turn Logging On/Off Command:

(L)( )

### **N—NEXT SCREEN**

The N command is used to advance to the next page of text or source. Typically, your display scrolls automatically when you press your enter key. You may use the N command to force the display to immediately advance to the next page when you make modifications and press your enter key.

Next Screen Command:

(N)( )

**O—OBJECT CHANGE**

The O command is used to access a file that has an Object code that is different from the file on your EDIT form. For example, if you are editing a Option List file (C/V) and wish to change to a (P/ ) file, you must use this method.

Object Change Command:

(O)(P/ )

**P—PARAGRAPH LOCATE\*\***

The P command is used to locate and display a specified paragraph in your program. The paragraph label must begin in the first position of your code line.

Paragraph Locate Command:

(P)(380 )

**R—RELOAD\*\***

The R command automatically reloads, or compiles, your CSL Source code (OBJECT = P/). The same rules apply as when performing RELOAD from the Command Line.

Reload Command:

(R)( )

**S—RESEQUENCE\*\***

The S command used to resequence the lines of your program in increments of 100. The lines numbered from 00000 through 00010 are reserved and are NOT included in the resequence.

Solution Series programs should never be resequenced.

Resequencing Command:

(S)( )

**T—TRANSFER RANGE\*\***

The T command is used with the H (HOLD) command to transfer held ranges within a file (Move) or to another file (Copy). The lines are copied/moved in increments of 10 unless otherwise specified.

To COPY from one file to another, do the following:

Execute the H (HOLD) command, specifying the lines to be copied.

(H)(00310/00330 )

Execute the K (KEY CHANGE) command to access the program to which you wish to copy the lines.

(K)(NEWFILE        )

### **T—TRANSFER RANGE\*\* continued**

Enter a T in the Command field and the beginning sequence line number at which you want to copy the lines (in HOLD) in the Parameter field. If you wish the copied lines to have an increment other than 10, type a slash after the beginning sequence line number and then a three-digit increment.

(T)(00800        )

OR

(T)(00800/050)

To MOVE code within a file complete the following sequence of commands:

Execute the H (HOLD) command, specifying the lines to be copied.

(H)(00310/00330        )

Enter a T in the Command field and the beginning sequence line number at which you want to move the lines (in HOLD) in the Parameter field.

(T)(00800        )

OR

(T)(00800/050)

*Note:*        *The T command MOVES code to another place within the program you are currently editing. It automatically deletes the lines from their original location.*

### **U—UNDO\*\***

The U command is used with the L command (LOGGING), to reverse or ‘undo’ the previously performed edit. For example, if you deleted lines of code by mistake, UNDO will restore them.

*Note:*        *Undo assumes that the L command (LOGGING) is active and the counter to the right of the EDIT Command Line contains a number. UNDO the last add, change or delete command:*

(U)(        )

UNDO all the edited lines that are accounted for in the counter command:

(U)(ALL        )

*CAUTION:* *If you ‘undo’ all prior editing, LOGGING automatically turns itself off.*

## X—EXECUTE PROGRAM/SCREEN

The X command is used to execute a form directly from the EDIT form command line, instead of using The Solution Series Command Line (top line on form). The Parameter field entries must correspond with The Solution Series Command Line entry fields. You must allow for spaces between entries, as necessary.

Execute Program/Screen Command:

```
(X)(EE-SCR 1234567890 )
      :      :      :      :
Program Code Key      Additional-Key
field   field field   field
(1-6)   (7-8) (9-18)   (19-33)
```





a.cnt.....	263, 282	Compound conditional.....	177
how it links topics.....	267	Conditional statement.....	171
interpreted.....	268	Configuring context-sensitive access to your field help.....	281
location.....	265	Configuring context-sensitive access to your form help.....	278
a.hlp.....	266	Constants.....	219
A—AUTO ADD MODE.....	302	context sensitive help	
Add a new program.....	85	how it works for fields.....	266
ADDITIONAL—KEY.....	289	how it works for forms.....	265
ADDL—KEY—PROGRAM.....	289	context-sensitive access to your field help	
Alphanumeric comparison.....	175	configuring.....	281
Alphanumeric MOVE.....	164, 165, 168	context-sensitive access to your form help	
AND in a compound conditional.....	177	configuring.....	278
APPL—INDICATOR.....	289	context-sensitive help	
ASCII.....	256	creating field help.....	276
ASCII conversion chart.....	269	creating form level help.....	274
Auto add mode.....	89	CONTROL—COUNTRY.....	291
AUTOHEADERS—SWITCH.....	289	Copying System Control Repository data ....	119
AUTO—KEY—SWITCH.....	289	Course overview.....	3
BACKREF—HOLD.....	289	Creating a CSL program.....	75
Backup command.....	103	Creating your field level context-sensitive help.....	276
BALNCE—NUMBER.....	289	.....	274
Basic verbs.....	63	Creating your form level context-sensitive help.....	274
Batch control record.....	47	.....	274
B—BACKUP 19 LINES.....	302	Cross—Reference Report.....	131
BYPASS—ENTRY.....	237	Cross—Reference Report (CROSSX/CROSSP).....	143
CALCULATE.....	183	CSL basic commands.....	63
Calculating.....	183	CSL programming rules.....	57
CALL.....	193	CSL programming utilities.....	75
CALL/RETURN.....	191	CURRENT—DATE.....	291
CALL—STACK.....	290	CURRENT—DATE—CYYMDD.....	291
CBSV and CSL relationship.....	19	CURRENT—TIME.....	292
C—CHANGE STRING.....	302	custom field topics	
Century dates.....	239	integrating with field help.....	282
CHAIN.....	191, 199	custom help	
Change string command.....	109	designing.....	273
Class test.....	175	Cyborg Scripting Language (CSL).....	17
CLEAR-SCRATCH.....	225	cyborg.cnt.....	263, 267
CNT		integrating custom help programs.....	280
description.....	259	location.....	264
CODE.....	290	Data display verbs.....	67
CODE—1.....	290	DATE—FORMAT.....	292
CODE—2.....	290	D—DELETE RANGE.....	303
CODE—SET—SWITCH.....	290	Delete line command.....	111
COLOR—INDICATOR.....	290	Designing custom help.....	273
COMM—CHAR.....	291	Displaying System Control Repository data.....	121
COMM—MODE.....	291	Edit commands.....	101
COMPANY—NUMBER.....	291		
Comparison rules.....	175		
Compiling CSL source.....	91		

- 
- Edit manipulation commands ..... 101, 107
  - Edit miscellaneous commands ..... 101
  - Edit navigation ..... 103, 105
  - Edit navigation commands ..... 101
  - Edit text body ..... 83
  - Edit Utility (EDIT) ..... 81
  - Editing sequence lines ..... 89
  - E—END AUTO ADD MODE ..... 303
  - End auto add mode ..... 89
  - Ending conditional statements ..... 173
  - exceptions
    - fields ..... 264
    - programs ..... 264
  - Exceptions
    - in Module Code Reference File ..... 264
  - Execute a program command ..... 117
  - EXPORT ..... 125
  - Exporting/Importing System Control Repository data ..... 125
  - F—FIND STRING ..... 303
  - Field Code Exceptions
    - in Module Code Reference File ..... 264
  - Field description topic IDs ..... 261
  - field level context-sensitive help
    - creating ..... 276
  - Field level context-sensitive topic
    - description ..... 257
  - Field Level Help Contents File
    - description ..... 265
  - Field menu ..... 135
  - Field table list ..... 131
  - Field table menu ..... 131
  - Figurative constant ..... 157
  - FIND ..... 231
  - FIND— ..... 235
  - FIND and PROCESS using special date fields ..... 239
  - Find string command ..... 109
  - form level context-sensitive help
    - creating ..... 274
  - Form level context-sensitive topic
    - description ..... 257
  - Form/Program description topic IDs ..... 261
  - FROM HERE ..... 231
  - G—GO TO SEQUENCE NUMBER ..... 303
  - GID files ..... 268
  - GO TO ..... 185, 189
  - Go to command ..... 103
  - help
    - how it is integrated ..... 266
  - Help Contents file
    - description ..... 259
  - Help development
    - scoping the work ..... 257
  - Help development process ..... 257
  - Help development tool
    - recommendations ..... 257
  - Help project file
    - description ..... 259
  - help topic
    - definition ..... 257
  - Help topic
    - description for field topics ..... 257
    - description for form topics ..... 257
  - H—HOLD RANGE ..... 303
  - HIGH—VALUE ..... 292
  - Hold range command ..... 113
  - HORIZONTAL ..... 292
  - HPJ
    - description ..... 259
  - Hypertext development process ..... 257
  - IF/ELSE summary ..... 181
  - I—INSERT HELD RANGE ..... 304
  - IM-A-SUBROUTINE ..... 191
  - Imperative statement ..... 171
  - INQUIRY—NAME ..... 69
  - Insert held range command ..... 113
  - Integrating your custom field topics with field help ..... 282
  - Integrating your custom program help with the modular help system ..... 280
  - Key change command ..... 105
  - KEY01—AREA ..... 292
  - KEY01—SIZE ..... 292
  - KEY02—AREA ..... 293
  - KEY02—SIZE ..... 293
  - KEY—FIELD ..... 292
  - KEY—WORD ..... 292
  - Keyword (#)
    - rules ..... 262
  - Keyword ID (K)
    - description ..... 262
  - K—KEY CHANGE ..... 304
  - Label to field cross-reference ..... 131
  - LINE—ADVANCE ..... 293
  - LINK ..... 191, 197
  - L—LOGGING ON/OFF ..... 304
  - Logging command ..... 107
  - Logical data structure ..... 35
  - LOW—VALUE ..... 293
  - MACHINE—SWITCH ..... 293
  - Macro verbs ..... 63
  - MAINTI ..... 125
  - Master record segments ..... 33
  - MATCH— ..... 233
  - Menu Titles
-

in Module Code Reference File .....	264	Pointer 8 .....	223
MENU-PAGE .....	293	Pointer manipulation .....	229
MENU-STOP .....	293	Pointers .....	229
modcode.ini .....	263	Pointers and segments .....	39
configuring field level context sensitivity	281	P-PARAGRAPH LOCATE .....	305
configuring form level context sensitivity	278	Prescribed sequence lines .....	87
Modular Support System Help Contents File		PRINT .....	67
description .....	264	Print/Display System Control Repository data	
Module Code Reference File		.....	131
description .....	264	PROCESS/END PROCESS .....	237
how it works .....	265	PRODUCTION-VERSION .....	295
Module Codes		Program Memory .....	131
in Module Code Reference File .....	264	Program Memory Map report .....	147
MORE-HIGH-VALUE .....	294	Program types .....	15
MORE-LOW-VALUES .....	294	Program/Field reports .....	131
MORE-SPACES .....	294	PROGRAM-FIELD .....	295
MOVE statement .....	157	Programming with work areas .....	213
Moving data .....	157	Purging System Control Repository data.....	123
Negative compound conditionals.....	179	QM-PHASE .....	295
Negative conditions .....	179	QUERY-ALT-KEY .....	295
Next screen command.....	103	QUERY-FROM-DUP .....	295
NEXT SENTENCE .....	173	QUERY-PROGRAM .....	295
NEXT-LINE.....	71	QUERY-SWITCH .....	296
N-NEXT SCREEN.....	304	QUERY-TO-DUP .....	295
Numeric comparison.....	175	QUERY-TO-VALUE .....	295
Numeric MOVE.....	159, 161, 163, 168	QUOTE .....	296
Numeric MOVE rule.....	161, 163	Record access verbs .....	65
Object change command.....	105	RECORD-LOCKED .....	296
O-OBJECT CHANGE .....	305	RECORD-UPDATED.....	296
OPERATING-SYS-CODE.....	294	Relation data .....	171
OPERATOR-ID .....	294	Relational conditions .....	171
OPTION-NUMBER.....	294	Resequence source code .....	117
OR in a compound conditional .....	177	RESTORE-SCRATCH .....	225
Other Pointer 6 areas.....	217	rich text format	
OUTPUT.....	67	RTF .....	259
PAGE-LENGTH .....	294	R-RELOAD.....	305
PAGE-LINES.....	294	RTF	
Paragraph labels.....	185	description.....	259
Paragraph locate command.....	103	footnotes.....	260
PERFORM.....	185	RTF files	
PERFORM/EXIT.....	187	rules.....	258
PERM.....	217	rules for RTF files.....	258
Physical data structures.....	21	SAVE-SCRATCH.....	225
Pointer 12 PRINT-FIELD .....	223	SAVE-SEGMENT-KEY .....	296
POINTER 44.....	225	Screen Label to Field Name Cross Reference	
POINTER 45 SCRATCH .....	227	.....	137
Pointer 6 numeric fields .....	217	Screen line control .....	71
Pointer 7 .....	219	SCREEN-ERROR .....	296
Pointer 7 command line .....	221	SCREEN-WARNING .....	296
Pointer 7 field definitions .....	289	Segment Layout Report .....	131, 139, 141
Pointer 7 fields in alphabetical order .....	288	Sending and receiving field different format	161,
Pointer 7 fields in pointer order .....	287	167	
Pointer 7 naming conventions.....	286	Sending and receiving field same format.....	159

Sequence line errors.....	89	Using SRTFLD.....	141
Sequential input FILE05.....	129	U—UNDO.....	306
Sequential output FILE10.....	129	W7-01-005.....	296
SESSION-ID.....	297	W7-01-034.....	293
SESSION-STATUS.....	297	W7-01-093.....	296
Sorting cross-reference records.....	145	W7-01-096.....	296
Sorting segment records.....	141	W7-01-097.....	296
SPACE.....	297	W7-01-176.....	297
SPACE-OVER :nn.....	69	W7-01-238.....	292
Special display verbs.....	69	W7-01-239.....	294
Special symbol rules.....	59	W7-01-252.....	290
S—RESEQUENCE.....	305	W7-01-253.....	290
STARTING WITH.....	231	W7-01-298.....	291
STAT-KEY.....	297	W7-01-299.....	291
Subroutines.....	195	W7-01-366.....	296
Support subdirectory		W7-01-398.....	289
location.....	278	W7-01-399.....	292
Symbols.....	183	W7-01-403.....	293
Syntax conventions.....	61	W7-01-428.....	297
System Control Repository edit.....	77	W7-01-429.....	291
System Control Repository update.....	129	W7-01-441.....	290
TEMP.....	217	W7-01-442.....	295
Text body columns.....	83	W7-01-443.....	292
The Copy Utility (COPY).....	119	W7-01-444.....	289
The Display Utility (DISPLY).....	121	W7-01-448.....	289
The Module Code Reference		W7-01-449.....	290
location.....	264	W7-02-000.....	297
The Purge Utility (PURGE).....	123	W7-02-008.....	292
The Solution Series architecture.....	13	W7-02-044.....	293
The Solution Series processing modes.....	41	W7-02-094.....	296
Topic ID (#)		W7-02-252.....	290
description.....	261	W7-02-254.....	294
rules.....	261	W7-02-322.....	295
topic IDs		W7-02-343.....	295
for fields.....	261	W7-02-364.....	295
for forms.....	261	W7-02-368.....	294
Topic Title (\$)		W7-02-370.....	294
description.....	260	W7-03-400.....	294
rules.....	260	W7-03-445.....	293
TRACE-HOLD.....	298	W7-04-280.....	294
TRACE-READY.....	298	W7-04-430.....	289
TRACE-SWITCH.....	298	W7-06-116.....	291
Transfer control.....	185, 191	W7-06-122.....	292
Transfer held range command.....	115	W7-06-240.....	291
Transferring control within a program.....	185	W7-06-246.....	295
TRANS-NUMBER.....	298	W7-06-254.....	292
T—TRANSFER RANGE.....	305	W7-06-316.....	295
Undo command.....	107	W7-06-372.....	293
Usage.....	183	W7-06-391.....	291
Using EXPORT.....	127	W7-06-434.....	289
Using field table list.....	133	W7-07-272.....	289
Using F-SEGM.....	141	W7-09-035.....	294
Using RELOAD.....	91	W7-10-254.....	292

W7-14-078.....	296	Windows help	
W7-15-264.....	289	overview.....	258
W7-19-324.....	295	Windows Help	
W7-19-345.....	295	compilation process .....	263
W7-24-010.....	292	components .....	258
W7-32-046.....	293	Work area access.....	215
W7-48-128.....	290	Working storage and pointers .....	37
W7-61-177.....	294	X—EXECUTE PROGRAM/SCREEN .....	307

# Hewitt

---

## **Cyborg Scripting Language Report Customization Participant's Guide**

**Document Issue: 5.1**

Document Issue Status: Fifth Release  
Document Issue Level: 5.1  
Document Issue Date: April 2004  
Software Version: 5.1

### **Copyright Notice**

Copyright© 2004 by Hewitt Associates LLC. All rights reserved.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series®  
are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear,  
but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

---

## Contents

<b>Contents</b> .....	<b>i</b>
<b>Section 1: Course Overview</b> .....	<b>1</b>
Course introduction.....	3
Course logistics.....	5
Course materials.....	7
<b>Section 2: Report Processing Overview</b> .....	<b>9</b>
Introduction.....	11
Packaged report features .....	13
Report process.....	15
Submitting, routing, and viewing reports.....	33
Using the Enhanced Reporting features.....	47
Section summary.....	65
Section 2 exercise.....	67
<b>Section 3: The Report Extract</b> .....	<b>69</b>
Introduction.....	71
The Report Extract program.....	73
Defining the report.....	75
Report initialization and selection.....	81
The extract record .....	83
Section summary.....	99
Section 3 exercise.....	101
<b>Section 4: Report Detail and Totals - The RTEDIT Form</b> .....	<b>103</b>
Introduction.....	105
Report detail, totals, and printing.....	107
Print positions and the extract record.....	111
Print position and the report output.....	113
Report extract layout.....	117
Compiling report print positions .....	121
Section summary.....	123
Section 4 exercise.....	125
<b>Section 5: Sort Key Options</b> .....	<b>127</b>
Introduction.....	129
Sort key options .....	131
Sort key length and placement .....	141
Section 5 exercise 1.....	145

Special sort techniques.....	147
Section summary.....	155
Section 5 exercise 2.....	157
<b>Section 6: Report Basics Plus .....</b>	<b>159</b>
Introduction.....	161
Run-time input parameters.....	163
Cyborg Scripting Language basics plus.....	171
Section 6 exercise 1.....	181
Segment/Pointer manipulation.....	183
Accessing table data.....	191
Section summary.....	195
Section 6 exercise 2.....	197
<b>Section 7: Reports-Special Functions .....</b>	<b>199</b>
Introduction.....	201
Labor and history records .....	203
Variable report headings .....	209
Summary reporting.....	211
Report special formatting.....	213
Section summary.....	215
Section 7 exercise 1.....	219
Section 7 exercise 2.....	221
<b>Section 8: Introduction to Special Print Options .....</b>	<b>223</b>
Introduction.....	225
Special print option overview .....	227
New organization number (Control 1–2) is starting.....	237
Heading record are ready to print.....	239
Additional heading lines .....	241
Detail data with matching sort key is ready .....	243
Total data is ready .....	245
A detail line is ready to print.....	247
A total line is ready to print.....	249
The report is done .....	251
Section summary.....	253
Section 8 exercise.....	255
<b>Appendix A: Exercise Answers .....</b>	<b>257</b>
Section 2—exercise answers.....	258
Section 3—exercise answers.....	263
Section 4—exercise answers.....	264
Section 5—exercise 1 answers.....	268
Section 5—exercise 2 answers.....	269
Section 6—exercise 1 answers.....	275
Section 6—exercise 2 answers.....	279
Section 7—exercise 1 answers.....	283
Section 7—exercise 2 answers.....	287
Section 8—exercise answers.....	292
<b>Appendix B: Extract Record Files .....</b>	<b>299</b>
Extract record files .....	301
Extract record files—the report extract step .....	301
Extract record files—special print options technique 1 .....	303
Extract record files—special print options technique 2 .....	305

<b>Appendix C: Segment Layout Reports .....</b>	<b>307</b>
E segment—basic employee .....	308
F segment—employee name and address .....	316
G segment—employee labor splits .....	318
H segment—employee HEDs .....	319
H segment—employee HEDs (labor/history record) .....	322
J segment—employee taxes .....	323
J segment—employee taxes (history record) .....	330
L segment—employee spouse/dependent .....	331
L segment—employment status .....	332
L segment—employee salary .....	334
L segment—employee location.....	335
L segment—employee training .....	336
<b>Appendix D: FIND-Verbs and Tables.....</b>	<b>337</b>
FIND- verbs .....	338
HR base module tables.....	339
Benefits tables .....	340
<b>Appendix E: Print Position Compile (RETYPE) Error Resolution.....</b>	<b>341</b>
Print Position Compile (RETYPE) Error Resolution.....	342
<b>Glossary .....</b>	<b>345</b>
<b>Index .....</b>	<b>363</b>

---

**NOTES**

---

## Section 1: Course Overview

---

### Table of Contents

Course introduction.....	3
Course logistics.....	5
Course materials .....	7

## Course Introduction

- **Purpose and benefits**
- **Audience**
- **Prerequisites**
- **Goals**
- **Expectations**

---

NOTES

---

## Course introduction

### **Purpose**

The purpose of this course is to teach you the skills necessary to create packaged report programs.

### **Benefits**

The benefit of learning this information will be your ability to report on data in flexible and useful formats.

### **Audience**

This course has been designed for project team members or technical end-users who intend to create packaged reports.

### **Prerequisites**

Before taking this course you should have completed the following courses:

- Using The Solution Series: Administrative Solutions
- Introduction to Cyborg Scripting Language

### **Goals**

At the conclusion of this course you should be able to:

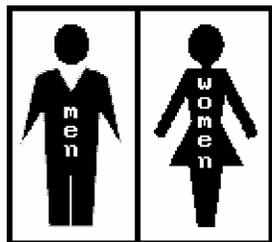
- Describe the packaged report process flow, identifying the report and optional output files.
- Identify the components of an extract record.
- Write and run detail and total reports.
- Use special features and functions in reporting.

### **Expectations**

To achieve the goals of this course you should:

- Ask questions.
- Share examples of your own experiences. This sharing of information among participants enhances the learning process.
- Ask where to obtain additional information if you have an interest in a point that is introduced.
- Ask the instructor to provide additional information during breaks or at lunch.

# Logistics



---

**NOTES**

---

## Course logistics

Use the space below and in the right column to take notes about the course logistics.

### Meals

---

---

---

### Breaks

---

---

---

### Telephones

---

---

---

### Restrooms

---

---

---

### Security

---

---

---

### Questions

---

---

---

# Course Materials



---

**NOTES**

---

## Course materials

### Course materials

All form illustrations are shown for the Windows version of The Solution Series. Instructions to access or complete a form are provided using Windows.

### Table of contents

Each section has a table of contents listing on the section title page.

### Text layout

This guide is designed in the following manner:

- Left-side pages typically contain copies of overhead transparencies or forms.
- Right-side pages contain information about the overhead transparency or form and an area for your notes.

### Section exercise

Exercises give you an opportunity to practice what you have learned in each section. All sections except the course overview section have exercises.

### Appendices

The appendices are in the back of your participant guide. Appendices contain:

- Exercise Answers—Answers to section exercises
- Extract record files
- Segment layout reports
- FIND-verbs and tables
- Print position compile (RETYPE) error resolution

### Glossary

Glossary and syntax for the Cyborg Scripting Language verbs.

### Index

An alphabetical listing of content cross-referenced to page numbers.

---

**NOTES**

---

# Section 2: Report Processing Overview

---

## Table of Contents

Introduction.....11  
Packaged report features .....13  
Report process.....15  
Submitting, routing, and viewing reports .....33  
Using the Enhanced Reporting features.....47  
Section summary.....65  
Section 2 exercise .....67

## Objectives

- **Describe the report process flow**
- **Identify the report process files**
- **Schedule, run, and view a packaged report**

---

### NOTES

---

## Introduction

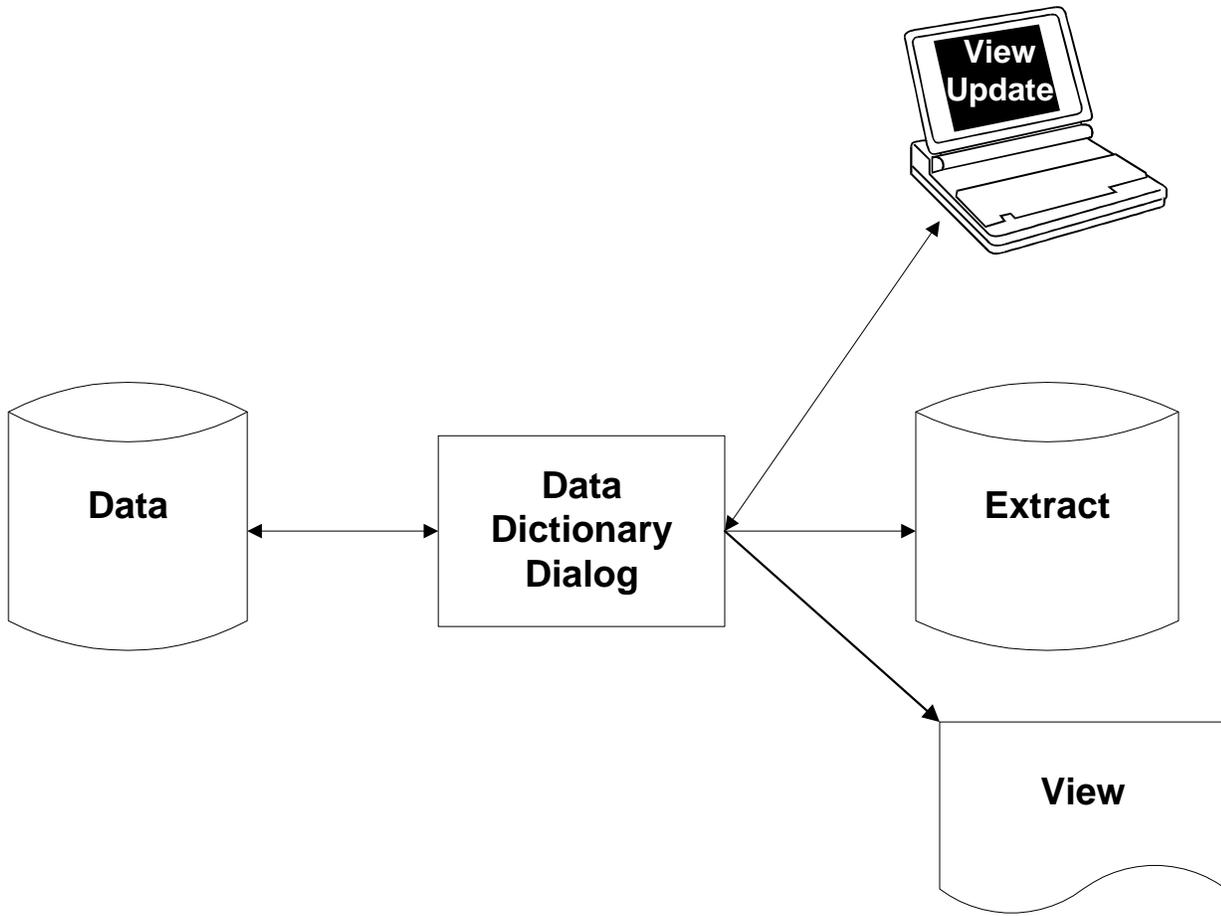
### **Purpose**

In this section, you will learn about the packaged report processes and execution.

### **Objectives**

Upon completion of this section you will be able to:

- Describe the report process flow
- Identify the report process files
- Schedule, run, and view a packaged report



---

**NOTES**

---

## Packaged report features

Using Cyborg Scripting Language (CSL) to create a report program allows you to incorporate many valuable features. Some of these features are generic to all CSL programs.

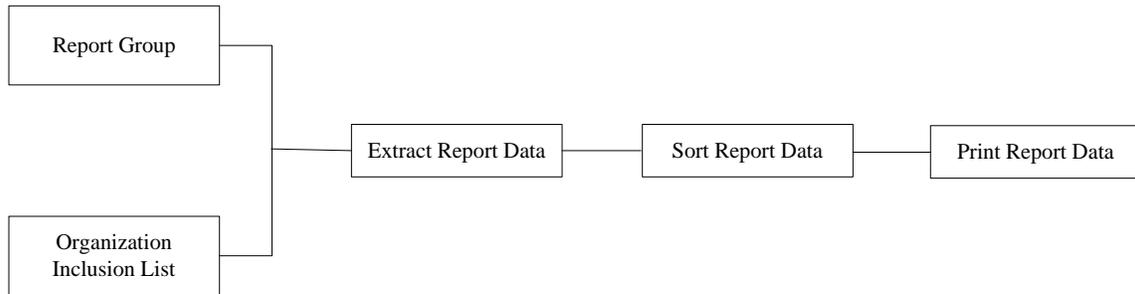
Features specific to packaged reports include:

- Default headings
- User defined title and column headings
- Automatic reading of Company and Employee records
- Multiple output queues
- Run-time input parameters
- User Defined Sorts
- Supplemental print logic (optional)

Features generic to all CSL programs:

- Exhaustive use of Data Dictionary
- Formula calculation logic
- Debugging facility
- Platform independence

# The Solution Series Packaged Reporting



---

## NOTES

---

## Report process

### Report steps

The Solution Series packaged reporting procedure includes the following four steps. These steps must be completed to produce your reports.

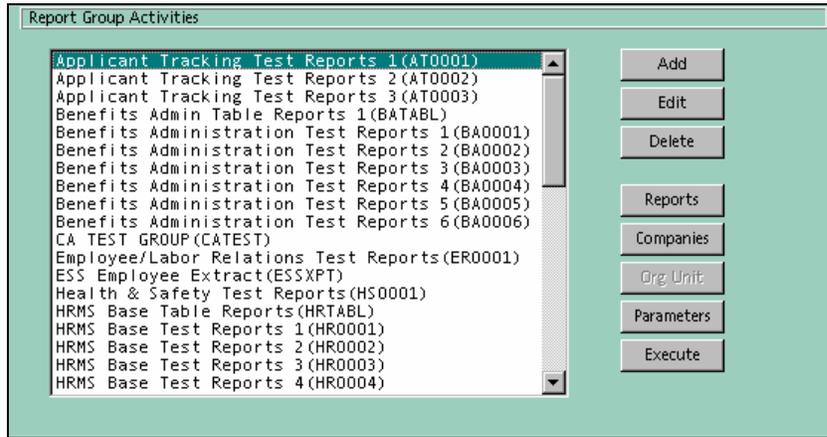
1. Report scheduling
2. Report extract
3. Extract sort
4. Report print

### Report development

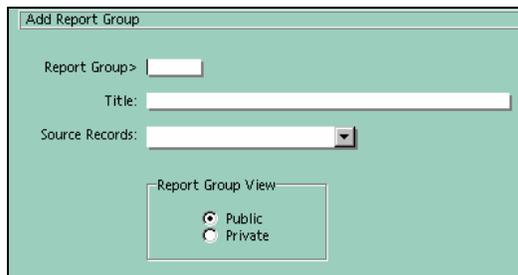
CSL packaged report program development includes the following four components:

1. Report extract program
2. Report extract/print layout(s)
3. Input Parameters form
4. Optional supplemental print program

# Report Groups (RGMSTR)



# Adding a Report Group (RGADD)



---

## NOTES

---

## Report process, continued

### Establish a report group

Before starting the report process, the Report Group must be established. The Report Group Activities form (RGMSTR) displays all available report groups. Multiple operations in report groups can be performed from this form. Reports may be run individually or be grouped for background processing. A Report Group may contain up to 20 different reports.

Make the following selections from the Navigator

**Component:**  Reporting  
**Process:** Report Scheduling  
**Task:**  Schedule Report Groups

**Result:** The Report Group Activities form displays.

### Adding a report group

- **Add button**—The Add button on the Report Group Activities form (RGMSTR) is used to launch the Add Report Group form (RGADD). This form is used to add a new report group. In order to add a new report group complete the following steps:
  1. Type a six-character report group name
  2. Type a title for the report group name
  3. Select the type of source records
  4. Select a report group view
  5. Press Enter

## Edit Report Group form (RGCHG)

Edit Report Group BA0001

Title: Benefits Administration Test Reports 1

Source Records: Employee Master

Report Group View

Public

Private

## Report Group Contents form (RPTGRP)

Report Group Contents Of Benefits Administration Test Reports 1 BA0001

Reports In Group (Double Click Report To Delete)

- 2E-RPT Adp Earnings Report
- 40-RPT Calc Credited Serv Granted
- 41-RPT Calc Vesting Earned
- 42-RPT Vested Employees By Percent Vested
- 48-RPT Retiree Payment Stop Date Passed
- 49-RPT Retiree Deceased
- 4A-RPT Welfare Benefits Register

Available Reports (Double Click Report To Add)

- Applicant Tracking
- Benefits Admin
- EEO/AAP
- Emp/Labor Relations
- HRMS Base
- Health & Safety

---

## NOTES

---

## Report process, continued

### Editing a report group

- **Edit button**—The Edit button on the Report Group Activities form (RGMSTR) is used to launch the Edit Report Group form (RGCHG). The Edit Report Group form (RGCHG) is used to change the title, source records, or view of the report group. In order to edit a report group, select it on the Report Group Activities form (RGMSTR) and click the Edit button. The following updates can be made:

1. Type any title changes to the report group
2. Select the type of source records used for the report group
3. Select a report group view
4. Press Enter to complete any updates to the report group

### Deleting a report group

- **Delete button**—The Delete button on the Report Group Activities form (RGMSTR) is used to delete a selected report group. In order to delete a report group, select it on the Report Group Activities form (RGMSTR) and click the Delete button.

### Adding reports to a report group

- **Reports button**—The Report button on the Report Group Activities form (RGMSTR) is used to launch the Report Group Contents form (RPTGRP). This form is used to add and remove reports from a report group. In order to add reports to a group complete the following steps on the Report Group Contents form (RPTGRP):

1. Select an Available Report module to display the associated reports.
2. Select a report name from Available Reports to add it.
3. Press enter.

## Limit Companies form (RGLC12)

Company ID	Company Name	Organization Type
<input type="checkbox"/>	01JENN Hudson Systems, Inc.	
<input type="checkbox"/>	991111 ACME CE/H ACCUMLATORS	CE/H ACCUMULATION ORGANIZATION
<input type="checkbox"/>	993333 ACME MANUFACTURING	APPLICANT TRACKING CTL 1-2
<input type="checkbox"/>	995555 ACME RETIREES	RETIREE ORGANIZATION
<input type="checkbox"/>	997777	
<input checked="" type="checkbox"/>	999999 ACME MANUFACTURING	PRODUCTION MFTG ORGANIZATION

## Position Management Report Group Parameters form (RUN-PM)

Report Group>

---

### NOTES

---

## Report process, continued

### Selecting an organization

- **Companies button**—The Companies button on the Report Group Activities form (RGMSTR) is used to launch the Limit Companies form (RGLC12). This form is used to establish the organizations used for the group reports. In order to limit the organization levels, complete the following steps:
  1. Select the companies to be included in the group reports.
  2. Press Enter.

### Entering report group parameters

- **ORG Units button**—The ORG Units button on the Report Group Activities form (RGMSTR) is used to launch the Position Administration Report Group Parameters form (RUN-PM).

This form (RUN-PM) is used to configure the report group and launches the Parameter Selection form (RPARMS). In order to configure the report group (Position Administration reports are processed by organization level rather than by company) complete the following steps:

1. Select the organization levels to be included.
2. Press Enter.

## Parameter Selection form (RPARMS)

Parameter Selection For Benefits Administration Test Reports 1 BA0001

2E-RPT	Adp Earnings Report	<input type="checkbox"/>	Set Parameters
40-RPT	Calc Credited Serv Granted	<input type="checkbox"/>	Set Parameters
41-RPT	Calc Vesting Earned	<input type="checkbox"/>	Set Parameters
42-RPT	Vested Employees By Percent Vested	<input type="checkbox"/>	Set Parameters
48-RPT	Retiree Payment Stop Date Passed	<input type="checkbox"/>	Set Parameters
49-RPT	Retiree Deceased	<input type="checkbox"/>	Set Parameters
4A-RPT	Welfare Benefits Register	<input type="checkbox"/>	Set Parameters
4B-RPT	Plan Participation Register	<input type="checkbox"/>	Set Parameters
4C-RPT	Plan Beneficiary Listing	<input type="checkbox"/>	Set Parameters
4D-RPT	Plan Dependent Listing	<input type="checkbox"/>	Set Parameters

## Parameter Entry form (RPTARG)

Report Parameters For Adp Earnings Report 2E-RPT

Report Group - Benefits Administration Test Reports 1 BA0001

ADP Date: 12-31-1989

Plan ID: 500

## Execute Report Group form (RGEXEC)

Execute Report Group

AT0001 Report submitted successfully.

Launch CYBPRT to view output listing(s)

---

## NOTES

---

## Report process, continued

### Setting report parameters

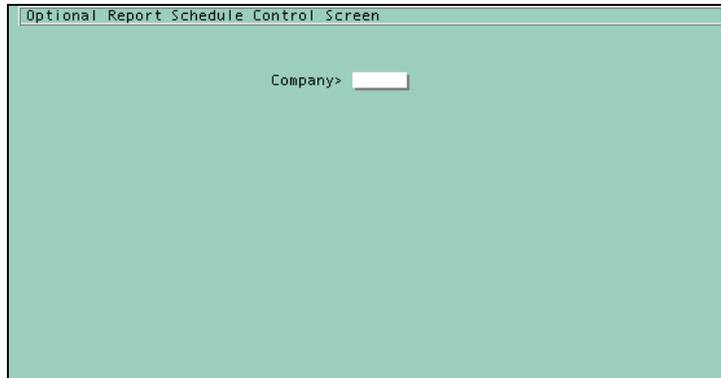
**Set Parameters button**—The Set Parameters button on the Parameter Selection form (RPARMS) allows you to select any report listed for that particular group. The parameters for a report are displayed on the Parameter Entry form (RPTARG).

To launch the Parameter Entry form (RPTARG) and view any report parameters, click the Set Parameters button next to the desired report. If you plan to update any report parameters complete the following steps:

1. Type in the required parameters.
2. Press Enter to return to the parameters form.

**Execute button**—The Execute button on the Report Group Activities form (RGMSTR) allows you to execute a report group. The Execute Report Group form (RGEXEC) will appear if the report group was submitted successfully.

# Optional Valid Reports for Organization (C12RPT)



---

## NOTES

---

## Report process, continued

### Organization report validation

The Optional Report Schedules Control form (C12RPT) is an optional feature of the report process. Using this form, you can specify which reports are valid for each organization on your Employee Database (FILE02). The form is set up by organization number.

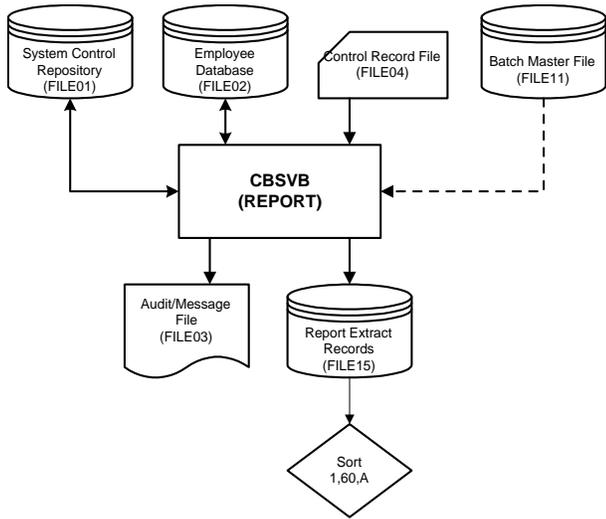
Make the following selections from the Navigator

- Component:**  Reporting
- Process:** Report Scheduling
- Task:**  Specify Reports for Organization

**Result:** The Optional Report Schedule Control form (C12RPT) displays.

- The Report Extract (REPORT) assumes all reports are valid and will be produced for each organization, unless a Report Schedule Control form (C12RPT) is entered.
- When a Report Schedule Control form (C12RPT) is present, the Report Extract (REPORT) verifies that each report is valid before allowing it to be produced for the organization.

# Report Extract (REPORT)



**Input Files:**  
 FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File  
 FILE11 Batch Master File (optional)

**Execute:** CBSVB  
**Output files:** FILE03 Audit/Message File  
 FILE15 Report Extract File

## REPORT Control Record:

Report parameters for the first 50 employees in the WEEKLY schedule:

```

    1 1 2 2 3 3 4 4 5 8
    ....5....0....5....0....5....0....5....0....5....0//.0
    REPORT WEEKLY F
  
```

Report parameters for all employees in the WEEKLY schedule:

```

    1 1 2 2 3 3 4 4 5 8
    ....5....0....5....0....5....0....5....0....5....0//.0
    REPORT WEEKLY
  
```

Report parameters using optional FILE11 for the WEEKLY schedule:

```

    1 1 2 2 3 3 4 4 5 5 8
    ....5....0....5....0....5....0....5....0....5....0....5/0
    REPORT WEEKLY F11
  
```

---

## NOTES

---

## Report process, continued

### Report extract

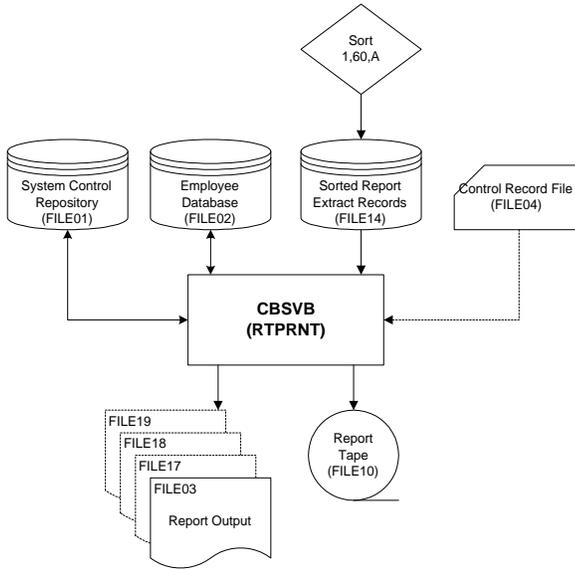
The Report Extract program (REPORT) uses the schedule of reports and organizations setup via the Report Group Activities form (RGMSTR) and the Report Schedule Control (C12RPT) form parameters. REPORT executes each report program to create report extract records for the scheduled reports.

### Using REPORT

To execute REPORT, the FILE04 Control Record is:

	<u>Position</u>	<u>Option/Step/Description</u>
<b>Program Field</b>	23	Type REPORT.
<b>Key Field</b>	31–36	Type the name of the Report Group.
<b>Additional Key Field</b>	41–44	Optionally, limit the number of employees included on the report by typing an 'F' to process the FIRST 50 employees. <i>Note: Enter a three-digit parameter for increments other than 050 (for example, F100).</i>
	52	Blank for normal reporting, 'R' for Roll-up or 'C' for Consolidated reporting.
	53–55	Type 'F11' to produce reports using a Payroll Batch Master (FILE11) instead of FILE02. Or Type 'Qnn' where nn is the Alternate Key id if you want your report based on Alternate Key selection.

# Report Print (RTPRNT)



**Input Files:** FILE01 System Control Repository

FILE02 Employee Database

FILE04 Control Record File

CBSVB

**Execute:**

**Output files:**

FILE03 Report Print File

FILE10 Optional Tape Report

FILE17 Alternate Print File

FILE18 Alternate Print File

FILE19 Alternate Print File

## RTPRNT Control Record:

```

    1 1 2 2 3 3 4 4 5 8
    ...5...0...5...0...5...0...5...0...5...0...5...0...//.0
  
```

RTPRNT

---

## NOTES

---

## Report process, continued

### Extract sort

The sort step prepares the extracted report records for the final step of The Solution Series reporting (RTPRNT). RTPRNT produces the actual reports.

SORT is not a Solution Series process; it is an Operating system utility that must perform the following functions:

- Read the report extract records (FILE15).
- Sort each record based on the sort key starting in position 1 for a length of 60 in ascending sequence.
- Write the sorted report extract records to FILE14. The output from the sort will be read into the report phase.

### Report print

The Report Print program (RTPRNT) is the final step in report generation.

- The Report Print program (RTPRNT) reads the sorted extract records (FILE14).
- The data on the extract records is then reformatted and written to the report output file (either FILE03, FILE17, FILE18, or FILE19).

### Using RTPRNT

To execute RTPRNT, you must provide a control record as FILE04.

	<u>Position</u>	<u>Option/Step/Description</u>
<b>Program Field</b>	23–28	Type 'RTPRNT'

# Batch Report Output

## FILE03 from REPORT Phase

```
-----
CSSS <0002( (999999(REPORT( (1A-R      (          )10:24:44 09-11 XXXX
-----
----COMPLETE----
```

## FILE 03 , FILE17, FILE18 OR FILE19 from RTPRNT Phase

CORPORATION 99	ACME MANUFACTURING	ALPHABETIC LISTING OF ACTIVE				REPT	PAGE 1		
DIVISION 9999	PRODUCTION CTL 1-2	AND INACTIVE EMPLOYEES				1A-R	TIME 10:25:59 DATE 09-11-2002		
EMPLOYEE-NAME	EMPLOYEE NUMBER	CTRL THREE	CTRL FOUR	CTRL FIVE	CTRL SIX	MAIL DISTRIBUTION	HIRE DATE	EMPLOYEE STATUS	
ADAMS, RICHARD	1117	3030	4040	5050	6060	2FLR-22W	2/19/1984	01 ACTIVE-SALRD-REG-F/T	
ALSON, GEOFFREY	3003	3030	4040	5050	6060	2FLR-22W	2/19/1984	01 ACTIVE-SALRD-REG-F/T	
ANDERSEN, KARI	6004	3333	4444	5555	6666	CORP-STENO	5/16/1994	01 ACTIVE-SALRD-REG-F/T	
ANDERSON, DANIEL M	1616	3333	4444	5555	6666	CORP-STE3	4/24/1984	01 ACTIVE-SALRD-REG-F/T	
ANDREWS, HENRY A.	2013	3388	4488	5555	6666	RC-212	3/9/1986	01 ACTIVE-SALRD-REG-F/T	
AUSTIN, STEVEN	1234	3030	4040	5050	6060	B2-STN5	12/17/1978	0* RESERVED VALUE	
BALDWIN, ALICE A.	1043	3333	4444	5555	6666	5FLR-12W	8/20/1983	03 ACTIVE-HOURLY-REG-F/T	
BARNES, JOHNSON	2002	3388	4488	5508	6608		5/1/1985	01 ACTIVE-SALRD-REG-F/T	
BARTHOLOW III, JON	1113	3333	4444	5555	6666	6FLR-8E	3/12/1982	01 ACTIVE-SALRD-REG-F/T	
BEACHEM, JUDITH	6009	3030	4040	5050	6060	B2-STN10	9/26/1983	01 ACTIVE-SALRD-REG-F/T	

## NOTES

## Report process, continued

### Print files

The Solution Series reporting output files (print files) are the following 132 character sequential files:

- **FILE03** when the Forms Code is 1
- **FILE17** when the Forms Code is 2
- **FILE18** when the Forms Code is 3
- **FILE19** when the Forms Code is 4

☞ *Refer to Section 3: The Report Extract for more detail about the Forms Code.*

### Error messages

Errors in the report process are displayed in FILE03. These messages include:

#### REPORT phase

**\*\*\*RUN REPORT LIST IS MISSING\*\*\*** This message occurs when the Report Group name in the control record is invalid.

**\*\*\*NO REPORTS FOUND\*\*\*** This message occurs when the extract step has produced no output due to an invalid control record setup.

#### RTPRNT phase

**XX-R Report Bypassed - No RTEDIT Records** No RTEDIT records have been established for the report.

☞ *Refer to Section 4: Report Detail and Totals—The RTEDIT Form for additional information about the RTEDIT form.*

**XX-R Record Bypassed - Bad Record Type** Invalid RTEDIT Record Type.

☞ *Refer to Section 3: The Report Extract for additional information about record types and Section 4: Report Detail and Totals—The RTEDIT Form for additional information about the RTEDIT form.*

## **Submitting, routing, and viewing reports**

- **Batch Job Initiation (SUBMIT)**
- **View Held Reports (VIEW)**

---

### **NOTES**

---

## Submitting, routing, and viewing reports

### Submitting, routing, and viewing reports

The Solution Series allows you to submit, route, and view reports or queries from an online session.

The programs used in this process are:

- Batch Job Initiator (SUBMIT)
- View Held Reports (VIEW)

### Initiate reports

The Batch Job Initiator program (SUBMIT) provides a series of forms that will:

- Provide the QUERY and REPORT fields for creating the Control Record. The control record is required for the background execution of The Solution Series.
- Dependent on your operating system, either execute the script or display a message indicating the script to submit after logging off The Solution Series.

### Route reports

You have the option of routing the report output for online viewing or printing.

### View reports

The VIEW program allows you to view the output from a background execution of a report or query that was started through the SUBMIT process if online viewing was selected.

## Batch Job Initiation (SUBMIT)

```
SELECT OPTION ==> 1
BATCH JOB INITIATION MENU
1 Reports
2 Query
```

---

### NOTES

---

## Submitting, routing, and viewing reports, continued

### Batch job initiator

The Batch Job Initiator program (SUBMIT) will initiate a script for a query or report process by performing the following functions:

- Creates control records to direct the background process to the specific report group or query to execute.
- Either executes the script for the requested report group or query or displays a message directing you to submit a script once you sign-off The Solution Series.

### Initiate a report

The SUBMIT program's first form prompts you for the type of report you wish to execute, a query or report. To execute SUBMIT and initiate a report perform the following steps:

Tools ► Reporting ► Initiate Scheduled Reports

**Result:** The Batch Job Initiation Menu form (SUBMIT) is displayed.

#### Select Option

1. Type 1
2. Press ENTER

## Batch Job Initiation (SUBMIT)

REPORT BATCH JOB INITIATOR

Enter Report Group name:

Hold output for online review?  (Enter Y or N)

Normal, roll-up, or consolidate?  (Enter space, R, or C)

---

### NOTES

---

# Submitting, routing, and viewing reports, continued

## Initiate a report

The report batch job initiator is the second form in the report initiation process. The fields on this form will be used to create the control record and determine the routing of the report output.

To initiate the report perform the following steps:

<b>Selection:</b>	<b>Step:</b>
<hr/> <b>Report Group</b>	1. Type the name of the report group to be processed.
<b>Hold Output for online review?</b>	2. Type 'Y' to route the report output to the Employee Database (FILE02) for online viewing.
	OR
	Type 'N' to route the report output to the Audit/Message/Report File (FILE03) or Alternate Print File (FILE17, FILE18, or FILE19).
	3. Press ENTER.

# Batch Job Initiation (SUBMIT)

## External Batch Job Submission

```
REPORT BATCH JOB INITIATOR
LOG OFF TO SUBMIT JOB JRPTXXX
```

## The Solution Series Batch Job Submission

```
REPORT BATCH JOB INITIATOR
JOB JRPTXXX  SUBMITTED
```

---

## NOTES

---

## Submitting, routing, and viewing reports, continued

### Initiate a report, continued

When you have completed all entries in the Report Batch Job Initiator form (SUBMIT) and pressed Enter, the submit process will perform one of the following:

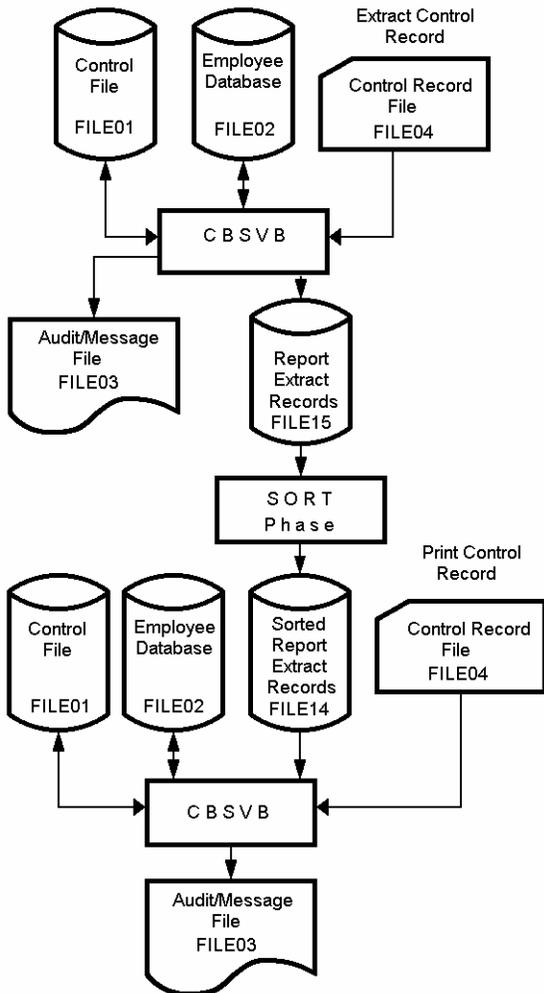
- Display the message 'JOB JRPTxxxx SUBMITTED', and submit the script to execute the Report process.

Or

- Display the message 'LOG OFF TO SUBMIT JOB JRPTxxxx', which requires you to log off The Solution Series and submit the job stream JRPTxxxx.

JRPTxxxx is a predefined script that will execute the REPORT process, where xxxx is the OPERATOR-ID of the user who initiated the script.

# JRPTxxxx - Report Batch Initiator



**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File

**Execute:** CBSVB  
**Output files:** FILE03 Audit/Message File  
 FILE15 Report Extract Records

## JRPTxxxx Extract Control Record:

```

    1 1 2 2 3 3 4 4 5 8
    1...5...0...5...0...5...0...5...0...5...0...5...0...//.0
    
```

F04F01 <CTRLRT xxxxA

**Input Files:** FILE01 System Control Repository  
 FILE02 Employee Database  
 FILE04 Control Record File  
 FILE14 Sorted Extract Records

**Execute:** CBSVB  
**Output files:** FILE03 Audit/Message File

## JRPTxxxx Print Control Record:

```

    1 1 2 2 3 3 4 4 5 8
    1...5...0...5...0...5...0...5...0...5...0...5...0...//.0
    
```

F04F01 <CTRLRT xxxxD

## NOTES

## Submitting, routing, and viewing reports, continued

### Initiate a report, continued

The JRPTxxxx script must be established prior to executing the submit process. The processing flow of the script will follow the report steps of extract, sort, and print.

The JRPTxxxx script requires two control records, one for the extract step and one for the print step.

<b>Extract Step</b>	<u>Position</u>	<u>Option/Step/Description</u>
<b>Program Field</b>	23–28	Type 'F04F01'
<b>Key Field</b>	31–40	Type '<CTRLRT'
<b>Additional Key Field</b>	41–44	Type the 4 character OPERATOR-ID of the user that will use this job stream
	45	Type 'A'
<b>Print Step</b>	<u>Position</u>	<u>Option/Step/Description</u>
<b>Program Field</b>	23–28	Type 'F04F01'
<b>Key Field</b>	31–40	Type '<CTRLRT'
<b>Additional Key Field</b>	41–44	Type the 4 character OPERATOR-ID of the user that will use this script
	45	Type 'D'

# View Held Reports

```
ACTION ==>  D-Delete, P-Print, V-View ID ==> 030
ID# PGS DATE TIME
030 0008 04-26-1999 14:32:20
```

---

## NOTES

---

## Submitting, routing, and viewing reports, continued

### Viewing report and query output

The View Held Report program (VIEW) allows you to view, print, and/or delete the packaged reports routed for online viewing from the SUBMIT process.

Tools ► Reporting ► View Held Report

**Result:** The View Held Report form (VIEW) is displayed.

### View held report

The View Held Report program (VIEW) displays a list of all the reports that have been routed for online viewing based on your OPERATOR-ID.

- Each report is assigned an ID# that is used to identify each separate submission. The ID# is used to view, print, or delete.
- Other information displayed about each report includes:
  - PGS—The number of pages for each report group.
  - DATE—The date the reports were created.
  - TIME—The time the reports were created.

# View a Held Report

```
ACTION ==> V D-Delete, P-Print, V-View ID ==> 030
ID# PGS DATE TIME
030 0008 04-26-1999 14:32:20
```

```
SIDE ==> L-Left, R-Right PAGE ==> 0002 LINE ==> 01 Pages=0008
CORPORATION 99 ACME MANUFACTURING BIRTHDAY LISTING
DIVISION 9999 PRODUCTION MFTG ORGANIZATION AS OF 01-01-1999
BIRTH BIRTH CTL CTL
MONTH DAY EMPLOYEE NAME THREE FOUR
FEBRUARY 01 ADAMS, RICHARD 3030 4040
FEBRUARY 01 ALSON, GEOFFERY 3030 4040
*SAVE-MONTH 02 2
```

---

## NOTES

---

# Submitting, routing, and viewing reports, continued

## View a report

To view a report, perform the following steps:

<u>Selection:</u>	<u>Step:</u>
<b>Action</b>	Type 'V'
<b>ID</b>	Type the three digit ID# of the report group to be viewed Press ENTER
	<b>Result:</b> The report is displayed.

## Report navigation

Three fields are available for navigating the report:

- **SIDE**—Used to view the left (L) or right (R) side of the report.
- **PAGE**—Used to display a particular page of the report.
- **LINE**—Used to display a page starting at a specific line number.

## Report navigation example

To view the right side of the report, perform the following:

<u>Selection:</u>	<u>Step:</u>
<b>SIDE</b>	Type R and press Enter.
	<b>Result:</b> The right side of the page is displayed on the computer.

## Using the Enhanced Reporting Features

- **Online reporting**
- **Methods for launching reports**
- **Online process monitoring**
- **Online viewing of completed reports**
- **Report disposition**

---

### NOTES

---

## Using the Enhanced Reporting features

### Online reporting

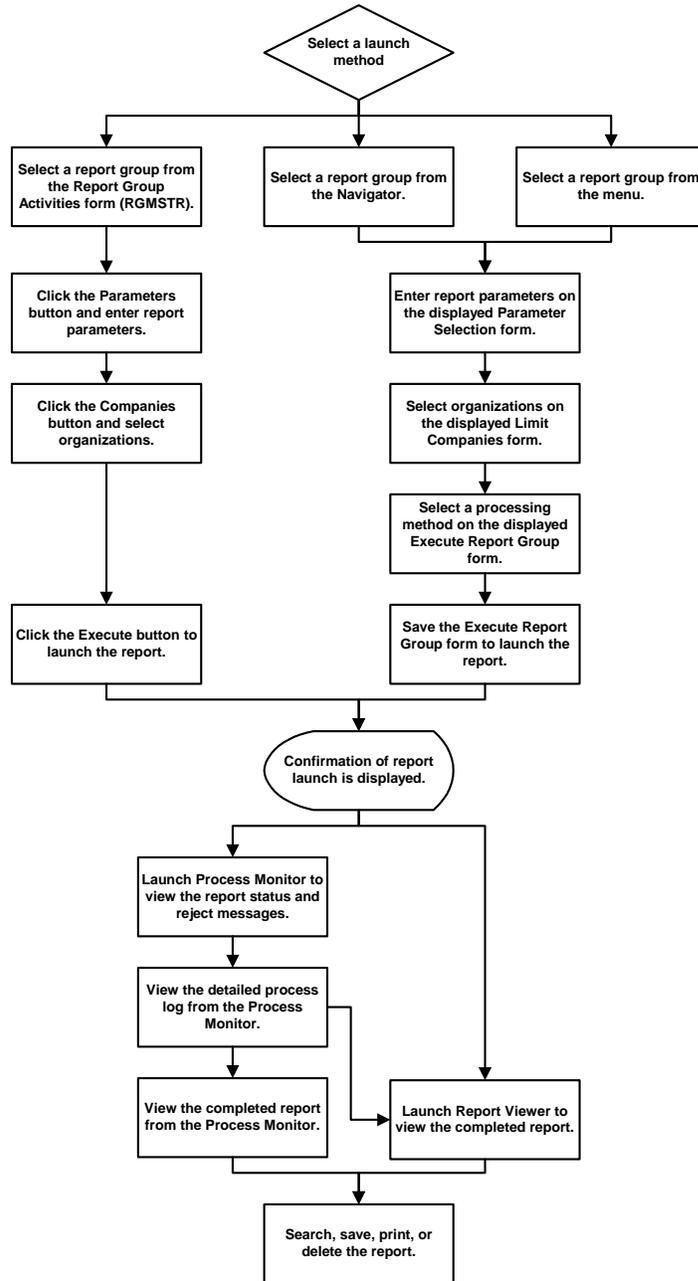
The online reporting process does not replace the Batch Job Initiation Menu form (SUBMIT) or the View Held Report form (VIEW). Instead, it is an additional process that allows you to launch, monitor, and view reports online.

You will be able to run reports as a background process on the server. You will have a choice of running these reports directly from the Report Group Activities form (RGMSTR), from the menus and Navigator, or from checklists.

To supplement this process, the Process Monitor utility allows you to monitor the status of any report that is running. Similar to the View Held Report form (VIEW), the Report Viewer utility displays the completed report in a customizable window. You can print a report, and you can search and jump within reports. You can also save and delete reports.

To help you identify reports that your organization may wish to run, you can still view the details and an example of each delivered report by using the report info icon in the Navigator.

# Enhanced Online Reporting



## NOTES

## Using the Enhanced Reporting features, continued

### The online reporting process

In order to process a report you can launch it from the Navigator, menu, or the Report Group Activities form (RGMSTR). The following describes the steps used in the online reporting process.

- In order to select a report group from the Navigator or menu:
  - Enter the report parameters on the Parameter Selection form
  - Select organizations on the Limit Companies form
  - Select a processing method on the Execute Report Group form
  - Save the Execute Report Group form to launch the report
- In order to select a report group from the Report Group Activities form (RGMSTR):
  - Click the parameters button and enter report parameters
  - Click the companies button and select organization
  - Click the execute button to launch the report

Once the steps are completed, confirmation of the report launch is displayed. Launch the Process Monitor in order to view the report.

### Process to view a report

Once confirmation of the report launch is displayed, access the completed report from either the Process Monitor or the Report View. The following describes the steps used to view completed reports:

- Launch the Process Monitor in order to:
  - View the report status
  - View reject messages
  - View the detailed process log
  - View the completed report
- Launch the Report Viewer in order to:
  - View a completed report
  - Search
  - Save
  - Print
  - Delete a report

# Launching a Report Group with the Navigator or Menu

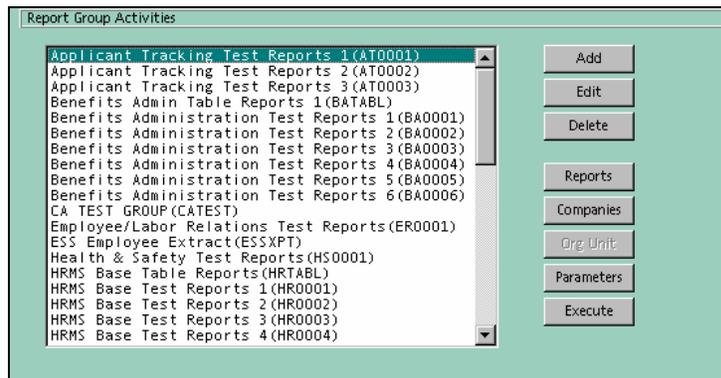
Navigator



Menu



# Launching a Report Group with the Report Group Activities form (RGMSTR)



## NOTES

## Using the Enhanced Reporting features, continued

You may launch a report group with the Navigator, the menu, or from the Report Group Activities form (RGMSTR). The following topics explain each of these methods:

### Using the Navigator or menu to launch a report

All delivered report groups are available by clicking the Reporting component of the Navigator.

You may access the same options on the menu by selecting:

Tools ► Reporting

After you select a report group from the Navigator or menu, the Parameter Selection form (RPTPRM) is displayed first.

If a report does not have any parameters, you may bypass the form by pressing Enter.

If there are multiple reports in a group, after each form is saved the Parameter Selection form is displayed repeatedly so that you may enter parameters for the next report in the group.

After setting parameters for all reports in the group, press Enter to select the companies for which reports should be printed.

Using the Limit Companies form (RPTLC), you can select the organizations to be included in the report.

The last form displayed is the Execute Report Group form (RPTSCR). You must select an option from the Processing Method option list (SC57) and save the form by pressing Enter to launch the report.

A second form (RPTRUN) confirms that the report has been successfully submitted and instructs you to use Report Viewer to see the output.

### Using the Report Group Activities form (RGMSTR) to launch a report

☞ Refer to *Report Process, Establish a report group*, in this section, for instructions on launching a report group with the Report Group Activities form (RGMSTR).

# Launching a Report with checklists



---

## NOTES

---

## Using the Enhanced Reporting features, continued

### Using checklists to launch a report

Checklists can now link report groups, as well as forms, dialogs, and other checklists within The Solution Series to guide you through a business process. Checklists can be added to and accessed from the Bookmarks menu, Favorites toolbar, or the menus and Navigator.

Only report groups that are defined on the menus can be included in a checklist.

Two examples of checklists that include report groups are the payroll checklists that are delivered in The Solution Series, one for the United States and one for Canada.

### Online process monitoring

You may find it helpful to monitor the progress of a report run. If a report is expected to produce a very large output file, you may be able to determine and resolve significant errors before the report process ends.

Using the Process Monitor utility, you will be able to determine which processes and reports have been completed. You can also view and print the completed reports from the Process Monitor.

### Launch toolbar access

You can access the Process Monitor from the Launch toolbar. If the Launch toolbar is not displayed, you may view it by selecting from the menus:

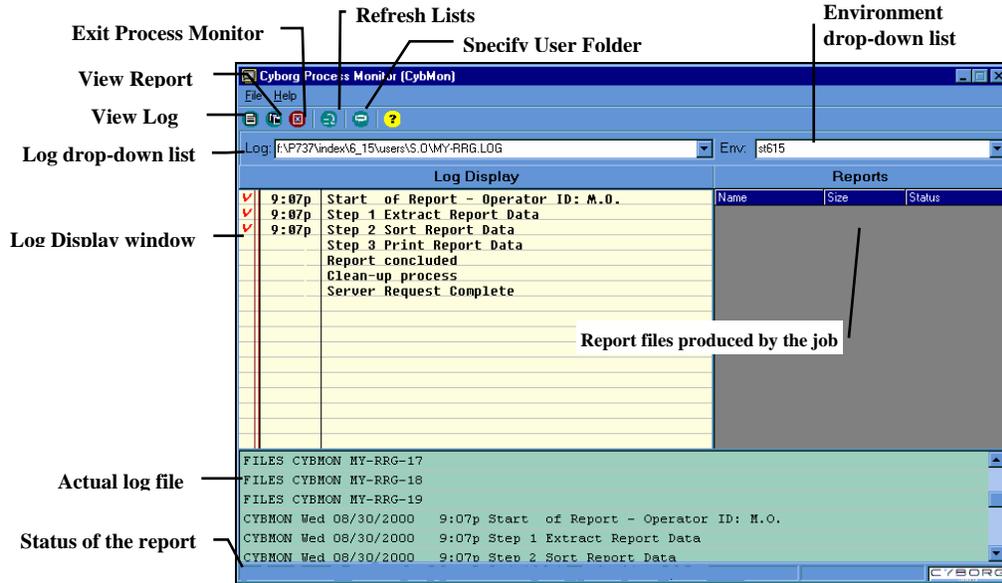
**V**iew ► **C**ustomize **T**oolbars

You may then select Launch from the list. The following toolbar will be displayed on your desktop:



*Note:* Your Launch toolbar may also display other applications that you use with The Solution Series.

# The Process Monitor



## NOTES

## Using the Enhanced Reporting features, continued

### The Process Monitor utility

The Process Monitor utility allows you to view the progress of a report while it is running. It also stores logs of previously run reports.

With this utility, you can set up and select processes by 'environment' or by user. Select the Specify a User Directory per Environment button on the toolbar to specify environments.

As delivered, the Process Monitor will refresh every five seconds. To manually refresh the display click the Refresh Lists button.

*Note: If you have restricted access to the server, you may need to contact your technical staff to change environments.*

*Note: The Specify Environment Folder and Specify User Folder dialogs will display the first time the Process Monitor utility is run. Consult your technical staff if you do not know the correct locations.*

The Log drop-down list contains a list of all report logs located in your user folder. The log display is similar in appearance to a checklist. The scrolling window displays the latest events in the log.

The Env drop-down list is used to select which environment the Process Monitor uses. This field is populated automatically.

The Log Display window is used to display all the steps within the script, as well as the status and completed time of each step. A checkmark appears beside completed steps. These steps are pre-defined within the script.

The Reports area displays the log (LIS) files produced by the script. The files are predefined by the report and its associated scripts. The size and the status of the log files are also shown here.

This list is sorted by clicking any of the title columns. For example, click the Name title bar to sort the files alphabetically by name.

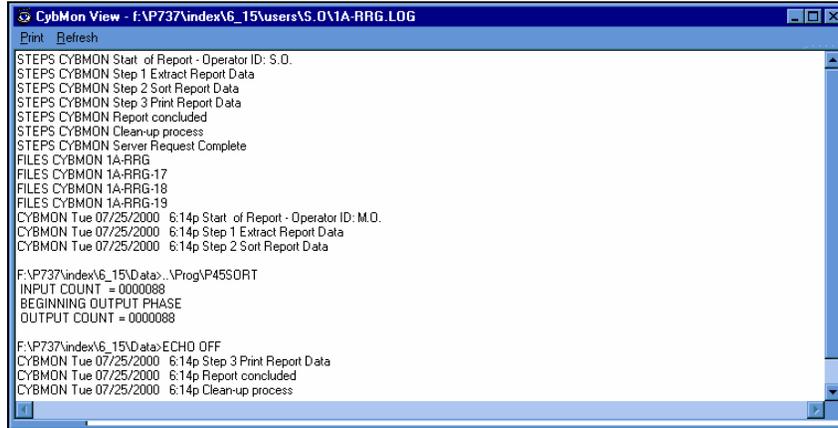
You may double-click the report or click the View Report button on the toolbar to view the report with the Report Viewer utility. Only files with a 'Complete' status can be accessed using the Report Viewer utility.

Active logs, or logs for reports that are still running, are automatically assigned an LIW extension. When complete, the report is automatically renamed with an LIS extension. These extensions help the Process Monitor determine which reports are complete and which ones are still in progress.

The bottom part of the Process Monitor window displays the actual log file. You may scroll through the log using the scroll bar or view the log in its own window by clicking the View Log button on the toolbar. Clicking this button accesses a separate window that can be resized, refreshed, and printed.

If the Process Monitor finds a **\*\*\*REJECT\*\*\*** message anywhere in the log file, it assumes an error and displays the message with the following line in red. The status bar also contains a blinking warning message.

# The Process Log



---

## NOTES

---

## Using the Enhanced Reporting features, continued

### View the process log

You can view the process log in three different ways. First, the checklist in the Log Display window is a display of the log by steps. A checkmark and time stamp will signal the completion of the step.

After all the steps have been completed, the status bar at the bottom of the process log will display 'Completed'.

The second method to view the process log is the scroll window on the bottom section of the Process Monitor. This window will permit you to scroll through the file looking for messages or errors. Errors will be displayed in red. There will also be a red error message flashing **\*\* Warning Errors \*\*** on the status bar located at the bottom of the Process Monitor.

The third method to view the process log is to display it in a separate window. Display the entire log by using the View Log button or by selecting from the menu:

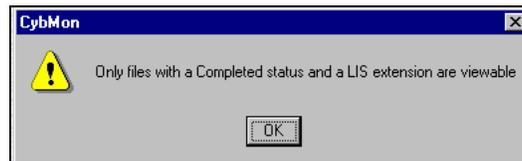
**F**ile ► **V**iew Log

*Note: If the process log is not complete, this window will not refresh automatically. Use the Refresh button to manually invoke the refresh function.*

## Viewing a report

Reports		
Name	Size	Status
AUDITRL1.LIS	163122	Complete
AUDITRL2.LIS	35129	Complete
CHEQUENUM.LIS	2284	Complete
COMBREG.LIS	118431	Complete
DEPSLIPS.LIS	0	Complete
PAYSLIPS.LIS	23000	Complete
PAYXTR.LIS	671	Complete
TRANSLOAD.LIS	806	Complete
TRANSLOAD2.LIS	941	Complete

## Error Message



---

## NOTES

---

## Using the Enhanced Reporting features, continued

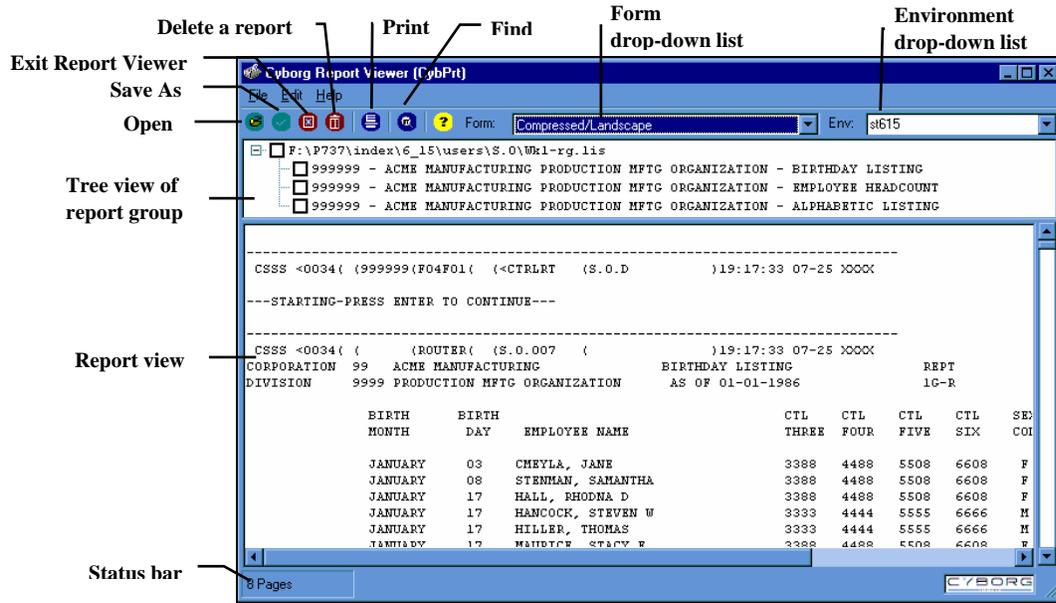
### **View a report**

You may select a report to view by double-clicking on the report in the Reports window in the Process Monitor dialog. The Report Viewer utility is launched in a separate dialog and the selected report is displayed. Only reports that have an extension of LIS are viewable.

### **Error Message**

LIW files are reports that are in progress; an error message is displayed if a file with an extension of LIW is selected.

# View completed reports



## NOTES

# Using the Enhanced Reporting features, continued

## Online viewing of completed reports

After a report has been initiated online, you may want to review it prior to making the decision to save or print it. Viewing the report online provides you with an opportunity to review any errors and verify the report content and totals.

The new Report Viewer is a Solution Series utility that provides a quick and intuitive method to manage, view, print, save, and delete selected reports.

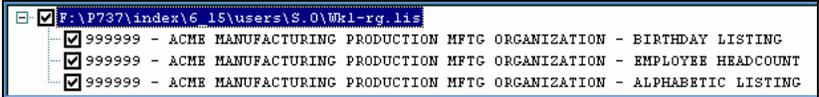
You may access the Report Viewer directly from the Launch toolbar. If the Launch toolbar is not displayed, you may view it by selecting from the menus.

The Report Viewer displays the completed report in a customizable window. The reports are stored in your own user folder, and you can customize your viewing and printing preferences. You can print a report, and you can search and jump within reports. You can also save and delete reports.

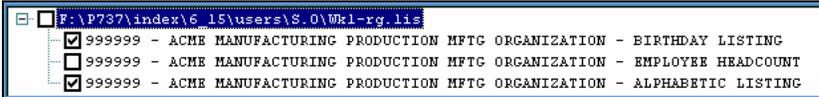
From the Report Viewer, you may select a report to view by clicking the Open Report button on the toolbar. The Open dialog will display all the output (LIS) files available in the specified User directory. You may select a report (LIS file) from the list to view it.

The top portion of the Report Viewer provides a tree view of the selected report group and all of its components. If you highlight an individual report in the tree view, the beginning of that report will automatically display. You can print, rename, and delete reports by selecting the appropriate check box.

If you select the root entry, or the entire report, all of its components will be automatically selected, as shown in the following example:



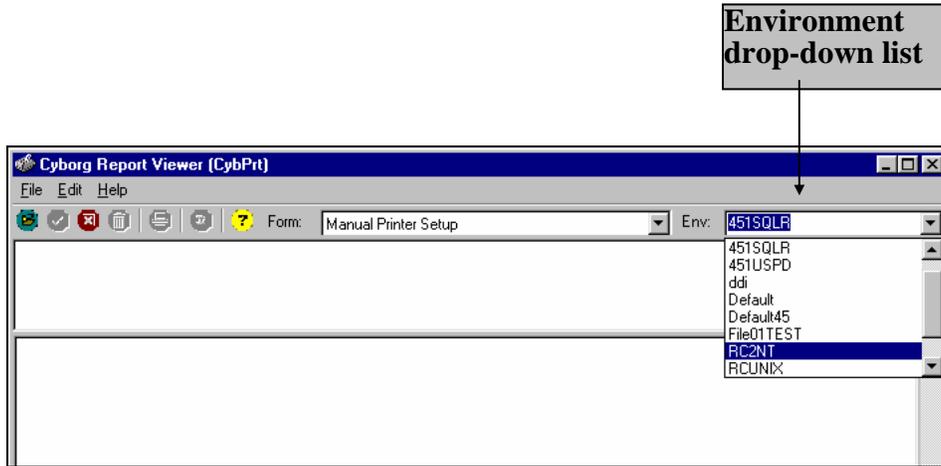
If you want to select one or more of the reports in the report group, clear the root entry check box and each appropriate report check box. In the following example, all reports except the Employee Headcount will be included:



The Form drop-down list is used to select a print format to produce the report. Your organization can customize these form entries, which can be used to handle any type of printer. Use the Manual Printer Setup entry to pass no control codes to the printer. This option is used when the printer is manually configured:



# Select Environment



---

## NOTES

---

## Using the Enhanced Reporting features, continued

### Select the environment

The Env drop-down list is used to select the environment used by the Report Viewer utility. This field is populated automatically from the Environments folder.

*Note: The Specify Environment Folder and Specify User dialogs will automatically display the first time the Report Viewer utility is run. Consult your technical staff if you do not know the correct Environment or User folder.*

The middle of the form displays the report detail. The scroll bars below and to the right of the report can be used to position the page for viewing or to scroll to other pages.

The status bar at the bottom of the Report Viewer window displays the total number of pages in the report.

### Report disposition

After the report is completed and you have viewed it to determine its accuracy, you may rerun the report, search it, print it, or delete it. Keep in mind that storage space could be an issue on your organization's server, so reports that are no longer needed should be deleted or moved to another storage location.

If you run the same report twice, the new report will replace the old one. Be sure to save files to different names if you want to archive them. To save a report to a different name, you must select the report and click the Save As button on the toolbar.

To search a report, you must click the Find button on the toolbar. The Find dialog will be displayed. This tool is especially useful when searching for error or warning messages.

The Find tool searches across entire files, not just selected reports. It finds the first occurrence of the specified string. Any subsequent search starts wherever the last search finished, unless you select the Start from Top check box. You can also perform case-sensitive searches.

To print a report, you must select the report and then click the Print button on the toolbar. When you select a report to be printed, the Print dialog will be displayed.

To delete a report group, you must click the Delete a Report button on the toolbar. Accept the warning message to delete the report.

You must be assigned the appropriate security to delete report files. Contact your technical staff for more information.

## Summary

- **Report features**
- **Report process**
- **Submitting, routing, and viewing reports**
- **Using the Enhanced Reporting features**

---

### NOTES

---

## Section summary

In this section you learned about how reports are processed, scheduled and run. We also identified alternate print files.

### Batch report process

- Reports can extract data from either the Employee Database (FILE02) or the \_\_\_\_\_.
- Once the report(s) are scheduled, the \_\_\_\_\_ program extracts data. This data is then sorted and finally printed using the \_\_\_\_\_ program.
- The final report output is written to either FILE03 or one of the alternate print files \_\_\_\_\_, \_\_\_\_\_ or \_\_\_\_\_. The report output file is dependent on the forms code literal.

### Submitting, routing and viewing reports

- As an alternate to submitting reports off-line, the \_\_\_\_\_ program allows you to initiate a batch job from within The Solution Series.
- If the option 'Hold output for online review?' was selected from the SUBMIT form, you may preview the report using the \_\_\_\_\_ program.
- In addition to previewing your report, the VIEW program is used to \_\_\_\_\_ and \_\_\_\_\_ reports for online review.

### Using the Enhanced Reporting features

- Reports may be launched directly from the \_\_\_\_\_, \_\_\_\_\_, or \_\_\_\_\_.
- The \_\_\_\_\_ allows you to view the progress of a report while it is running.
- The \_\_\_\_\_ displays the completed report in a customizable window.

## Section 2 Exercise

---

**NOTES**

---

## Section 2 exercise

### Purpose

The purpose of this exercise is to give you practice using SUBMIT to initiate the report run and VIEW to see the output.

### Directions

Take 10 minutes to complete the following steps:

1. Schedule the following report for batch submission. Review HELP documentation for the purpose of each report.

Reports: 1G-RPT, 1A-RPT

Company: 999999

2. Use the SUBMIT program to initiate the report process, using the report group name you established in Step 1. Be sure to route your report output for online review.

*Note:* Depending on your operating system and platform, it may be necessary to log-off your online session and initiate the batch report job.

3. Preview the report output using the VIEW program.

### Purpose

The purpose of this exercise is to give you practice using the Enhanced Reporting features to initiate and review a report.

### Directions

Take 10 minutes to complete the following steps:

1. Schedule the following report for batch submission. Review HELP documentation for the purpose of each report.

Reports: 1G-RPT, 1A-RPT

Company: 999999

2. Use the Navigator or checklist to initiate the report group name you established in Step 1.
3. Use the Process Monitor utility to check the progress of the report.
4. View the report output by double clicking on the report in the Reports window in the Process Monitor dialog.

---

**NOTES**

---

## Section 3: The Report Extract

---

### Table of Contents

Introduction.....	71
The Report Extract program .....	73
Defining the report.....	75
Report initialization and selection .....	81
The extract record .....	83
Section summary.....	99
Section 3 exercise .....	101

## Objectives

- **Identify the report  
extract program structure**
- **Identify the extract record components**
- **Use Cyborg Scripting Language verbs  
to build an extract record**
- **Recall basic scripting skills**

---

### NOTES

---

## Introduction

### **Purpose**

In this section, you will learn to create a report extract program.

### **Objectives**

Upon completion of this section you will be able to:

- Identify the report extract program structure
- Identify the extract record components
- Use CSL verbs to build an extract record
- Recall basic scripting skills

## **Report Extract Program Structure**

- **Report definitions**
- **Report headings**
- **Initialization**
- **Employee selection**
- **Sort key definition**
- **Report data**
- **Write extract record**

---

**NOTES**

---

## The Report Extract program

### **Extract program flow**

The Report Extract program's logic requires a specific execution sequence. Sequence includes:

### **Report definition**

The program is defined as a report program. Instructions for overriding standard headings are options here. Also, if required, input parameters are defined.

### **Report headings**

Heading literals are prepared for the user created portions of the four standard headings.

### **Initialization**

Optionally, Work Areas and Employee Database segment pointers are initialized.

### **Employee selection**

Conditional logic tests include/exclude records from the report's extract file dependent upon the purpose of the report.

### **Sort key definition**

Data that defines the report's sequencing and optional control break is extracted.

### **Report data**

Data is extracted for the report's detail.

### **Write extract record**

The sort and report data is written to the report extract file.

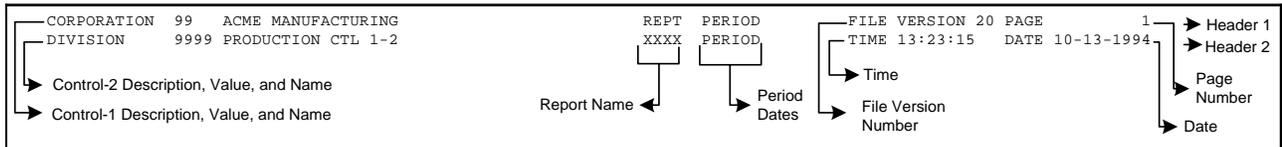
# Defining the Report

## NAMING THE REPORT

- 6-Character program Name
- User-Defined Programs Begin With 'X'
- Program Name Ends in 'PT'
- EXAMPLES: XA-RPT, X9-RPT, XAAAPT

## DEFINE-REPORT

Format: *DEFINE-REPORT*{report options...}.  
Example: Report Definition with no overrides/input parameters.  
Result: DEFINE-REPORT.  
All default report headings appear.



## NOTES

## Defining the report

### Naming the report

There are rules associated with naming a batch report program. You must use all six characters. By convention, user-defined reports start with an 'X'. A report program does not execute under the REPORT utility unless its fifth and sixth characters are 'PT'.

☞ *Refer to the example program names on the facing page.*

### Defining the report

The **DEFINE-REPORT** verb must be the first verb in your program. This verb can be issued alone or with report options. The options are used to:

- Suppress standard headings
- Allocate memory for input parameters

☞ *Refer to Section 7: Reports-Special Functions for more information about allocating memory for input parameters.*

### Standard headings

The system supplied headings are present unless suppressed with an option in the **DEFINE-REPORT** verb. The sample report on the facing page shows the default headings. The standard headings include:

- Company name and values
- Period End Literal
- File Version Number
- Page Number
- Report Program Name (1st 4 characters)
- Date and Time

# DEFINE-REPORT Options

<b>DEFINE-REPORT Default Headings</b>	<b>DEFINE-REPORT Option</b>
Control 1-2	NO-CONTROL-1-2
Second Header Line	NO-HEADER-2
Third Header Line	NO-HEADER-3
Fourth Header Line	NO-HEADER-4
Period End Dates	NO-PE-DATES
File Version Number	NO-VERSION-NUMBER
Page Number	NO-PAGE-NUMBER
Time	NO-TIME
Date and Time	NO-DATE-OR-TIME
No Headers	NO-HEADINGS
Triple Space After Headings	TRIPLE-AFTER-HDNGS

Example 1: DEFINE-REPORT NO-HEADINGS.

Example 2: DEFINE-REPORT NO-PE-DATES NO-TIME.

---

## NOTES

---

## Defining the report, continued

### DEFINE–REPORT options

Any of the options can be included in the DEFINE–REPORT verb. These options are listed as a string of instructions with a period (.) terminating the DEFINE–REPORT syntax.

- When suppressing an entire heading line(s), no additional physical space is available for printing detail on the report.
- The Period End Dates are left blank. Data can be supplied as overrides.

☞ *Refer to Section 7: Reports–Special Functions for more information.*

- Additional headings can be created in the Special Print Options program if you do not specify NO–HEADINGS.

☞ *Refer to Section 8: Introduction to Special Print Options for additional information.*

### Title and column headings

As you may have noticed, the report’s title, subtitle, and detail column headings are not created using DEFINE–REPORT. These headings are discussed in the next topic.

# Title and Column Headings

## HEADER-n

Format: *HEADER - n : nnn ' literal string '*

Example: DEFINE-REPORT.  
 HEADER-1 :53 'SAMPLE BATCH REPORT'.  
 HEADER-2 :57 'CLASS EXAMPLE'.  
 HEADER-3 :01 'EMPLOYEE           EMPLOYEE NAME'.  
 HEADER-3 :57 'BIRTH            HIRE'.  
 HEADER-4 :01 'NUMBER'.  
 HEADER-4 :57 'DATE            DATE'.

Result: Report Title, Subtitle and Detail Column Headings

	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	1	1	1	1	1	1	1
	1	..	5	..	0	..	5	..	0	..	5	..	0	..	5	..	0	..	5	..	0	..	5	..	0
CORPORATION	99	ACME	MANUFACTURING																FILE	VERSION	00	PAGE		1	
DIVISION	9999	PRODUCTION	CTL 1-2																TIME	13:23:15	DATE	10-13	1994		
EMPLOYEE		EMPLOYEE	NAME																						
NUMBER																									

→	SAMPLE BATCH REPORT	REPT PERIOD	FILE VERSION 00	PAGE	1
→	CLASS EXAMPLE	EXMR PERIOD	TIME 13:23:15	DATE 10-13	1994
→	BIRTH	HIRE →	HEADER-3		
→	DATE	DATE →	HEADER-4		
→	HEADER-2				
→	HEADER-1				

## NOTES

## Defining the report, continued

### Title and column headings

Title and column headings are constructed using literal values in the header instruction.

- Header lines can be coded in any order
- Header instructions (for a given line) can be repeated
- Headers are automatically initialized to spaces

### HEADER–n

To create title and column headings unique to your report you must use the **HEADER–n** statements where ‘n’ is the header number 1–4:

Title:	HEADER–1 :nnn	(049–083, total length 35)
Sub–Title:	HEADER–2 :nnn	(049–083, total length 35)
Column Heading 1:	HEADER–3 :nnn	(001–132)
Column Heading 2:	HEADER–4 :nnn	(001–132)

Where **:nnn** is a beginning print position for the heading literal, which can be expressed as a length of 1, 2, or 3 digits. The numbers in parentheses represent valid print positions.

### Heading literals

The heading literal is enclosed in single quotes ('). A header instruction in a given EDIT sequence line cannot have more than one pair of single quotes. Positions not included in a HEADER–n definition will be blank filled.

# Initialization and Selection

## SET-EMP-PTRS-TO-1ST

**Format:** SET-EMP-PTRS-TO-1ST.

**Example:** P100-INITIALIZE.  
SET-EMP-PTRS-TO-1ST.  
P200-SELECT.  
IF CONTROL-4-CODE NOT EQUAL 'MANU'  
RETURN.

**Result:** The Employee Record E, F, G, H, J, L, and P segment pointers are set to the first occurrence.

## SET-CO-PTRS-TO-1ST

**Format:** SET-CO-PTRS-TO-1ST.

**Example:** P100-INITIALIZE.  
SET-CO-PTRS-TO-1ST.  
P200-SELECT.  
IF SEX-CODE NOT EQUAL 'M' OR 'F'  
RETURN.

**Result:** The Company Record A, B, C, and D segment pointers are set to the first occurrence.

## SPACE-EXTRACT-RECORD

**Format:** SPACE-EXTRACT-RECORD.

**Example:** P100-INITIALIZE.

SPACE-EXTRACT-RECORD.  
SET-EMP-PTRS-TO-1ST.  
P200-SELECT.  
IF CONTROL-3-CODE NOT EQUAL '1000'  
RETURN.

**Result:** The Extract Record area (SCREEN) is initialized to spaces, and the Employee Record segment pointers are set to the first occurrence.

---

## NOTES

---

## Report initialization and selection

### Initialization

Initialization logic is optional in a report extract program. However, you may wish to always include these three verbs in a report initialization paragraph:

- **SET-CO-PTRS-TO-1ST**  
Sets the Company Record A, B, C, and D segment pointers to the first occurrence.
- **SET-EMP-PTRS-TO-1ST**  
Sets the Employee Record E, F, G, H, J, L, and P segment pointers to the first occurrence.
- **SPACE-EXTRACT-RECORD**  
Moves spaces to pointer 11 (SCREEN) for 150 positions starting at position :1601. The SCREEN area is the work area for extract record building.

Additionally, Work areas and Perm/Temp counters should be initialized at this point.



*Refer to Section 5: Sort Key Options for additional information.*

### Selection

After initialization, you may optionally have single or multiple levels of selection logic. In the first example above, the employee's G Segment CONTROL-4-CODE is being tested. Only employees with a value of 'MANU' are reported.

### RETURN verb

A RETURN is implied at the end of the extract program. When encountered, the REPORT process executes the rest of the reports in the report group and then reads the next employee record and re-executes the program.



## The extract record

### Extract record

The report extract program builds a fixed length, 150-character record that contains 2 parts:

- Sort portion
- Report portion

### Sort portion

The sort portion is variable in length and contains from 11 to 60 characters. The minimum length of 11 characters includes:

- A forms code identifying the print file
- A report code identifying the report
- A Control 1-2 value

The sort portion may also contain additional fields to further sequence the report.

### Report portion

The report portion is variable in length and contains 90 to 139 characters:

- A one-character record type
- An optional one-character carriage control code
- The data to appear on the report

# The Sort Key

PRINT forms-code report-code FORMS/REPORT-CODE { PRINT-GRAND-TOTAL  
NO-PRINT-GRAND-TOTAL }

## FORMS/REPORT-CODE and [NO-]PRINT-GRAND-TOTAL

**Example:** Building the sort key for the extract record indicating no total or control break at end of report.

```
P300-SORT.  
PRINT '1EXMR' FORMS/REPORT-CODE  
NO-PRINT-GRAND-TOTAL.
```

**Result:** The EXMRPT report has no total or control break at the end of the report.

---

## NOTES

---

## The extract record, continued

### Sort key

The data that comprises the sort portion of the extract record can be referred to as the sort key. The first 11 characters of the sort key must always be the forms code, report code, and organization number (Control 1-2) value.

### Forms code

The forms code indicates the output print file to which the report is written. The valid forms code values are:

<u>Forms Code</u>	<u>Output File</u>
1	FILE03
2	FILE17
3	FILE18
4	FILE19

### Report code

The first four characters of the report program name are used to create the report code. The combination of the forms code and report code literals, as in the example, creates the five character literal '1EXMR'. These values must always be the first part of the extract record and must be followed by the FORMS/REPORT-CODE verb.

### Organization number (Control 1-2)

The organization number value follows the forms and report code literal. This can be expressed as one field (CONTROL-1-2) or two separate fields (CONTROL-1 and CONTROL-2).

### Sort key options

The NO-PRINT-GRAND-TOTAL and PRINT-GRAND-TOTAL keywords indicate whether totals are given at grand total time.

# Calculating the Sort Length

## SORT-LENGTH-*nn*

Format: *PRINT* { *literal* } *SORT-LENGTH-*nn**.

**Example:** Indicating that the sort key has 21 characters.

```
P300-SORT.  
PRINT '1EXMR' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL  
SORT-LENGTH-21  
CONTROL-1-2  
EMPLOYEE-NUMBER.
```

**Result:** The extract record is built with a 21-character sort key.

Sort Length for the EXMRPT is calculated by adding:

```
FORMS CODE 1  
REPORT CODE 4  
CONTROL-1-2 6  
EMPLOYEE-NUMBER 10  
21
```

---

## NOTES

---

## The extract record, continued

### Sort length

The length of the sort key must be manually calculated. Simply count the length of each item in your sort key. Using the EXMRPT as an example, the sort length is 21.

- The SORT-LENGTH-21 follows the PRINT-GRAND-TOTAL keyword in this example.

☞ *Refer to Section 5: Sort Key Options for additional information about sort length placement.*

- The sort key length is passed to the print phase of the report process and indicates where the sort portion ends and the report portion begins.

In this example, the sort key is composed of alphanumeric data. Therefore, the use of PRINT or OUTPUT to extract the sort key data does not impact the length. However, the rules associated with the PRINT and OUTPUT must be considered when the following data types are used in the sort key:

- Date data
- Numeric data
- Computational data

# Calculating the Sort Length

## PRINT

PRINT { literal  
field-name-1 } { literal...  
field-name-2. }

## OUTPUT

OUTPUT { literal  
field-name-1 } { literal...  
field-name-2.. }

**Example:** Sort Length using PRINT

```
P300-SORT.  
PRINT '1EXMR' FORMS/REPORT-CODE  
NO-PRINT-GRAND-TOTAL  
SORT-LENGTH-31  
CONTROL-1-2 EMPLOYEE-NUMBER  
EMPLOYMENT-DATE.
```

Sort Length using OUTPUT

```
P300-SORT.  
OUTPUT '1EXMR' FORMS/REPORT-CODE  
NO-PRINT-GRAND-TOTAL  
SORT-LENGTH-27  
CONTROL-1-2 EMPLOYEE-NUMBER  
EMPLOYMENT-DATE.
```

## SPACE-OVER :nn

Format: SPACE-OVER :nn

Example: Spacing over when a segment is not found.

```
P300-SORT.  
PRINT '1EXMR' FORMS/REPORT-CODE  
NO-PRINT-GRAND-TOTAL  
SORT-LENGTH-27 CONTROL-1-2.  
FIND-JOB-EFFECTIVE.  
IF FOUND  
PRINT JOB-CODE  
ELSE  
SPACE-OVER :06.  
PRINT EMPLOYEE-NUMBER
```

---

## NOTES

---

## The extract record, continued

### PRINT

The PRINT verb moves field data and literal values to Pointer 11 (SCREEN).

- PRINT edits fields according to the edit length and edit routine specified on the Field Name Table.
- PRINT **must** be used to extract Option List Description field names for a report.

### OUTPUT

The OUTPUT verb moves field data or literal values to Pointer 11 (SCREEN).

- OUTPUT moves data without editing or converting it in any way, and displays it as it is stored in the record.
- OUTPUT **must** be used when a field in the extract record is to be totaled. This is applicable to both numeric and computational fields.

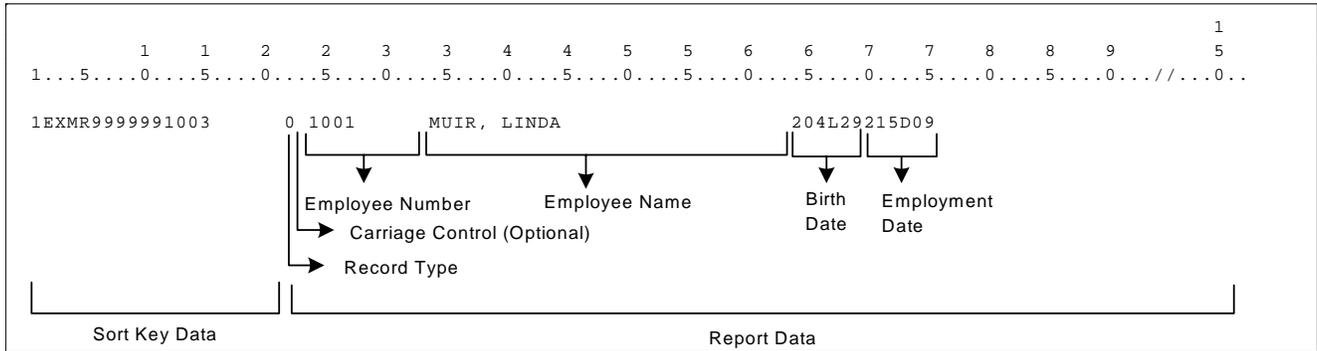
### SPACE-OVER :nn

The SPACE-OVER :nn verb is used to move spaces to Pointer 11 (SCREEN) to fill in a specified number of positions in the sort or report portion of the extract record when the employee does not have the segment data.

- The number of spaces must be indicated by a colon and a two-digit number between 01 to 60.

*Note: The sort portion of each record written must be uniform in length. It may contain segment data, numeric or alphanumeric literals, or spaces.*

# The Report Portion of the Extract Record



## NOTES

## The extract record, continued

### Report portion

The report portion of the extract record follows the sort key data. It is variable in length. Like the sort portion, the report portion of the extract records for a given report program must be uniform in content. This portion is also built with PRINT or OUTPUT statements and contains:

- A record type
- A carriage control indicator (optional)
- The report data

### Record type

A record type literal value must always be the first character following the sort key. Valid values are **0–9** and **A–Z**. A choice within this range is arbitrary, however the following rules apply:

- A report program only processes up to 32 extract record types.
- When the sort key for a group of extract records does not vary, the record type literal must vary.
- The print program processes records with matching sort keys in record type order. This may impact your decision on record types

☞ *Refer to Section 8: Introduction to Special Print Options for additional information.*

- The record type value is used in the RTEDIT layout.

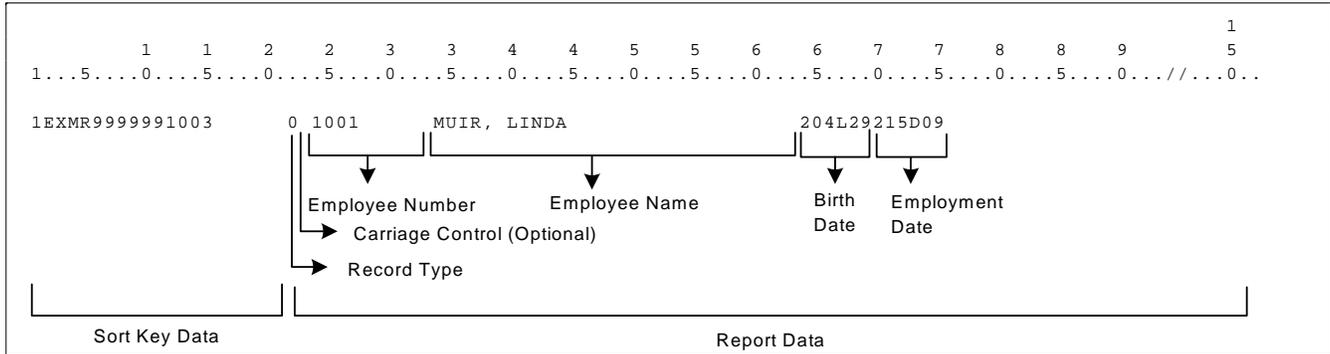
☞ *Refer to Section 7: Reports—Special Functions for additional information.*

# Carriage Control Options

**Example 1:** Extract With Carriage Control

P400-OUTPUT.

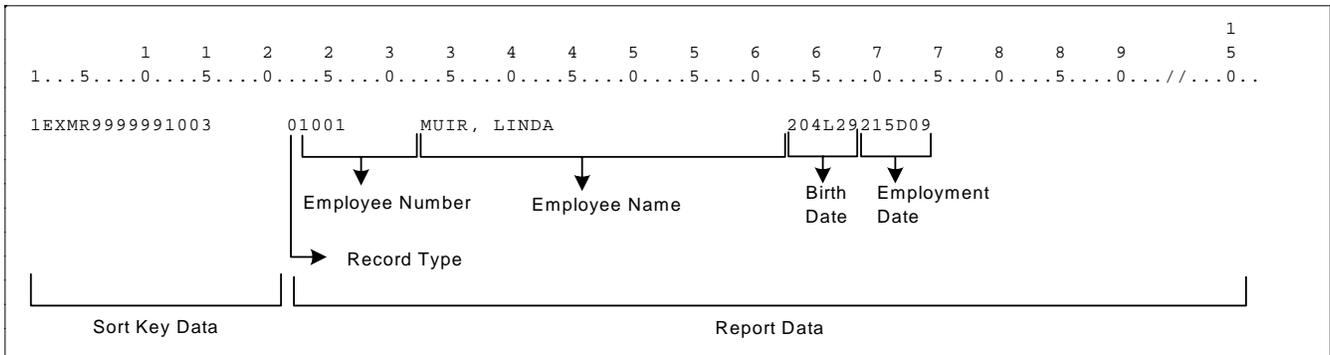
```
PRINT '0' EMPLOYEE-NUMBER EMPLOYEE-NAME.
OUTPUT BIRTH-DATE EMPLOYMENT-DATE.
```



**Example 2:** Extract Without Carriage Control

P400-OUTPUT.

```
PRINT '0' EMPLOYEE-NUMBER EMPLOYEE-NAME.
OUTPUT BIRTH-DATE EMPLOYMENT-DATE.
```



## NOTES

## The extract record, continued

### Carriage control

A carriage control literal value can be moved to the extract record. If present, it must immediately follow the record type. This optional value controls the spacing between detail lines on the printed report:

- (blank) Single-spacing (this is the default)
- 1 Top of Page
- 0 Double-spacing
- - (hyphen) Triple-spacing

### Example code and extract

The first example shows a carriage control value of blank. This indicates single spacing. In the second example no carriage control literal is designated, therefore, the default single spacing applies.

*Note:* When a carriage control literal is included in the extract record, it must be accounted for in the RTEDIT layout. This is discussed in the next section.

# Completing the Extract Record

## WRITE-EXTRACT

**Format:** *WRITE-EXTRACT*

**Example:** Move the data extracted for the report to the extract file, FILE15.  
P400-OUTPUT.

PRINT '0' EMPLOYEE-NUMBER EMPLOYEE-NAME.

OUTPUT BIRTH-DATE EMPLOYMENT-DATE.

**WRITE-EXTRACT.**

**Result:** The data in the SCREEN work area is written to the extract file (FILE15).

---

## NOTES

---

## The extract record, continued

### Report data

The remainder of the extract record is filled with field and/or literal data. The amount of data written to the extract record is dependent upon the use of PRINT vs. OUTPUT. General rules:

- Numeric or Computational fields that are to be totaled **must** be moved to the extract area with OUTPUT.
- Option List Description fields can only be moved to the extract record using PRINT.
- It is more efficient to use OUTPUT to move data to the extract record since it requires no editing prior to moving the data, and the fields are smaller when output in their storage format.

### Writing the record

After the data for the report has been moved to the extract area, the WRITE-EXTRACT must be issued. This instruction moves data from the SCREEN area (Pointer 11) to the output file, FILE15.

*Note:* Every extract record must also have a **Sort Key**.

# The EXMRPT Example

Security:	SECURITY `**`, @ CLASS REPORT EXAMPLE @LAST MODIFIED ON: 09-14-94 BY: USER AUTHOR: USER	XPP
Report Definition:	DEFINE-REPORT.	
Headings:	HEADER-1 :53 'SAMPLE BATCH REPORT'. HEADER-2 :57 'CLASS EXAMPLE'. HEADER-3 :01 'EMPLOYEE EMPLOYEE NAME'. HEADER-3 :57 'BIRTH HIRE'. HEADER-4 :01 'NUMBER'. HEADER-4 :57 'DATE DATE'.	
Initialization:	P100-INITIALIZE. SET-EMP-PTRS-TO-1ST. SPACE-EXTRACT-RECORD.	
Selection:	P200-SELECT. IF CONTROL-4-CODE NOT EQUAL 'MANU' RETURN.	
Sort Key:	P300-SORT. PRINT '1EXMR' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL SORT-LENGTH-21 CONTROL-1-2 EMPLOYEE-NUMBER.	
Report Data:	P400-OUTPUT. PRINT '0' EMPLOYEE-NUMBER EMPLOYEE-NAME. OUTPUT BIRTH-DATE EMPLOYMENT-DATE. WRITE-EXTRACT. RETURN.	

---

## NOTES

---

## The extract record, continued

### **EXMRPT sample**

The EXMRPT example can be used as a prototype. Its structure follows the flow of a typical report extract program that includes:

- Report definitions
- Report headings
- Initialization
- Employee selection
- Sort key definition
- Report data
- Write extract record

## Section Summary

- **The Report Extract program**
- **Defining the report**
- **Report initialization and selection**
- **The extract record**

---

NOTES

---

## Section summary

### The report extract program

In addition to the Security and Last Modified coding, the report extract program requires the following structure:

- **Report Definitions**—The \_\_\_\_\_ command controls the printing of standard headings. If no options are chosen, the system defaults are used for heading information.
- **Report Headings**—The \_\_\_\_\_ command controls the printing of User defined titles and column headings.
- **Initialization**—Common initialization includes setting the segment pointers to the first occurrence using the \_\_\_\_\_ for the employee record segments and \_\_\_\_\_ for the company record segments.
- **Selection**—When only certain employees are to appear on the report \_\_\_\_\_ is included.

### The report extract record

The extract record consists of two parts: \_\_\_\_\_ and \_\_\_\_\_.

- **Sort Key**—The sort key minimum requirements include a one-character \_\_\_\_\_ followed by a four-character \_\_\_\_\_ and six-character Organization Number (Control 1-2) value. The minimum SORT-LENGTH-nn is \_\_\_\_\_.
- **Report Data**—The report portion of the extract record requires a \_\_\_\_\_ in the first position. When the extract record is complete, it is written to \_\_\_\_\_ using the \_\_\_\_\_ verb.



## Section 3 exercise

### Purpose

The purpose of this exercise is to give you practice developing a report extract program.

### Directions

Using the EDIT program, develop a report extract program.

- Use the system default headers and follow the sample layout for the creation of the title and column headings.
- Be sure to position the employee segment pointers to the first occurrence.
- Include only Non-union employees on the report (UNION-CODE field contains no data).
- Display the report single-spaced in Employee Number order.
- Include the following fields on the report:

EMPLOYEE-NAME

EMPLOYEE-NUMBER

PAY-FREQUENCY-CODE

PAY-FREQUENCY

PAYMENT-CODE and PAYMENT-TYPE

SALARY (H Segment, HED Number 001)

EMPLOYMENT-DATE

BIRTH-DATE

---

**NOTES**

---

## Section 4: Report Detail and Totals - The RTEDIT Form

---

### Table of Contents

Introduction.....	105
Report detail, totals, and printing.....	107
Print positions and the extract record .....	111
Print position and the report output .....	113
Report extract layout.....	117
Compiling report print positions.....	121
Section summary.....	123
Section 4 exercise .....	125

## Objectives

- **Define the purpose of the report print step**
- **Describe the function of the report print position edit form**
- **Identify the relationship between the extract record, report print position records, and the report output**
- **Create and compile print position records**

---

### NOTES

---

## Introduction

### **Purpose**

In the previous section, you developed a program to write the report extract records. The second step in developing a report is to create print position records using the Report Print Position Edit form (RTEDIT).

### **Objectives**

Upon completing this section, you will be able to:

- Define the purpose of the report print step
- Describe the function of the report print position edit form
- Identify the relationship between the extract record, report print position records, and the report output
- Create and compile print position records

# Report Detail and Totals

SEQ NBR	FIELD NAME	----PRINT----		
		POS.	LINE	TOTAL

---

## NOTES

---

## Report detail, totals, and printing

### Overview

The format and placement of data items from the extract records on the report is accomplished using print position records. You create print position records with Report Print Positions Edit (RTEDIT), and compile the group of records with Report Print Positions Compile (RETYPE). These 'RT' and 'R' records are stored on FILE01.

### Print position records

The print position records serve four purposes:

- Provide a layout of the extract record for each record type within a report
- Define the print positions for each data element extracted for the report
- Determine data element report format
- Determine report totaling

# Report Print Positions Edit (RTEDIT)

REPORT PRINT POSITION PARAMETERS RTEDIT

The RTEDIT program defines output detail and total line parameters for CBSVB reports.

Complete the text boxes below, then hit enter:

For inquiry only, enter 'I':

Enter the source FILE NAME:  four positions  
and RECORD TYPE:

---

## NOTES

---

## Report detail, totals, and printing, continued

### Report print position edit

The Report Print Position Edit (RTEDIT) form is used to establish the extract record layout and the print instructions for your report. It is a required element in packaged report writing.

Make the following selections from the Navigator

- Component:**  Development Tools
- Process:** Programming Utilities
- Task:**  Edit Report Print Positions

**Result:** The Report Print Position Parameter form displays.

### RTEDIT naming requirement

The RTEDIT name is defined as the 1st four positions of the Report Extract program. Each record type used on the extract records will have a set of Report Print Position records. The record type immediately follows the name on each record.

- Our sample program EXMRPT uses 'EXMR0' as the name. The first four characters are the first four characters of the extract program name. The fifth character is the record type used.

*Note:* A print position layout is needed for each different Record Type used by the extract program.



## Print positions and the extract record

### Relation to the extract

The fields on the Print Position Edit form provide the layout of the data in the extract record. Only fields that follow the Record Type literal are represented here. The field definitions to define the extract layout are:

### Command

The Command field is used to (A) Add, (C) Change, or (D) Delete an entry.

### Seq Nbr

The two-digit Seq Nbr is a range of 00–98. The Seq Nbr designates the order of the fields on the extract record (that is, the fields must be listed in extract record order).

### Field name

The Field Name represents the data in your extract record.

All items that follow the record type are represented by the field name(s). This includes the Carriage Control character **if designated** in your extract record.

The fields named here must be valid entries in the Field Name Table. The field name characteristics for the data dictionary name entered in RTEDIT are used when the data is printed on the report.

The Field Name CARRIAGE-CONTROL can be used to describe the line spacing literal value.

If an invalid entry is made, an at-sign (@) is placed in the erroneous field. All errors must be resolved before the print position records may be compiled.



## Print position and the report output

### Relation to the output

The '---PRINT---' fields on the Print Position form provide the detailed print positioning and totaling options for the report output.

### Print Pos

The Print Pos specifies the beginning print position for each field in the detail or total line. This **MUST** be expressed as three digits.

- Valid print positions are 001–132.
- The CARRIAGE-CONTROL must be print position 000.
- Items that are detail and total can be specified in the same PRINT POS and in the same PRINT LINE. They are treated as separate line items by RTPRNT.

### Print line

The Print Line field specifies the line on which the extracted data is to be printed on the report (default is 01).

The Print Line is relative to the current employee being printed, not the physical line on the report page. For example, the field name associated with Print Line 07 will print on the 7th output line for the **employee**, not the 7th **line** of the report.

### Print total

The Print Total field determines if a field will appear on the report as a detail, total, or detail and total field. For fields that have edit characteristics, it is important to coordinate the format of extract to the print indicator chosen.



## Print position and the report output, continued

### EXMR0 example

This provides a visual example of the interrelationship between the report extract, report print positions and the final report output.

- Sequence 01  
CARRIAGE–CONTROL is assigned print position 000 on line number 01 (default). If the carriage control literal is present in the extract, the print position must always be '000'.
- Sequence 05  
EMPLOYEE–NUMBER is assigned print position 001 on line number 01. This is an alphanumeric field extracted with PRINT. It is moved to the detail line with 'N' for print.
- Sequence 10  
EMPLOYEE–NAME is assigned print position 013 on line number 01. This is an alphanumeric field extracted with PRINT. It is moved to the detail line with 'N' for print.
- Sequence 15  
BIRTH–DATE is assigned print position 057 on line number 01. It is unedited in the extract record (OUTPUT), so 'N' is used to edit the data for the report line.
- Sequence 20  
EMPLOYMENT–DATE is assigned print position 070 on line number 01. It is unedited in the extract record (OUTPUT), so 'N' is used to edit the data for the report line.

# Extract and Print Position Record Coordination

Extract Verb	Print Total	Report Output Result
<b>PRINT</b>	N	Detail data is edited before being moved to the report - valid only if the field alphanumeric.
	O	Detail data is moved to the report in extract record format.
	P	Detail data is moved to the report at total time - valid only if the field is alphanumeric.
	T	Invalid option.
	Y	Invalid option.
<b>OUTPUT</b>	N	Detail data is edited for the report.
	O	Detail data is moved to the report unedited.
	P	Detail data is moved to the report edited at total time.
	T	Total data is moved to the report edited at total time.
	Y	Detail and totals of the detail are moved to the detail and total lines edited.

SOCIAL-SECURITY-NBR		
Verb	Extract	RTEDIT Total
PRINT	308-82-3775	N, O, or P
OUTPUT	308-82-3775	N, O, or P

NORMAL-HOURS		
Verb	Extract	RTEDIT Total
PRINT	\\40.00	O
OUTPUT	0004000	N, Y, or T

CONTROL-3		
Verb	Extract	RTEDIT Total
PRINT	Midwest	O or P
OUTPUT	(Invalid)	(Invalid)

SALARY-EFFECTIVE		
Verb	Extract	RTEDIT Total
PRINT	10-01-1992	O
OUTPUT	207C32	N or P

HED-AMOUNT-YTD		
Verb	Extract	RTEDIT Total
PRINT	\\30,158.92	O
OUTPUT	#@%&#	N, Y, or T

---

## NOTES

---

## Report extract layout

### Extract/print position

The table above summarizes the combination of extract verb and PRINT/TOTAL options used for the desired report result. As you can see, you must coordinate the appropriate extract verb with the RTEDIT option. Here are some general guidelines.

### Alphanumeric fields

Alphanumeric fields are the same regardless of using PRINT or OUTPUT, therefore the options of N, P, or O are valid.

*Note: Option List description fields should always be PRINTed since the OUTPUT verb extracts the entry field instead of the corresponding description field.*

### Date fields

Date fields are stored differently than they are displayed.

- If the date is OUTPUT, you may use the N or P option to display the result in MM-DD-YYYY format. The O option is used to display the result in storage format (CYYMDD or YYMMDD depending on the type of date).
- If the date is PRINTed, use only the O option to display the result in MM-DD-YYYY format if display length is 10 or MM-DD-YY if display length is 8.

### Numeric fields

Numeric fields are stored in many different formats: binary, packed, computation, or display numeric.

- Regardless of the storage format of the number, if the number is PRINTed, the only valid option is O.
- Conversely, if the number is OUTPUT, use the N, O, P, T, or Y option for the desired result. Remember, an item can only be totaled if it is in its storage format without edit characters.

# Syntax Errors

SEQ	FIELD	-----PRINT-----
NBR	NAME	POS. LINE TOTAL
A 05	@8-04-850HOLD	007 01 N *****

---

## NOTES

---

## Report extract layout, continued

### When field names do not match

You may find standard report programs whose extracted field names do not match name by name the items in the print position layout. Realize that print position instructions are only used during the print process.

- The FIELD NAMES in the print position list need only match the edit characteristics of the extracted field data. They are not used to reference the actual Employee Database or Control File data. They are directly referencing the extract record (FILE14).

### Literal values

Literal values may be present in the extract record. For example, you may have the following literal string **'NO JOB DATA ON FILE FOR EMPLOY'** in the extract. This must be described in your RTEDIT layout as a 30 character alphanumeric field. Use an established field (for example, JOB-TITLE) or create a field for this purpose.

### Work fields

For various reasons, your extract record may contain data that was first moved to a work field. You may have, for example, moved the employee's JOB-CODE to a field designated W8-06-800HOLDJOB. This item must be represented with a valid field name. In this case, the use of the field JOB-CODE is recommended. It exactly describes the field in the extract record both for print purposes and it is self-documenting.

# The Compile Report Print Positions (RETYPE)

```
COMPILE RTEDIT REPORT RECORDS                                RETYPE

The RETYPE program must be executed before the RTEDIT
screen data can be used. RETYPE verifies report data and
logic.

    Complete the text boxes below, then hit enter: X
    Enter the first four characters
    of the RTEDIT FILE NAME: EXMR
```

---

## NOTES

---

## Compiling report print positions

### Compiler

The Report Print Position Compile program (RETYPE) compiles print instructions for all Record Types associated with your report extract program.

To compile the print instructions:

Make the following selections from the Navigator:

- Component:**  Development Tools
- Process:** Programming Utilities
- Task:**  Compile all Report Print Positions

**Result:** The Print Position Compile form (RETYPE) is displayed.

Next, complete the following steps:

1. Type the report program's first four characters (for example, for the EXMRPT, 'EXMR').
2. Press Enter.

**Result:** ----Complete---- is displayed.

*Note:*

*Subsequent changes to the Program or Print Position records require a CSL Compile and a Report Print Positions Compile of the program in that order.*

### Errors

If the Report Print Position Compile is unsuccessful, refer to Appendix E: Print Position Compile (RETYPE) Error Resolution for a complete list of error messages and resolutions.

## Section Summary

- **Report detail, totals, and printing**
- **Report print positions records and the extract record**
- **Report print positions records and the report output**
- **Compiling print position records**

---

### NOTES

---

## Section summary

In this section you learned the role report print position edit plays in report development and maintenance.

### Report print position records and the extract record

- The report print position edit fields, Seq Nbr and \_\_\_\_\_, are used to provide the layout of the data in the extract record.
- The order (Seq Nbr) of the fields in the Report Print Position Edit form must correspond to the order of the data in the extract record. Only data following the \_\_\_\_\_ is represented in the Report Print Position Edit form.

### Report print position records and the report output

- Valid print positions for the Print Pos fields are \_\_\_\_\_ through \_\_\_\_\_ for the report detail and totals, and position 000 for the \_\_\_\_\_.
- When a field is to be totaled, the verb used to write the data to the extract record must be \_\_\_\_\_.

### Compiling print position records

- The \_\_\_\_\_ program compiles the print instructions for the report program.
- RETYPE is executed \_\_\_\_\_ regardless of the number of record types used by a program.



## Section 4 exercise

### Directions

Establish the Print Position record layout for the extract program developed in Section 3: The Report Extract. Run your report and view the output.

1. Use the Print Position Edit (RTEDIT) to layout the report data created in Section 3: The Report Extract. See the report example for print position guidelines.
2. Compile the print instructions. Resolve any errors.
3. Schedule your report using the Schedule Report Groups (RGMSTR) forms. Include only the organization number (Control 1-2) 999999 in your output.
4. Initiate the batch report job
5. Preview your report output.

---

**NOTES**

---

## Section 5: Sort Key Options

---

### Table of Contents

Introduction.....	129
Sort key options .....	131
Sort key length and placement.....	141
Section 5 exercise 1 .....	145
Special sort techniques.....	147
Section summary.....	155
Section 5 exercise 2 .....	157

## Objectives

- **Identify the sort key options**
- **Relate the sort key options to the report format**
- **Identify the placement of the sort length verb**
- **Recognize special techniques for constructing sort keys**
- **Develop a report with control breaks and totals**

---

NOTES

---

## Introduction

### **Purpose**

In this section, you will learn the rules and syntax associated with report control breaks, calculating the sort key length, and special techniques for writing sort logic.

### **Objectives**

Upon completion of this section you will be able to:

- Identify the sort key options
- Relate the sort key options to the report format
- Identify the placement of the sort length verb
- Recognize special techniques for constructing sort keys
- Develop a report with control breaks and totals



## Sort key options

### Sort basics

As you recall, each report has its own sort logic. The list of fields/literals extracted as the sort key provides the order in which the report prints. When a report requires no control breaks or totaling, the only sort key option required is on the highest sort level. You must use either:

PRINT-GRAND-TOTAL or

NO-PRINT-GRAND-TOTAL

### Special sort keywords

However, your report may require sub totals or page advancing to enhance the readability and usefulness of the report. Special sort key options are used to associate this type of logic with your sort key fields. These instructions produce control records for the print program when the report extract program is compiled. When the report prints, specific actions are taken when these designated sort key fields change.

### Changes to sort logic

Remember that changes to sort logic necessitate an execution of the CSL Compiler (RELOAD) and Report Print Position Compiler (RETYPE). The sort control information is associated with each report print position record layout and must be kept in sync.

## Sort Key Options

SORT KEY OPTION	REPORT WITH TOTALS	REPORT WITHOUT TOTALS
PRINT-GRAND-TOTAL	After single line advancement, the report end control break total is printed as: *REPORT CODE XXXX 9999.	After single line advancement the report end control break is printed as: *REPORT CODE XXXX.
NO-PRINT-GRAND-TOTAL	No report end control break/totals printed.	No report end control break is printed.
BREAK-DEFAULT	Sort key change prints a control break with totals after single line advancement.	Sort key change prints a control break after single line advancement.
NO-PRINT-SUBTOTAL	No control break or totals are printed when this sort key data element changes.	No control break is printed when this sort key data element changes.
NEW-PAGE-BEFORE	A new page with headings is printed before the control break/totals caused by the sort key change.	A new page with headings is printed before the control break caused by the sort key change.
NEW-PAGE-AFTER	A new page heading is printed after the control break/totals caused by the sort key change.	A new page with headings is printed after the control break caused by the sort key change.
SET-PAGE-NUMBER-TO-1	A new page with page number 1 is printed following the control break/total caused by the sort key change.	A new page with page number 1 is printed following the control break caused by the sort key change.
DOUBLE-SPACE-BEFORE	Double line advancement occurs before printing the control break/totals caused by the sort key change.	Double line advancement occurs before printing the control break caused by the sort key change.
DOUBLE-SPACE-AFTER	Double line advancement occurs after printing the control break/totals caused by the sort key change.	Double line advancement occurs after printing the control break caused by the sort key change.
TRIPLE-SPACE-AFTER	Triple line advancement occurs after printing the control break/totals caused by the sort key change.	Triple line advancement occurs after the control break caused by the sort key change.
TRIPLE-SPACE-BEFORE	Triple line advancement occurs before printing the control break/totals caused by the sort key change.	Triple line advancement occurs before the control break caused by the sort key change.

## Employee Counts

```

P200-INITIAL.
  MOVE :1 TO PERM-01-V0.
P300-SORT.
  PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL SORT-LENGTH-21 ...
P400-OUTPUT.
  OUTPUT PERM-01-V0.
    
```

---

### NOTES

---

## Sort key options, continued

### Report totals

For a report to have totals, you decide what is to be totaled by extracting certain fields as total fields. The sort key option commands are used to designate when the report totals print. So, if you expect your report to have totals, you must:

- Associate a sort key level field with a special sort key option command.
- OUTPUT the item(s) to be totaled in the report data extract record.
- Designate the total field's PRINT TOTAL field as a 'Y' or 'T' in the RTEDIT layout.

### Employee counts

Employee count and field totals are accumulated when a designated control break is processed by the RTPRNT print program. The employee detail is output to the extract record. To count employees, output a PERM or TEMP counter initialized to 1. A total is printed at every designated control break.

### Line advancing

The sort key options that affect line control are used in conjunction with the totaling commands. When a sort key field has line control association, totaling is assumed at that level. If a total is not desired at that point, you must specify it using a sort key option.

# 1A-RPT Sample and Sort Syntax

**1A-RPT Example:** PRINT '11A-R' FORMS/REPORT-CODE  
 PRINT-GRAND-TOTAL  
 DOUBLE-SPACE-BEFORE CONTROL-1 BREAK-DEFAULT  
 SORT-LENGTH-42 CONTROL-2  
 EMPLOYEE-NAME-25 EMPLOYEE-NO.

CORPORATION 99	ACME MANUFACTURING	ALPHABETIC LISTING OF ACTIVE					REPT	PAGE 2	
DIVISION 9999	PRODUCTION CTL 1-2	AND INACTIVE EMPLOYEES					1A-R	TIME 09:50:42 DATE 09-18-1992	
EMPLOYEE-NAME	EMPLOYEE NUMBER	CTRL THREE	CTRL FOUR	CTRL FIVE	CTRL SIX	MAIL DISTRIBUTION	HIRE DATE	EMPLOYEE STATUS	
MORRIS, ROBERT	1005	3030	4040	5050	6060	PLANT-24	08-01-84	03 ACTIVE-HOURLY-REG-F/T	
MORSE, GORDAN	1004	3388	4488	5508	6608	10TH-5500	08-30-84	03 ACTIVE-HOURLY-REG-F/T	
PENDARVIS, MARTIN M.	2010	3388	4488	5508	6608	PLANT-24	09-06-84	03 ACTIVE-HOURLY-REG-F/T	
PITARO, JOSEPH C.	2011	3030	4040	5050	6060	WH-100	01-03-86	03 ACTIVE-HOURLY-REG-F/T	
PRESCOTT, KEVIN	1313	3333	4444	5555	6666	B3-RECV	02-15-86	03 ACTIVE-HOURLY-REG-F/T	
REYNOLDS, BRENDA	2001	3388	4488	5508	6608	11TH-4040	05-20-84	03 ACTIVE-HOURLY-REG-F/T	
RITTER, BRENDA	6010	3388	4488	5508	6608	11TH-4040	04-15-85	01 ACTIVE-SALRD-REG-F/T	
SANDERS, STEVEN S	6005	3030	4040	5050	6060	5FLR-10W	02-01-75	01 ACTIVE-SALRD-REG-F/T	
SANTANA, LOUISE	3016	3388	4488	5508	6608	11TH-4002	07-22-80	01 ACTIVE-SALRD-REG-F/T	
SCHAEFER, JOANNA S	3012	3388	4488	5508	6608	ACCT-100	06-15-82	01 ACTIVE-SALRD-REG-F/T	
*CONTROL-1	99	78							
*REPORT CODE	1A-R	78							

## NOTES

## Sort key options, continued

### 1A-RPT

The 1A-RPT prints a grand total that is preceded by a double space and a default control break when the Org Level 1 changes. The report is sorted by organization number (Control 1-2), employee name, and employee number order.

General notes on sort keys:

- The sort key must be unique to print detail. In this example, the employee number (partial) is appended to the employee's name (partial) to make it unique.
- The Org Level 1 (Control-1) and Org Level 2 (Control-2) can be expressed as CONTROL-1-2. Only when separate control breaks are required on either of the levels is the separate naming required.

# 1I-RPT Sample and Sort Syntax

**11-RPT Example:** PRINT '11I-R' FORMS/REPORT-CODE  
**PRINT-GRAND-TOTAL**  
**DOUBLE-SPACE-BEFORE CONTROL-1 BREAK-DEFAULT**  
**CONTROL-2 BREAK-DEFAULT**  
**CLP-TYPE NEW-PAGE-BEFORE NO-PRINT-SUBTOTAL**  
**SORT-LENGTH-40 EMPLOYEE-NAME-25.**

CORPORATION 99		ACME MANUFACTURING		UNEXPIRED CERTIFICATES, LICENSES			REPT		PAGE 11	
DIVISION 9999		PRODUCTION CTL 1-2		AND PERMITS REPORT			1I-R		TIME 09:50:42 DATE 09-18-2002	
CLP			EMPLOYEE	CTRL	CLP	CLP DATE	CLP DATE	CLP	EXPIRE	
TYPE	CRT/LICENSE/PERMIT	EMPLOYEE NAME	NUMBER	FOUR	NUMBER	RECORDED	EXPIRED	STATUS	IND	
R030	REAL-ESTATE-SALES	MOORE, SAMUEL	1002	4488	NJDIST3101	10-12-1985				
-----										
CORPORATION 99		ACME MANUFACTURING		UNEXPIRED CERTIFICATES, LICENSES			REPT		PAGE 13	
DIVISION 9999		PRODUCTION CTL 1-2		AND PERMITS REPORT			1I-R		TIME 09:50:42 DATE 09-18-2002	
CLP			EMPLOYEE	CTRL	CLP	CLP DATE	CLP DATE	CLP	EXPIRE	
TYPE	CRT/LICENSE/PERMIT	EMPLOYEE NAME	NUMBER	FOUR	NUMBER	RECORDED	EXPIRED	STATUS	IND	
*CONTROL-2	9999	19								
*CONTROL-1	99	19								
*REPORT CODE	1I-R	19								

## NOTES

## Sort key options, continued

### 1I-RPT

The 1I-RPT contains the following sort and totaling detail:

- The report is sorted by organization number (Control 1-2), CLP-Type, and employee number order.
- A grand total is printed, preceded by a double space.
- The Org level 1 (Control-1) and Org Level 2 (Control-2) are printed. There is no special line spacing associated with them.
- Each page contains the detail for a specific CLP Type. No totals are given on this level because of the NO-PRINT-SUBTOTAL keyword. If this was not present, a subtotal would be assumed at the CLP-TYPE level.

# XK-RPT Sort Syntax

**XK-RPT Example:** OUTPUT '1XK-R' FORMS/REPORT-CODE **NO-PRINT-GRAND-TOTAL**  
 CONTROL-1 **NO-PRINT-SUBTOTAL** CONTROL-2 **NO-PRINT-SUBTOTAL**  
 PLAN-ID **NO-PRINT-SUBTOTAL** NEW-PAGE-AFTER  
 EMPLOYEE-NAME-15 **NO-PRINT-SUBTOTAL** EMPLOYEE-NUMBER  
**TRIPLE-SPACE-BEFORE** **NO-PRINT-SUBTOTAL** SORT-LENGTH-42  
 DEPENDENT-1.

CORPORATION 99 ACME MANUFACTURING				SPOUSE/DEPENDENT CHANGES				REPT	FILE VERSION 00 PAGE 1			
DIVISION 9999 PRODUCTION CTL 1-2				BY PLAN				XK-R	TIME 09:14:46 DATE 10-23-2002			
PLAN	EMPLOYEE	DEPENDENT	DEP	BIRTH	F/T	INSURANCE						
ID	EMPLOYEE NAME	NUMBER	OPTION	DATE	XREF	DEPENDENT NAME	SEX	DATE	STDT	RELATION	CARRIER	
101	BLOOM, ALEXANDER	3001	EMPLOYEE-&	01-01-1983	001	BLOOM, BRENDA	F	12-24-1940	N	WIFE	CONN.GENER	
101	BLOOM, ALEXANDER	3001	EMPLOYEE-&	01-01-1983	002	BLOOM, ABAGAIL	F	01-17-1915	N	MOTHER		
101	BLOOM, ALEXANDER	3001	EMPLOYEE-&	01-01-1983	003	BLOOM, ELLEN	F	03-24-1972		DAUGHTER		
101	BLOOM, ALEXANDER	3001	EMPLOYEE-&	01-01-1983	004	BLOOM, THOMAS J	M	08-16-1975		SON		
101	BLOOM, ALEXANDER	3001	EMPLOYEE-&	01-01-1983	005	BLOOM, WILLIAM	M	06-14-1978		SON		
101	SPENSER, WILLIAM	3024	EMPLOYEE-&	01-01-1983	001	SPENSER, DENISE	F	06-26-1961		WIFE	PRUDENTL	
101	SPENSER, WILLIAM	3024	EMPLOYEE-&	01-01-1983	002	SPENSER, ASHLEY	F	05-02-1983		DAUGHTER		
101	SWALTER, STEVEN Y	3005	EMPLYE&SPS	07-22-1983	001	SWALTER, SHERRIE	F	10-22-1948	N	WIFE	ATENA	
101	SWALTER, STEVEN Y	3005	EMPLYE&SPS	07-22-1983	002	SWALTER, CHRIS	M	06-08-1973		SON		
101	TEACHEN, JUDITH	3009	SINGLE/EMP	12-01-1984	001	TEACHEN, ROSE	F	11-21-1917		MOTHER	MEDICARE	
101	TEACHEN, JUDITH	3009	SINGLE/EMP	12-01-1984	002	MURPHY, MARGARET	F	01-17-1915		AUNT	MEDICARE	

## NOTES

## Sort key options, continued

### XK-RPT

Here is a detailed explanation of the sort syntax for XK-RPT program:

- No totals are required. Control breaking on the Plan ID and employee name requires sort key options on the control levels that precede these fields in the sort key.
- There are no totals or special line advancing requirements for the report code or organization levels (Control 1-2). Note that the Org Level 1 and 2 could alternately be named CONTROL-1-2 in this example.
- Each Plan ID in the report is on a new page. No totaling is required. The sort key option for new page is coded, and then the line advance override is coded.
- The next sort order is on 15 characters of the employee's name. No control break is needed here.
- Triple spacing is designated for each new employee number. No totaling or control break description is required. If NO-PRINT-SUBTOTAL is not coded here, a report line would be produced as follows: \*EMPLOYEE-NUMBER  
1111.
- The sort length designation includes the final field in the sort order, DEPENDENT-1.

# Sort Key Length and Placement

## 1A-RPT Sort Fields and Sort Length

FORMS/REPORT-CODE	5
SORT-LENGTH-42	
CONTROL-1	2
CONTROL-2	4
EMPLOYEE-NAME-25	25
EMPLOYEE-NO	<u>6</u>
Total	42

---

## NOTES

---

## Sort key length and placement

### Sort key length

As you recall, the length of the sort key is important during the print phase. The Report Print (RTPRNT) program uses this piece of information to determine where in the extract record the sort key ends and the report data starts. The assumption is that the first character following the sort length is the record type. Then, the Report Print Position Edit (RTEDIT) guidelines are applied to the extract record data.

Therefore, if the sort length is incorrect, the printed result is unpredictable.

### 1A-RPT example

The length of the sort portion or key of the extract record must be manually calculated. Recall that you must count the length of each item in your sort key. Using the 1A-RPT, the sort fields total a length of 42.

# Sort Key Placement—Multiple Control Breaks

**1A-RPT Example:** PRINT '11A-R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
DOUBLE-SPACE-BEFORE CONTROL-1 BREAK-DEFAULT  
**SORT-LENGTH-42** CONTROL-2  
EMPLOYEE-NAME-25 EMPLOYEE-NO.

**1I-RPT Example:** PRINT '11I-R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
DOUBLE-SPACE-BEFORE CONTROL-1 BREAK-DEFAULT  
CONTROL-2 BREAK-DEFAULT  
CLP-TYPE NEW-PAGE-BEFORE NO-PRINT-SUBTOTAL  
**SORT-LENGTH-40** EMPLOYEE-NAME-25.

**XK-RPT Example:** OUTPUT '1XK-R' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL  
CONTROL-1 NO-PRINT-SUBTOTAL CONTROL-2  
NO-PRINT-SUBTOTAL  
PLAN-ID NO-PRINT-SUBTOTAL NEW-PAGE-AFTER  
EMPLOYEE-NAME-15 NO-PRINT-SUBTOTAL EMPLOYEE-NUMBER  
TRIPLE-SPACE-BEFORE NO-PRINT-SUBTOTAL **SORT-LENGTH-42**  
DEPENDENT-1.

---

## NOTES

---

## Sort key length and placement, continued

### Sort key placement

Another important consideration is the placement of the SORT-LENGTH-*nn* verb in the extract program.

- Only the fields defined before the SORT-LENGTH-*nn* verb may be used to produce control breaks.
- SORT-LENGTH-*nn* MUST follow the last sort key option command.
- SORT-LENGTH-*nn* ALWAYS includes the length of all sort key fields whether they precede or succeed the command.

# Section 5 Exercise 1

CORPORATION 99	ACME MANUFACTURING	CONTROL BREAK EXAMPLE	REPT PERIOD	FILE VERSION 00	PAGE 4
DIVISION 9999	PRODUCTION CTL 1-2	TOTALS AND CARRIAGE CONTROL	ANYR PERIOD	TIME 09:38:54	DATE 03-16-2002
FREQ	EMPLOYEE NAME	EMPLOYEE NUMBER			
4	BARTHOLOW III, JONATHAN	1113			
4	JOHNSEN, RICH DANIEL	1112			
4	WELKER, GEORGE W	1114			
*CONTROL-4-CODE ADM 3					
4	MORSE, GORDAN	1004			
*CONTROL-4-CODE MANU 1					
4	MOORE, SAMUEL	1002			
4	MORITZ, KATHERINE C.	1007			
*CONTROL-4-CODE SALE 2					
CORPORATION 99	ACME MANUFACTURING	CONTROL BREAK EXAMPLE	REPT PERIOD	FILE VERSION 00	PAGE 5
DIVISION 9999	PRODUCTION CTL 1-2	TOTALS AND CARRIAGE CONTROL	ANYR PERIOD	TIME 09:38:54	DATE 03-16-2002
FREQ	EMPLOYEE	NAME	EMPLOYEE NUMBER		
*REPORT CODE	ANYR		6		

## NOTES

## Section 5 exercise 1

### Purpose

The purpose of this exercise is to give you practice coding sort logic for multiple control breaks. Directions assume an employee count at the control breaks indicated. Use the space below.

Sort Order	Total Break	Carriage Control Override
FORMS/REPORT-CODE	YES	TRIPLE SPACE BEFORE
CONTROL-1-2	NO	NO
PAY-FREQUENCY-CODE	NO	NEW PAGE BEFORE
CONTROL-4-CODE	YES	DOUBLE SPACE BEFORE DOUBLE SPACE AFTER
EMPLOYEE-NAME-15	NO	NO
EMPLOYEE-NUMBER	NO	NO

# Work Fields as Control Breaks

CORPORATION 99 ACME MANUFACTURING		SORT EXAMPLE USING WORK FIELDS		REPT PERIOD	FILE VERSION 00	PAGE 2
DIVISION 9999 PRODUCTION CTL 1-2				XS-R PERIOD	TIME 12:10:27	DATE 11-03-2002
EMPLOYEE NUMBER	EMPLOYEE NAME	SALARY GRADE	ANNUAL SALARY			
1764	HANCOCK, STEVEN W.	09	19,200.00			
2003	CMEYLA, JANE	02	13,340.00			
2006	COSTELLO, SUSANNE	03	18,280.08			
2008	HALL, RHONDA D.	07	1,662,970.40			
2010	PENDARVIS, MARTIN M.	05	16,380.00			
2013	ANDREWS, HENRY A.	02	16,900.00			
2014	GRANT, KEITH L.	05	19,760.00			
2016	SHEA, JEFFERY B.	05	20,696.00			
3004	MARSH, PAUL J.	09	15,600.00			
3005	SWALTER, STEVEN Y	09	18,322.20			
3007	HILLERY, THOMAS	09	19,200.00			
3012	SCHAEFER, JOANNA S.	07	19,240.00			
3013	CARLILE, WILLIAM E.	09	21,450.00			
3014	SULLIVAN, MIKE M.	04	11,544.00			
3019	HARRIS, CECELIA	04	15,990.00			
3022	WENDT, GARY D.	06	19,760.00			
3023	DANIELS, JEFFREY C.	08	21,619.00			
3024	SPENSER, WILLIAM M.	05	16,634.80			
3025	WILSON, BARBARA	07	20,800.00			
3029	DUNBAR, WALLCOTT A.	08	25,896.00			
3033	YOUNG, J.T.	08	23,904.00			
6004	ANDERSEN, KARI	09	15,600.00			
6005	SANDERS, STEVEN S	09	18,322.20			
6007	MILLER, THOMAS S	09	19,200.00			

\*W8-01-850GTYPE B

## NOTES

## Special sort techniques

### Control break descriptions

The labels that appear when a control break occurs are derived from the field names used to build the sort key.

- Only the 1st fifteen positions of the field name are used when printing a control break description. For example, the control break for JOB–TITLE–FIRST20 would appear as: **\*JOB–TITLE–FIRST**.

### Work fields

Work fields (Pointer 6, 7, 8, or 44 field referenced as Wn–00–000) can be used as sort key items. However, when a control break occurs, the control break description appears using the work field name as the description.

The above example uses work fields to group employees by their salary grade type. Group A is for annual types, B is for per pay period types and C represents hourly grades. When the control break occurs the description appears as **\*W8–01–850GTYPE**.

### Sort field limit

Regardless of the length of each individual sort key item, or the total sort key length, no more than nine control breaks can be defined for a report.

# Consolidating Organization Numbers (Control 1–2s) on a Report

**Example 1:** Consolidate ALL Organization Numbers (Control 1–2s) using alphanumeric characters

```
PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
'AAAAAA' ... @Dummy Control 1-2
```

**Example 2:** Consolidate ALL Organization Numbers (Control 1–2s) using literals of spaces

```
PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
'      ' ..... @Dummy Control 1-2
```

**Example 3:** Consolidate ALL Organization Numbers (Control 1–2s) using SPACE-OVER :06

```
PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
SPACE-OVER :06 ... @Dummy Control 1-2
```

**Example 4:** Consolidate Multiple Organization Numbers (Control 1–2s) using literal of spaces

```
PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
CONTROL-1 '      ' ..... @Dummy Control 2
```

---

## NOTES

---

## Special sort techniques, continued

### Consolidating organization numbers (Control 1–2s)

The Report Print (RTPRNT) program expects the extract record's format to include the forms code, report code, and organization number (Control 1–2) value. Therefore, all reports are sorted by report and organization number. However, it may be desirable to consolidate reports for all organization numbers or for specific organization numbers.

To override the requirement for the organization number value in the sort key, it will be necessary to 'dummy out' the organization number value when building the extract record's sort key.

To consolidate all or multiple organization numbers:

- Replace the organization number with six alphanumeric characters or spaces.
- Replace the portions of the organization number with alphanumeric characters or spaces.

*Note:* Using this technique to consolidate organization numbers causes only the organization number information for the 1st or highest collated value organization number scheduled to be printed in the report heading.

# Multiple Extract Records For Employee

```

1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 0 0 1 1 2 2 3 3 4
1...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0...5...0
11XXR9955551234567890A0LARKIN, LIBBY          9955551111NORTHEAST MANUF222MILLERSBURG PLA3333CARTON ASSEMBLY 4444POST PRODUCTION
11XXR9955551234567890B 002200000040000053000F101-10-196210-19-193011XXR9955551234567890B 002200000040000053000F162-01-1030-10-19
11XXR99999921025302 A0PATERSON, DOROTHY      9999991010CORP OFFICE 2020MERCERSBURG PLA3030ACCOUNTING 4040TAX
11XXR99999921025302 B 003500000086670199000F210-11-198405-25-1962
11XXR99999921025302 B 003500000086670199000F284-10-1162-05-25

```

Sort Key: P200--SORT.  
 PRINT '1XX-R' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL  
 SORT-LENGTH-21 CONTROL-1 CONTROL-2 EMPLOYEE-NUMBER.

Report Data P300-EXTRACT-RECORD-A.  
 Record Type A: PRINT 'A0' EMPLOYEE-NAME-25 CONTROL-1-2.  
 FIND CONTROL-3-CODE.  
 IF NOT-FOUND  
 SPACE-OVER :60 SPACE-OVER :16  
 ELSE  
 PRINT CONTROL-3-CODE CONTROL-3 CONTROL-4-CODE CONTROL-4  
 CONTROL-5-CODE CONTROL-5 CONTROL-6-CODE CONTROL-6.  
 WRITE-EXTRACT.

Record Type B: P310-EXTRACT-RECORD-B.  
 SET SCREEN TO :1622.  
 PRINT 'B ' .  
 FIND-SALARY.  
 IF NOT-FOUND  
 OUTPUT '000000000000000000000000'  
 ELSE  
 OUTPUT ANNUAL-SALARY HOURS-PER-PERIOD HOURLY-RATE.  
 PRINT SEX-CODE PAY-FREQUENCY-CODE EMPLOYMENT-DATE BIRTH-DATE.  
 WRITE-EXTRACT.

---

## NOTES

---

## Special sort techniques, continued

### Multiple extract

Due to your sort key length or the amount of data to be included in a report, one 150-byte record may not be sufficient. Therefore, it is possible to output multiple 150 byte records per employee.

### Sort key logic

Each record written for a given employee must be unique. This means that either the sort key or the record type is unique. There are two techniques used to build the sort key portion of the record:

- Re-execute the sort key logic for each extract record
- Reuse the sort key from the previous record written as it is in working storage

To re-execute the sort key logic, simply establish this as a unique sort key paragraph. Perform the routine each time an extract record is built. An example of this follows.

### XX-RPT example

As you can see, the XX-RPT writes two different extract records for each employee. Record Types A and B are written. Only one execution of the sort logic occurs.

Each time the extract record is formatted, the record is placed in the working storage SCREEN area. Since the beginning address of SCREEN for the extract record build is always :1601, the length of the sort key is added to this. For XX-RPT, it is set to 1622 (1601 + 21).

# 1K-RPT Sample and Sort Syntax

**Example:** P105-FORMAT-SORT.  
 SPACE-EXTRACT-RECORD. PRINT '11K-R'  
 FORMS/REPORT-CODE  
 NO-PRINT-GRAND-TOTAL SORT-LENGTH-15 CONTROL-1  
 CONTROL-2  
**SORT-KEY-SEPARATOR.**  
 CALCULATE SORT-KEY-SEPARATOR + :1 =  
 SORT-KEY-SEPARATOR.  
 EXIT.

CORPORATION 99 ACME MANUFACTURING				EMPLOYEE PROFILE				REPT	NAME: MEYER, JUNE				PAGE	1			
DIVISION 9999 PRODUCTION CTL 1-2				HUMAN RESOURCE BASE MODULE				1K-RPT	EMPLOYEE NUMBER: 1001				DATE 09-17-2002				
SC	SOCIAL	PAY	PAY	STATUS	SEX	EEO	UNION	WORKERS	BIRTH	EMPLMNT	TERMINATN	SHIFT	SPLIT	JOB	USER	FAIR	PERIOD
	SECURITY	FREQ	CODE	CODE	CODE	RACE	CODE	COMP	DATE	CD DATE	CD DATE	CODE	CODE	CATEGORY	FLD	LABOR	VERRIDE
E	888-88-8001	1	6		F	01			07-11-56	09-15-84		0	1				00
SC	EMPLOYEE NAME				ADDRESS LINE 1				ADDRESS LINE 2				CITY/STATE		ZIP		
F1	MEYER, JUNE				1010 MISTY LANE				UNIT 4				EL SEGUNDO, CA		93101		
					AREA		PHONE										
O4	001			MEYER, DOROTHY	203		222-9090		05 17 S. BREEZE DRIVE					EL SEGUNDO, CA			93101
SC	KEY				PHYSICIAN NAME				CODE	NUMBER	SC PHYSICIAN ADDRESS				CITY/STATE	ZIP	
																	CODE
O6	001			BLACK, DR. RICHARD	203		224-8854		07 1 MEDICAL CENTER					EL SEGUNDO, CA			93101
	BASIC-DATA		PREVIOUS	HIRE	AGENCY		RELOCATION		---	HOME	---		---	WORK	---		
SC	DATE	APPLICANT ID	SOURCE	FEE	EXPENSE	AREA	PHONE	AREA	PHONE	EXTENSION	RELIGION	LANGUAGE					
ZA	09-15-84	25000-99	01	01000		203	2221282	201	8369400	1567	EB	E1					
	RECORD	MARITAL	TOTAL	CITIZENSHIP	VISA	VISA	MILITARY	SECURITY									
SC	DATE	CODE	DEPENDENTS	CODE	TYPE	EXPIRATION	STATUS	CLEARANCE	CLEARING AGENCY								
ZB	09-15-84	S															

## NOTES

---

## Special sort techniques, continued

### Sort logic

The 1K-RPT is an exhaustive printout profiling the employee from a human resource standpoint. This report uses the technique of re-executing the sort logic to differentiate each record written. The report has intermittent headings for each section or segment type reported and can produce multiple pages per employee.

### Space extract

The SPACE-EXTRACT-RECORD is used to move spaces to the 150-character extract record area used to build a record for a packaged report (SCREEN starting at :1601).

### Sort key separator

Each record produced by this program must have a unique sort key considering the repeated use of Record Type '0'. A simple technique is to employ the field SORT-KEY-SEPARATOR in the sort logic. Using a CALCULATE statement, increment this value each time a record is written.

Caution: SORT-KEY-SEPARATOR is defined in Pointer 6 and may interfere with report parameters.



*Refer to Section 7: Reports—Special Functions for additional information about the use of Report Group input parameters.*

## Section Summary

- **Sort key options**
- **Sort key length**
- **Sort length placement**
- **Special sort techniques**

---

NOTES

---

## Section summary

In this section, you learned techniques for building an extract record with complex sort key logic.

### Sort key options

- The only required sort commands are: \_\_\_\_\_ or \_\_\_\_\_.
- To produce a control break with no special spacing, use the \_\_\_\_\_ verb.
- When using the carriage control override options, a total break is assumed. To suppress a total on a control break, use the \_\_\_\_\_ verb.
- The Sort Key Option(s) always \_\_\_\_\_ the field that they apply to.
- The \_\_\_\_\_ verb always follows the fields that have Sort Key Option specified. The length of the Sort Key is based on fields both before and after the \_\_\_\_\_ verb.

### Special sort techniques

- Regardless of the length of each field, or the total sort key length, no more than \_\_\_\_\_ control breaks can be defined for a report.
- A control break description is limited to \_\_\_\_\_ characters of the field's name.
- When a consolidated report is desired, a dummy \_\_\_\_\_ value must replace the \_\_\_\_\_ value in the extract record.

## Section 5 Exercise 2

---

**NOTES**

---

## Section 5 exercise 2

Copy the program created in Sections 3 and 4, and rename it. Modify the program to match the following specifications:

### Headings

In addition to the default headers, the title is Class Report and the subtitle is Pay Frequency Totals.

### Selection

Display only Non-union employees (UNION-CODE blank).

### Extract

Use the following field names to create either Sort and/or Report data. The report detail should be double-spaced.

#### **SORT KEY:**

FORMS/REPORT-CODE  
CONTROL-1-2  
PAY-FREQUENCY-CODE  
EMPLOYEE-NAME-20  
EMPLOYEE-NUMBER

#### **REPORT DATA:**

EMPLOYEE-NAME  
EMPLOYEE-NUMBER  
PAY-FREQUENCY-CODE  
PAY-FREQUENCY  
PAYMENT-CODE  
PAYMENT-TYPE  
SALARY (H001)  
EMPLOYMENT-DATE  
BIRTH-DATE  
Perm Counter (print position 040)  
used to count employees

### Control breaks

#### **What**

Perm Counter  
SALARY

#### **When**

Grand Total, Pay Frequency Total  
Grand Total, Pay Frequency Total

Use your own discretion in providing line advance overrides at control breaks.

---

**NOTES**

---

## Section 6: Report Basics Plus

---

### Table of Contents

Introduction.....	161
Run-time input parameters.....	163
Cyborg Scripting Language basics plus.....	171
Section 6 exercise 1 .....	181
Segment/Pointer manipulation.....	183
Accessing table data.....	191
Section summary.....	195
Section 6 exercise 2 .....	197

## Objectives

- **Access report schedule parameters**
- **Apply Cyborg Scripting Language basics to report writing**
- **Control/manipulate segment addressing**
- **Access table data**

---

### NOTES

---

## Introduction

### **Purpose**

In this section, you will learn additional CSL techniques related to report programming.

### **Objectives**

Upon completion of this section you will be able to:

- Access report schedule parameters
- Apply CSL basics to report writing
- Control/manipulate segment addressing
- Access table data

## Using Run-time Parameters

- **Enter parameters in the schedule report groups parameter selection form**
- **Allocate working storage for the parameters**
- **Include report parameter entry form name**
- **Use the parameters for employee selection**

---

### NOTES

---

## Run-time input parameters

### RGMSTR parameters

Input parameters are entered on the Report Group parameter entry form for each report. Up to 36 positions of parameter fields may be entered per report. Your report logic must be designed to address the input parameter work area and also name the parameter entry form program.

To use input parameters successfully in a report you must:

- Enter parameters in the Parameter Entry form accessed through the Schedule Report Groups (RGMSTR) Parameter button
- Allocate working storage for the parameters
- Include the Parameter Entry form program name
- Use the parameters for employee selection or other reporting options

### Document parameters

Be sure to document the use of the run-time parameters in the report documentation.

# Report Schedule Input Allocation

## **ALLOCATE–nn**

*DEFINE–REPORT ALLOCATE–nn.*

**Example:** Allocating 10 characters of input parameters.  
DEFINE–REPORT ALLOCATE–10.  
Result: The report logic can address columns 1–10 of the input parameters as a single parameter or separately.

<b>ALLOCATE VERB</b>	<b>DEFINITION</b>
ALLOCATE–DATE	Address the first six positions of the input parameter as a date in CYYMDD format.
ALLOCATE–nn	Address a number of positions of the parameters where nn is from 01–36 characters of input parameter values.

---

## **NOTES**

---

## Run-time input parameters, continued

### **ALLOCATE**

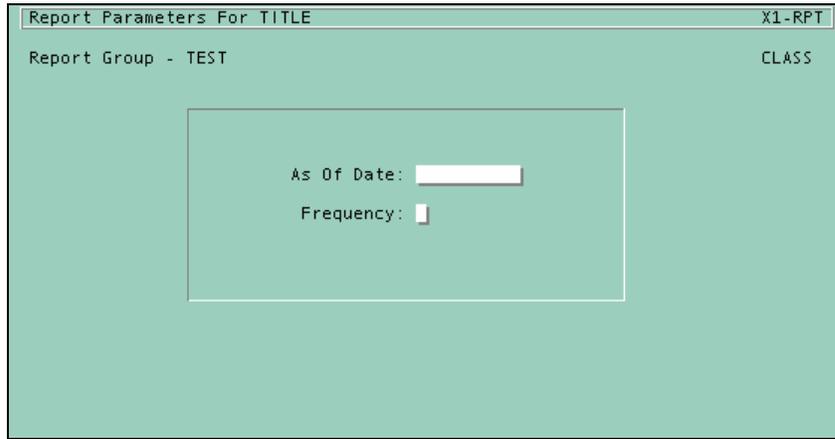
The **ALLOCATE** verb is used to address the parameter(s). **ALLOCATE** is a **DEFINE-REPORT** option and must be coded with that verb.

- If a date is the only input parameter, use the **ALLOCATE-DATE** verb. The input data must be a six-character date in **CYYMDD** format.
- If more than six characters of the parameters are used (perhaps including a date), use **ALLOCATE-*nn*** where ***nn*** equals a number in the range of 01–36. This number represents the number of positions filled by the input parameters.

For example, if the **EMPLOYEE-NUMBER** field is used as a run-time parameter, **ALLOCATE-10** is required to allocate working storage for the length of the field.

**ALLOCATE-06** and **ALLOCATE-DATE** are synonymous when a date is the parameter.

# Input Parameters—Pointer 6 Area



Example: P200-SELECTION.  
IF EMPLOYMENT-DATE IS LESS THAN SPECIAL-DATE  
RETURN.  
IF PAY-FREQUENCY IS NOT EQUAL TO W6-01-042FREQ  
RETURN.

Pointer 6: 

3	4	4	5	5	6	6	7							
6	...	0	...	5	...	0	...	5	...	0	...	5	...	0
207C313														

---

## NOTES

---

## Run-time input parameters, continued

### Parameter entry form

Run-time parameters are entered through a Parameter Entry form. There are standard forms available (GRPxxx) or customized may be developed. Line #00002 of the extract program indicates the name of the form to be accessed when the Set Parameters button is pushed. The format is:

@PARMS=GRPxxx

Customized parameter entry forms are named R-xxxx where xxxx is the first four characters of the report name. If one of those characters is a '-', e.g., X3-RPT, the dash is dropped. The parameter entry form would be R-X3R.

*Note:* *More information on developing parameter entry forms is covered in the Customizing Solution Series Fields, Forms, and Queries documentation.*

### Pointer 6

The run-time parameters entered on the Report Group parameter entry form are stored in Pointer 6 starting at displacement 036 during the report execution.

- For example, SPECIAL-DATE is the same as referencing the data using W6-06-036 if the first parameter is a date.
- All dates entered will be in century format in Pointer 6

*Note:* *TEMP Counters are addressed in this same area. Take special care to avoid addressing the same area when using both features in a program. Use the PERM Counters first.*

# Report Process Timing

FIRST-TIME-IND VALUE	FIRST-TIME-IND USE
F (system maintained)	First Time (per Organization Number [Control-1-2]) for this report's execution.
N (system maintained)	Next time (second and subsequent employee in a Organization Number [Control-1-2]) for this report's execution.
Y (report program maintained)	Run-time parameters in Pointer 6 have been altered during this report's execution.

**Allocation:** DEFINE-REPORT NO-PE-DATES ALLOCATE-07.

**Parameter** P100-INITIAL.

**Conversion:** IF FIRST-TIME-IND EQUAL 'F'  
 IF SPACES EQUAL W6-06-036DATE  
 MOVE 'Y' TO W6-01-035  
 MOVE CURRENT-DATE-CYYMDD TO W6-06-036DATE.

**Selection:** P110-SELECT-EMPLOYEE.  
 FIND-ACTIVITY.  
 IF NOT-FOUND RETURN.  
 IF ACTIVITY-DATE GREATER THAN SPECIAL-DATE  
 RETURN.

---

## NOTES

---

## Run-time input parameters, continued

### First time indicator

Your program should edit input parameters. It is best to check the input parameters for each company the first time the program executes. This is typically done at initialization time.

### Report timing

The value of the **FIRST-TIME-IND** field can be tested to determine the timing for your report (see chart). This field is often referenced as W6-01-035.

The core system maintains the 'F' and 'N' values. If your program dynamically changes the contents of the input parameter(s), you must set this switch value to a 'Y'. This action causes the parameter to be rewritten to another core system work area for the continuation of the extract program's processing for that company.

### Example

In the example, editing is done in the first time pass. When blank, the run-time parameter date **SPECIAL-DATE** is substituted with **CURRENT-DATE-CYYMDD**.

# Cyborg Scripting Language Basics Plus

## MOVE

$$\underline{\text{MOVE}} \left\{ \begin{array}{l} \text{literal} \\ \text{field name 1} \\ \text{figurativeconstant} \end{array} \right\} \underline{\text{TO}} \text{fieldname2.}$$

## CALCULATE

$$\underline{\text{CALCULATE}} \left\{ \begin{array}{l} \text{literal} \\ \text{fieldname 1} \end{array} \right\} \left[ \begin{array}{l} + \\ * \\ / \end{array} \right] \left\{ \begin{array}{l} \text{literal} \\ \text{fieldname 2} \end{array} \right\} = \text{fieldname3}$$

## FIND

$$\underline{\text{FIND}} \text{nonkey field 1} \left[ \begin{array}{l} \text{FROM HERE} \\ \text{STARTINGWITH} \left\{ \begin{array}{l} \text{literal} \\ \text{fieldname 1} \end{array} \right\} \end{array} \right]$$

## PROCESS

$$\underline{\text{PROCESS}} \text{nonkey field 1} \left[ \begin{array}{l} \text{FROM HERE} \\ \text{STARTINGWITH} \left\{ \begin{array}{l} \text{literal} \\ \text{field 1} \end{array} \right\} \\ \text{ENDINGWITH} \left\{ \begin{array}{l} \text{literal} \\ \text{field 2} \end{array} \right\} \end{array} \right]$$

*imperative statement ...*

END-PROCESS.

---

## NOTES

---

## Cyborg Scripting Language basics plus

### Basic skills

When writing report programs, the following tasks are often required in manipulating data from the employee database for your extract record:

- Accessing dated segments
- Calculating time spans
- Establishing loop logic through segment occurrences

### Basic verbs

These functions have been covered in previous courses, but require a thorough review since they are used quite extensively in reports. The verbs to be discussed include:

- FIND/MATCH–SEGMENT–CODE
- CALCULATE (for Time Spans)
- PROCESS/END–PROCESS

## Date Conversion—Work Fields

**Example 1:** IF ACTIVITY-DATE EQUALS EMPLOYMENT-DATE  
GO TO P200-REPORT-OK.

**Example 2:** IF EMPLOYMENT-DATE NOT EQUAL ACTIVITY-DATE  
GO TO P300-REPORT-IN-ERROR.

REGULAR DATE FORMAT	CENTURY FORMAT
CENTURY-SAVE-DATE	HOLD-DATE
SAVE-DATE	WORK-DATE
SAVE-DATE-2	WORK-DATE-CENTURY
WORK-DATE-YY6	WORK-DATE-YEAR
WORK-DATE-YYMMDD	WORK-DATE-MONTH
CURRENT-DATE	WORK-DATE-DAY
SPECIAL-YYMMDD	WORK-DATE-CYYMDD
SPECIAL-YYMMDD-2	CURRENT-DATE-CYYMDD
	SPECIAL-DATE
	SPECIAL-DATE-2

---

### NOTES

---

## CSL basics plus, continued

### Date fields

Date comparisons and calculations are often used in reports. There are several issues to consider when working with date fields:

- Dates are stored in four possible formats:
  - Century Date: CYYMDD format
  - Regular Date: YYMMDD format or CCYYMMDD format
  - Time Span: YYMMDD format
  - Year Only: NN format
- Date comparisons may be done between date fields of any format. Both dates are converted to CCYYMMDD format prior to the compare.

### Use of MOVE

Use the MOVE verb to convert a date from century format to regular format or vice versa. The date is transformed into the receiving field's format.

When working with a date in century format, always try to maintain the full four-digit year when it is converted to regular date format. The use of CENTURY-SAVE-DATE puts the date in CCYYMMDD format. This is particularly useful when working with BIRTH-DATE.

## FIND Verb—Dated Segments

**Example 1:** FIND JOB-CODE.  
IF NOT-FOUND RETURN.  
IF JOB-EFFECTIVE GREATER THAN CURRENT-DATE-CYYMDD  
RETURN. @Future  
    OUTPUT JOB-EFFECTIVE JOB-CODE.

**Example 2:** FIND JOB-CODE STARTING WITH WORK-DATE. @Past  
MATCH-SEGMENT-CODE.  
IF NOT-FOUND RETURN.  
OUTPUT JOB-EFFECTIVE JOB-CODE.

**Example 3:** MOVE SPECIAL-DATE TO SALARY-AS-OF-DATE.  
FIND-SALARY-AS-OF.  
IF NOT FOUND RETURN.  
IF SALARY-EFFECTIVE IS NOT GREATER THAN SPECIAL-DATE-2  
    GO TO P200-PROCESS  
ELSE  
    RETURN.

---

### NOTES

---

## CSL basics plus, continued

### **FIND**

The FIND verb is used to position a segment's pointer to a specific occurrence. However, when using FIND for a segment whose key is a century date, logical comparisons must be analyzed carefully.

#### **Example 1 future date?**

Example 1 is coded to exclude employees who have job related information with a future date. The FIND verb positions the pointer to the first occurrence of the Job Segment. Then the JOB-DATE is compared to the current date to find if the current segment is future dated.

#### **Example 2 past date?**

Example 2 is coded to position the segment pointer as of a specific date found in WORK-DATE. The MATCH-SEGMENT-CODE limits the compare of the IF NOT FOUND to verify only that the current segment has the same segment ID and segment code as the segment being sought.

#### **Example 3 date range**

Example 3 is coded to determine whether the LZF segment falls within a certain date range entered through run-time parameter.

- SALARY-AS-OF-DATE is used with the LZF segment to hold the effective date desired.
- FIND-SALARY-AS-OF positions the pointer to the LZF segment effective as of the given date. (MATCH-SEGMENT-CODE is not used with this verb.)
- The segment data is then compared to see if it is within the range of the run-time parameter dates.

# PROCESS Verb—Dated Segments

**Example 1:** MOVE :0 TO PERM-01-V0.  
PROCESS ABSENT-HOURS STARTING WITH WORK-DATE  
ENDING WITH HOLD-DATE.  
IF ABSENT-TYPE-CODE EQUALS 'U' BYPASS-ENTRY.  
CALCULATE PERM-01-V0 + :1 = PERM-01-V0.  
END-PROCESS.  
OUTPUT PERM-01-V0.

**Example 2:** P110-PROCESS-LOOP.  
PROCESS DISABILITY-CODE STARTING WITH  
CURRENT-DATE-CYMD DD ENDING WITH WORK-DATE.  
IF SPACES EQUAL DISABILITY-CODE RETURN.  
PRINT '15X-R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL  
TRIPLE-SPACE-BEFORE CONTROL-1-2  
DOUBLE-SPACE-BEFORE DISABILITY-CODE  
DOUBLE-SPACE-BEFORE SORT-LENGTH-30.  
OUTPUT EMPLOYEE-NUMBER DISABILITY-DATE.  
PRINT '10' EMPLOYEE-NUMBER EMPLOYEE-NAME-25  
RESTRICTED-WORK-CODE RESTRICTION  
DISABILITY-CODE DISABILITY.  
OUTPUT EMERGENCY-EFFECTIVE DISABILITY-DATE  
PERM-01-V0.  
WRITE-EXTRACT.  
END-PROCESS.

---

## NOTES

---

## CSL basics plus, continued

### PROCESS

Recall that the PROCESS verb is handy when you need to scan through more than one occurrence of a multiple occurrence segment.

- PROCESS **requires** the use of a non-key field in its syntax. Much like the FIND verb, it begins searching at the first occurrence unless otherwise directed.
- Your program may have multiple process loops but do not physically nest them.
- When the process hits END-PROCESS, your program is no longer accessing what was qualified by the PROCESS syntax. For example, if all HEDs have been processed, you are now accessing the J Segment (employee taxes).

### Example 1

Example 1 is coded to process an employee's unexcused absences (LVA Segment). The process is limited to occurrences that happened within a defined time frame. Then, the total number of accumulated absences is extracted for the report.

### Example 2

Example 2 is coded from the standard 5X-RPT program. This report lists disability occurrences (LZM) for the employee. In this case, the process includes:

- The Sort extract
- The Report data extract
- The Write of the extract record

# Date Calculations—Time Spans

**Example 1:** CALCULATE CURRENT-DATE-CYYMDD - BIRTH DATE = AGE.  
PRINT YEARS-OF-AGE.

**Example 2:** MOVE '000600' TO WORK-TIME-SPAN.  
CALCULATE SALARY-EFFECTIVE + WORK-TIME-SPAN = WORK-DATE.

**Example 3:** CALCULATE SAVE-DATE - WORK-DATE = JULIAN-TS-DATE.  
CALCULATE JULIAN-DAYS-1 / :365 = WORK-PCT.  
PRINT WORK-PCT.

Time Span Field	Field Redefinition
AGE	YEARS-OF-AGE
TIME-SPAN	TIME-SPAN-YR
	TIME-SPAN-YRS-MONTHS
WORK-TIME-SPAN	WORK-SPAN-DAYS
	WORK-SPAN-MONTHS
	WORK-SPAN-YEARS
JULIAN-TS-DATE	JULIAN-DAYS-1

---

## NOTES

---

## CSL basics plus, continued

### Date calculations

The CALCULATE verb can be used to derive a span of time between two dates, (for example, Age or length of service).

- Time Spans are **YYMMDD** formatted. The **YY** is the number of years, **MM** is the number of months and **DD** is the number of days between the two dates.
- Dates used in the calculation do not have to be in the same format (YYMMDD and CYYMDD only).

### Example 1

Example 1 is coded to calculate the employee's birth date. Notice that the two dates in the calculation are not of the same format.

### Example 2

Example 2 is coded to calculate a future date using a date field and a time span.

- The date field must be the first operand
- The Time Span field must be in YYMMDD format

### Example 3

Example 3 is coded to determine the number of days between the two events without rolling days into months. In the example a number of days is determined. This result is divided by :365 to result in a percentage. The percentage represents a percentage of the year.

### Reserved word

The **JULIAN-TS-DATE** (YYDDD) and **JULIAN-DAYS-1** (DDD) have special formats, where YY is the year (for example, 92) and DDD is the number of days into the year (for example, 166).

# Section 6 Exercise 1

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	0	0	1	1	1	1	1	1	1	1	
1	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0
CORPORATION 99 ACME MANUFACTURING										HUMAN RESOURCE LISTING										REPT PERIOD		FILE VERSION 00 PAGE 1						
DIVISION 9999 PRODUCTION CTL 1-2										X6AR PERIOD		TIME 15:59:50 DATE 11-11-2002																
EMPLOYEE NAME										CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY														
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX										XXXX	XXXX	XXXX	XXXXX	9,999,999.99														

---

## NOTES

---

## Section 6 exercise 1

### Directions

Write and run a report program from the following requirements:

### Input parameters:

**CYYMDDxxxx** where CYYMDD is the report As-of date, and xxxx is the CTRL-FOUR to be selected.

### Selection:

Include only those employees who have an HRMS Location Information segment (LZR) as of the report schedule date and a CTRL-FOUR value equal to the input CTRL-FOUR parameter.

*Note: If no input date is present, use the current date. If no CTRL-FOUR parameter is present, report on all CTRL-FOURs.*

Find the key dated segments using the input parameter date. (Hint: Use MATCH-SEGMENT-CODE.)

### Sort Key:

CTRL-FOUR, CTRL-FIVE, CTRL-SIX, and EMPLOYEE-NUMBER

### Fields:

EMPLOYEE-NAME

CTRL-FOUR, CTRL-FIVE, CTRL-SIX

JOB-CODE (If no job exists, print spaces)

ANNUAL-SALARY (If no salary exists, output zeros)

### Totals:

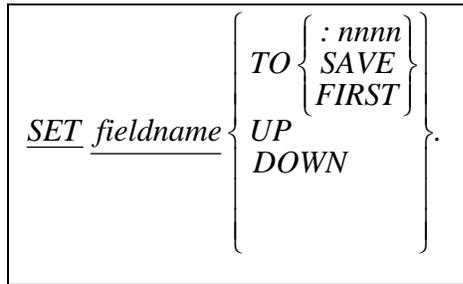
ANNUAL-SALARY total for each new CTRL-FOUR and Report Grand Total.

### Parameter entry form:

The standard parameter entry form GRPRSC may be used for this exercise. If you wish to customize it, copy it to R-xxxx using object type 'EL Source (All)' and modify it with Form Builder.

*Note: To access the LZF segment, use the macro verb FIND-SALARY-AS-OF with the AS-OF date in century format in SALARY-AS-OF-DATE.*

# Segment/Pointers Manipulation Using SET



Pointer Number	Data Description	Segment Values	Segment ID & Code field
22	Company HEDs	B001 ... B999	B-SEGMENT-TYPE
23	Company other detail	CAF ... CBx	C-SEGMENT-TYPE C-CARD-CODE
30	Employee Name and Address	F001 ... F999	F-SEGMENT-TYPE
31	Employee Labor Splits	G01 ... G99	G-SEGMENT-TYPE
32	Employee HEDs	H001 ... H999	H-SEGMENT-TYPE
34	Employee Taxes	J101 ... J4xxxxx	J-SEGMENT-TYPE
36	Employee Human Resource	LL0 ... LZZ	L-SEGMENT-TYPE L-CARD-CODE

**Example 1:**

```

SET HED-AMOUNT-CUR TO FIRST.
P200-PRINT-LOOP.
PERFORM P300-SORT.
PRINT HED-NUMBER HED-AMOUNT-MTD HED-AMOUNT-YTD.
WRITE-EXTRACT.
SET HED-AMOUNT-CUR UP.
IF H-SEGMENT-TYPE EQUALS 'H'
  AND HED-NUMBER LESS THAN '501'
  GO TO P200-PRINT-LOOP.
RETURN.
  
```

**Example 2:**

```

FIND-SALARY.
IF NOT-FOUND RETURN.
SET ANNUAL-SALARY DOWN.
IF L-SEGMENT-TYPE NOT EQUAL TO 'L'
  OR L-CARD-CODE NOT EQUAL 'ZF'
  RETURN
ELSE
  OUTPUT SALARY-EFFECTIVE
  ANNUAL-SALARY.
  
```

---

## NOTES

---

## Segment/Pointer manipulation

### Segments and pointers

As you recall, each company and employee segment is assigned a pointer address. In addition to using the FIND and PROCESS verbs, the **SET** verb may be used to alter the pointer address and position to a specific segment occurrence.

**FIRST**—positions the segment pointer to the first occurrence.

**UP**—positions the segment pointer to a key of greater value. For dated transactions, the greater value (same Segment Code) is the historical dated segment (past).

**DOWN**—positions the segment pointer to a key of lesser value. For dated transactions, the lesser value (same Segment Code) is the future dated segment.

### Example 1

Example 1 is coded to process through all HEDs with an HED number less than 501. This example provides the same result as coding using an ENDING WITH '500' qualifier.

### Example 2

Example 2 is coded to locate the salary information for the salary future to the current one. The Segment Type (ID) and Segment Code must be checked to verify that the pointer address is still addressing salary information.

## Segment/Pointers Manipulation Using SET

Save Pointer Address	Work Pointer (P-E PLACE)	Save Pointer (P-E-HOLD)
FIND-JOB-EFFECTIVE.	ZD Segment Address	Unknown
IF NOT-FOUND RETURN.		
IF JOB-CODE NOT EQUAL >10500' RETURN.		
SET JOB-CODE TO SAVE.	Unknown	ZD Segment Address
FIND-SALARY.	ZF Segment Address	ZD Segment Address
IF NOT-FOUND RETURN.		
IF ANNUAL-SALARY LESS THAN :30000.00 RETURN.		
PRINT SALARY-EFFECTIVE ANNUAL SALARY		
SET ANNUAL-SALARY TO SAVE.	ZD Segment Address	ZF Segment Address
PRINT JOB-EFFECTIVE JOB-CODE.		

---

### NOTES

---

## Segment/Pointer manipulation, continued

### Alternating pointer addresses

To obtain data from different occurrences of the same segment type, it is necessary to alternate between the two addresses of the segments. To accomplish this, use the SET verb with the SAVE qualifier.

- **SET**—changes the value of the pointer address for the data being accessed.
- **SAVE**—retains the current pointer address in a hold field (P-E-HOLD) and the hold pointer field (P-E-HOLD) is swapped with the current pointer field (P-E-PLACE). This effectively allows ‘toggling’ between two different addresses of the same pointer.

### Select and print routine

In the example, job and salary data are needed for the report when these two conditions are satisfied:

- Job Code = 10500 (LZD Segment)
- Annual Salary > or = \$30,000.00 (LZF Segment)

*Note:* The field names used in the SET syntax must be a field in the same pointer, but does not necessarily have to be a field with the same segment code.

The ‘ZD and ZF segment address’ in the example are shown for ease of understanding. A computational address value is maintained by the system.

## Additional Save/Reset Verbs

Segment	Save Verb	Restore Verb
A8	SAVE-MY-A8-PLACE	RESET-MY-A8-PLACE
AJ	SAVE-MY-FREQ-PLACE	RESET-MY-FREQ-PLACE
F	SAVE-MY-NAME-PLACE	RESET-MY-NAME-PLACE
G	SAVE-MY-LABOR-PLACE	RESET-MY-LABOR-PLACE
H	SAVE-MY-HED-PLACE	RESET-MY-HED-PLACE
J	SAVE-MY-TAX-PLACE	RESET-MY-TAX-PLACE
L	SAVE-MY-PLACE	RESET-MY-PLACE

### Example 1:

```
FIND CONTROL-3-CODE.
IF FOUND
    SAVE-MY-LABOR-PLACE
ELSE
    RETURN.
SET CONTROL-3-CODE UP.
IF G-SEGMENT-TYPE EQUALS 'G'
    PRINT 'EMPLOYEE HAS LABOR DIST SPLIT'.
RESET-MY-LABOR-PLACE.
PRINT CONTROL-3-CODE.
```

### Example 2:

```
PROCESS OTHER-NAME.
    PERFORM P200-SORT.
    PRINT OTHER-KEY OTHER-NAME OTHER-BIRTH-DATE.
    MOVE OTHER-KEY TO W6-03-100.
    SAVE-MY-PLACE.
    FIND EMPLOYER-NAME STARTING WITH W6-03-100.
    IF FOUND
        PRINT EMPLOYER-NAME OTHER-ADDRESS-1
    ELSE
        SPACE-OVER :60.
    RESET-MY-PLACE.
    WRITE-EXTRACT.
END-PROCESS.
```

---

## NOTES

---

## Segment/Pointer manipulation, continued

### Save/Reset

There are a number of instructions designed to keep track of the address of a single specific segment type.

- The SAVES are used when further manipulation of the same pointer is performed in a program.
- To Restore the pointer when required, use a companion RESET instruction.

### Example 1

Example 1 is coded to use the Save and Reset verbs for the Employee Labor Distribution (G) segment. The routine:

- Saves the place of the first occurrence of the G Segment (SAVE-MY-LABOR-PLACE).
- Sets the pointer to the next potential occurrence of the segment (SET CONTROL-3-CODE UP).
- Restores the pointer address to the first occurrence (RESET-MY-LABOR-PLACE).

### Example 2

Example 2 is coded to use the SAVE and RESET verbs for the Employee Spouse/Dependent (L01 -L02) segment. The routine:

- Processes all spouse/dependent (segment LO1) data using the PROCESS verb.
- Accesses the continuation portion of the spouse/dependent (LO2 segment) data but saves the initial pointer address (SAVE-MY-PLACE). The FIND verb locates the continuation segment (LO2 segment) that matches.
- Reestablishes access to the initial segment (next L01) using RESET-MY-PLACE.

# Additional Save/Reset Verbs—Pointer 36

## MOVE-PLACE-TO-HOLDn

## RESET-TO-HOLDn-PLACE

<p><b>Select Active Employees</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>P100-SELECTION.          FIND RESULTING-EMP-STATUS              STARTING WITH CURRENT-DATE-CYYMDD.          MATCH-SEGMENT-CODE.          IF NOT FOUND              RETURN.          IF RESULTING-EMP-STATUS NOT EQUAL TO '0'              RETURN.</p>
<p><b>Select Plan 101</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>MOVE-PLACE-TO-HOLD1.          FIND RESULTING-PLAN-STAT              STARTING WITH '101'.          IF NOT FOUND              RETURN.</p>
<p><b>Select Plan HED 521</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>MOVE-PLACE-TO-HOLD2.          FIND HED-XREF-AMT-PCT              STARTING WITH '521'          IF NOT FOUND              RETURN.</p>
<p><b>Print Status</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>MOVE-PLACE-TO-HOLD3.</p>
<p><b>Print Plan</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>P300-EXTRACT-DATA.          PERFORM P200-SORT.          RESET-TO-HOLD1-PLACE.          PRINT RESULTING-EMP-STATUS EMPLOYEE-STATUS.</p>
<p><b>Print Plan HED</b></p>	<div style="border-left: 1px solid black; border-top: 1px solid black; border-bottom: 1px solid black; width: 10px; height: 100%; margin: 0 auto;"></div>	<p>RESET-TO-HOLD2-PLACE.          PRINT RESULTING-PLAN-STAT.           RESET-TO-HOLD3-PLACE.          PRINT HED-XREF-AMT-PCT.          WRITE-EXTRACT.</p>

---

## NOTES

---

## Segment/Pointer manipulation, continued

### SAVE/RESET pointer 36

A special group of SAVE/RESET instructions have been designed exclusively for pointer 36 (L Segment) use. These are especially handy when accessing several segments in the human resource and benefits area. Common uses include selection routines or processing history for a number of different L Segments in a program.

*Note:* As is true of all the SAVE instructions, it is more efficient to use the SAVE/RESET pair then to issue secondary FIND statements.

### SAVE/RESET pairs

There are eight available pairs of instructions that save and reset a pointer address for pointer 36 (L Segment).

Save: MOVE-PLACE-TO-HOLDn, where n is 1-8.

Reset: RESET-TO-HOLDn-PLACE, where n is 1-8.

### Example

The example is coded to select only **active** employees who are enrolled in Plan 101 and have a Plan HED of 521.

- Each condition is tested, saving the pointer address of the segments that meet the selection criteria.
- Once all tests have been satisfied, the pointer addresses are reset to their appropriate 'FOUND' address and the detailed data is placed on the extract record.

# Table Records

## Basic Tables

AX	HRMS Company/Table Cross-Reference
TA	Job Code Table
TB	Salary Grade
TE	Occupation Group/EEO Job Category
TF	Activity Code Validation
TG	System Options Table
TX	EEO Location Table

## Benefits Administration Tables

AY	Benefits Company/Table Cross-Reference
TJ	Retirement Date Rules
TK	Plan Rules
TL	Plan Eligibility Rules
TM	Plan Factors
TN	Participation Rules
TO	Considered Earnings/Hours Accumulators Definition
TP	Master Plan Components
TS	Plan Prototype Contribution HEDs
TT	Plan Activity/Option
TU	Breaks-in-Service Rules
TV	Nondiscrimination Test Component Plans

---

## NOTES

---

## Accessing table data

### Table records

A table record contains information that is the same for a group of employees. For example, tables are used to define jobs, establish salary grades, or validate company rules and policies. Table data is stored on the System Control Repository (FILE01).

### Master tables

Table record access is controlled by the Company to Table Cross-Reference (TZAX and TZAY) Tables. These primary tables provide the CONTROL-NUMBER data element.

- These records define which Control Number is to be used for each organization number (Control 1–2) on your employee database during a system table read.
- A Table read is the process whereby the form or report program uses data on the employee database (or other table record) to locate a table record.

### Table rule #1

There must be a cross-reference record on file for each organization number (Control 1–2). The date of this record must precede the effective date of all associated employee transactions. For instance, before reading the TA (Job Code) Table, the AX Table JOB-CODES field is checked to obtain the control number of the TA table to access. Then, a TA Table record read is attempted.

# HRMS Table Verbs

## READ-Tx-TABLE

- Example 1:**
- ```
FIND-SALARY-AS-OF.  
IF NOT-FOUND SPACE-OVER :02 GO TO P400-WRITE.  
READ-TB-TABLE.  
PRINT TB-GEOGRAPHIC-INDEX TB-SALARY-TYPE.  
P400-WRITE.  
WRITE-EXTRACT.
```
- Example 2:**
- ```
FIND JOB-CODE.  
IF NOT-FOUND GO TO P300-NO-TABLE.  
READ-TA-TABLE.  
IF STAT-KEY GREATER '01' GO TO P300-NO-TABLE.  
PRINT JOB-CODE JOB-CODE-EXTENT JOB-TITLE.  
WRITE-EXTRACT. RETURN.  
P300-NO-TABLE.  
PRINT 'NO MATCHING JOB TABLE FOR THIS EMPLOYEE'.  
WRITE-EXTRACT. RETURN.
```

---

## NOTES

---

## Accessing table data, continued

### Table verbs

Table verbs require key information to read a table. Therefore, it is necessary to understand the following rules associated with table reads:

### Table rule #2

A table read verb is not issued without prior access to a segment or other table record that provides key data.

- In some cases, an employee level segment provides the key information.
- In other cases, an employee segment(s) and/or another table provide the key to reading the table.

### Table rule #3

Table read verbs **must not** be issued in an IF statement.

### Table rule #4

Table read must be verified to insure that a table record has been read. Check the contents of the **STAT-KEY** field.

- A value of **00** indicates a valid table read.
- A value of **01** indicates a table with a key greater than the one specified.

### Example 1

Example 1 is coded to first find the employee's current salary information segment. If the segment is found, the associated table record is retrieved and the table data is printed to the extract record. Otherwise, blanks are moved to the extract record.

### Example 2

Example 2 checks the **STAT-KEY** value to determine if a matching table record is retrieved for job related data. If so, table data is extracted. Otherwise, an error message literal is extracted.

## Section Summary

- **Run-time input parameters**
- **CSL basics plus for report writing**
- **Segment/Pointer manipulation**
- **Accessing table data**

---

### NOTES

---

## Section summary

### Run-time input parameters

- Run-time input parameters are entered on the Report Group parameter entry form when the report is scheduled. Up to 36 characters of data can be passed to a report.
  - The name of the parameter entry form is given on line sequence # \_\_\_\_\_ with the literal \_\_\_\_\_.
- \_\_\_\_\_ defines the number of characters defined as parameters for the report. The parameters reside in Pointer \_\_\_\_\_ starting at displacement \_\_\_\_\_ during the report execution.

### CSL basics plus for report writing

- Date fields may be converted from century to regular format or regular to century format using the \_\_\_\_\_ command.

### Segment/Pointer manipulation

- In addition to the FIND and PROCESS verbs, the \_\_\_\_\_ verb can be used to manipulate a pointer address to either the FIRST occurrence, or \_\_\_\_\_ and \_\_\_\_\_ to a specific occurrence.

### Accessing table data

- A Table Read verb should not be issued without prior access to \_\_\_\_\_ that provides key data.
- Table read verbs \_\_\_\_\_ be issued in an IF statement.

# Section 6 Exercise 2

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	1	1	1	1	1	1	1														
1..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5	..0	..5														
CORPORATION 99 ACME MANUFACTURING																		HUMAN RESOURCE LISTING						REPT PERIOD		FILE VERSION 00		PAGE 1										
DIVISION 9999 PRODUCTION CTL 1-2																		CLC2 PERIOD						X6BR PERIOD		TIME 15:59:50		DATE 11-11-2002										
EMPLOYEE NAME																		CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY																
XXXXXXXXXXXXXXXXXXXXXXXXXXXX																		XXXX	XXXX	XXXX	XXXXX	9,999,999.99																
:																		:	:	:	:																	
:																		:	:	:	:																	
*JOB-CODE 15405																		9,99,999,999		9,999,999.99		JOB TITLE....																

## NOTES

## Section 6 exercise 2

### Directions

Modify the report created in section 6, exercise 1 using the following specifications:

### Input parameters:

No change from exercise 1.

### Selection:

Include only active employees (RESULTING-EMP-STATUS ='0') who have an HRMS Location Information segment as of the Report Group date and a CTRL-FOUR value equal to the Report Group CTRL-FOUR parameter.

*Note: If no date is present use the current date. If no CTRL-FOUR parameter is present, report all CTRL-FOURs. Additionally, if there is no job or salary segment for the employee, exclude them from the report.*

### Fields:

EMPLOYEE-NAME

CTRL-FOUR, CTRL-FIVE, CTRL-SIX

JOB-CODE, JOB-TITLE (TA Table, print at total time only)

ANNUAL-SALARY

Employee Count (PERM counter)

Hint: (Use MOVE-PLACE-TO-HOLDn/RESET for all L Segment data)

### Sort Key:

CTRL-FOUR, JOB-CODE and EMPLOYEE-NUMBER

### Totals:

ANNUAL-SALARY Total and Employee Count Total for each new JOB-CODE and a Report Grand Total.

*Note: The results shown in Appendix A for this exercise were obtained using the current date and CTRL-FOUR of 4040.*

---

**NOTES**

---

## Section 7: Reports-Special Functions

---

### Table of Contents

Introduction.....	201
Labor and history records .....	203
Variable report headings .....	209
Summary reporting .....	211
Report special formatting.....	213
Section summary.....	215
Section 7 exercise 1 .....	219
Section 7 exercise 2 .....	221

## Objectives

- **Access labor and history records**
- **Create variable report headings**
- **Create a summary report**
- **Use formatting instructions and subroutines**

---

### NOTES

---

## Introduction

### **Purpose**

The purpose of this section is to cover miscellaneous functions of reporting. This includes accessing labor and history records as well as special reporting techniques.

### **Objectives**

Upon completion of this section you will be able to:

- Access labor and history records
- Create variable report headings
- Create a summary report
- Use formatting instructions and subroutines

## Employee Labor & History Records

Segment	Permanent Master	Labor Record	History Record
Employee Record Key	Organization Number (Control 1-2), 'M', Employee Nbr, '99'	Organization Number (Control 1-2), 'M', Employee Nbr, Unique Master Nbr	Organization Number (Control 1-2), 'M', Employee Nbr, Unique Master Nbr
E – Basic Employee Data	Occurrences=3 (E, EA, EB)	Occurrences=2 (E, EA)	Occurrences=2 (E, EA)
F – Name & Address	Occurrences=All	Occurrence=Name Code 001	Occurrence=Name Code 001
G – Labor Distribution	Occurrences=All	Occurrences=1	Occurrence=Location Nbr 01
H – Employee HEDs	Pointer=32 Occurrences=All Accumulators=Curr, Mtd, Qtd, Ytd	Pointer=33 Occurrences=Single Accumulators=Current	Pointer=33 Occurrences=Multiple Accumulators=Current
J – Employee Taxes	Pointer=34 Occurrences=All	Pointer=N/A Occurrences=N/A	Pointer=35 Occurrences=Multiple
L – HRMS/User defined data	HRMS data/User data	Additional Labor Data/User data	Occurrences=N/A
P – Period End Table	Occurrences=All	Occurrences=N/A	Occurrences=N/A
Record Group Values	99	+L, -L, ML, 9L	+H, -H, MH, 9H

---

### NOTES

---

## Labor and history records

### History and labor records

Payroll runs generate Labor and history records. History records are snapshots of the check(s) written to the employee. Labor records detail all timecards and adjustments processed for the employee.

The facing page is a comparison of the segments present on labor and history records vs. the permanent master record.

☞ *Refer to Appendix C: Segment Layout Reports for each segment's layout.*

### Accessing history and labor

As you recall, the source records field on the Report Group form (RGMSTR) determines which employee records are passed to your report program. The source records values include:

- Employee Master only
- History only
- Labor only
- Labor/Hist/Emp Mastr Position Administration only

### Record group

The type of Labor/History record may vary based on the type of payment produced. The source of the Labor/History record can be determined by checking the RECORD–GROUP field.

- + = Plus Adjustment
- = Minus Adjustment
- M = Manual Check
- 0–9 Systems produced payment

## Labor/History Records Stored Chronologically

<u>Start</u>	<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>
PM	L1	L1	L1	L1	L1
	H1	H1	H1	H1	H1
PM	L2	L2	L2	L2	L2
	L3	L3	L3	L3	L3
	H2	H2	H2	H2	H2
	PM	PM	L4	L4	
			L5	L5	
			L6	L6	
			L7	L7	
			H3	H3	
			PM	L8	
				H4	
				L9	
				H5	
				PM	

---

### NOTES

---

---

## Labor and history records, continued

### Pay cycles

Each time an employee is processed through a pay cycle labor and history records are created if the employee is to be paid. The following scenario shows the order the labor, history, and permanent employee records are stored:

#### Cycle 1

Input: One Time entry, One (G) Labor split

Creates: **L1** Labor record, **H1** History record

#### Cycle 2

Input: Two Time entries, One (G) Labor split

Creates: **L2** Labor record - 1st Time entry,  
**L3** Labor record - 2nd Time entry, **H2** History record

#### Cycle 3

Input: Maintenance Run (non-pay) no **L/H** created

#### Cycle 4

Input: One Time entry/2 earnings - Regular and OT, Two (G) Labor splits

Creates: **L4** Labor record for Regular/1st split,  
**L5** Labor record for Regular/2nd split,  
**L6** Labor record for OT;1st split,  
**L7** Labor record for OT;2nd split, **H3** History record

#### Cycle 5

Input: Two Time entries/Separate Checks, One (G) Labor split

Creates: **L8** Labor record/1st Time entry, **H4** History record/1st Check, **L9**  
Labor record/2nd Time entry, **H5** History record/2nd Check

# Labor/History Record Verification

**PERMANENT-MASTER  
Verification**

IF [NOT] PERMANENT-MASTER

**LABOR-RECORD  
Verification**

IF [NOT] LABOR-RECORD

**HISTORY-RECORD  
Verification**

IF [NOT] HISTORY-RECORD

Example:

```
IF NOT LABOR-RECORD
  RETURN.
PERFORM P200-SORT.
OUTPUT PAY-FREQUENCY-CODE PAYMENT-CODE SHIFT-CODE.
WRITE-EXTRACT.
```

Result:

Only the Labor record is processed in the report.

---

## NOTES

---

## Labor and history records, continued

### Labor and history access

Accessing labor and/or history is dependent on:

- Having labor and/or history records available on the Employee Database (FILE02).
- Using the Report Group (RGMSTR) Source Records values.

### Labor and history verbs

Here are the instructions used to verify which record has been passed to the History, Labor, or Labor/Hist/Emp Mastr report program:

#### IF PERMANENT–MASTER

- Report Group (RGMSTR) source record must be an Employee Master or Labor/Hist/Emp Mastr.
- This instruction tests the value of the employee record key master number for a value of '99'.

#### IF LABOR–RECORD

- Report Group (RGMSTR) source record must be Labor or Labor/Hist/Emp Mastr.
- This instruction tests the value of the RECORD–GROUP field (position 2) for a value of 'L'.

#### IF HISTORY–RECORD

- Report Group (RGMSTR) source record must be History or Labor/Hist/Emp Mastr.
- This instruction tests the value of the RECORD–GROUP field (position 2) for a value of 'H'.

# Variable Report Headings

CORPORATION 99 ACME MANUFACTURING	SALARY ANALYSIS	REPT	PAGE 1
DIVISION 9999 PRODUCTION CTL 1-2	AS OF 10-28-94	X9-R	TIME 16:01:35 DATE 08-07-1992

$\left\{ \begin{array}{l} PRINT \\ OUTPUT \end{array} \right\} \underline{Sort - data} \underline{' .hPPP'} \underline{heading - data} \underline{' ..'}$

**Where:** *h* = 1 for Heading Line 1  
2 for Heading Line 2  
3 for Heading Line 3  
4 for Heading Line 4  
*PPP* = report print positions 001-132

**Example:**  
IF FIRST-TIME-IND EQUAL TO 'F'  
PRINT '1X9-R' CONTROL-1 CONTROL-2 SPACE-OVER :26  
PRINT '.2065' CURRENT-DATE '..' WRITE-EXTRACT.

---

## NOTES

---

## Variable report headings

### Heading overrides

It may be useful to place varied data in the report headings each time the report is run. To accomplish this, special extract records can be written by the report program which supply overrides to the report headings.

### Period record

This record is termed a 'Period' record because it contains a period (.) in place of the Record Type value.

- The print program recognizes this record in the extract file and uses it to supply overrides to the heading(s).
- The ending period literal is required in the syntax.
- An unlimited number of period records can be created to provide information to the four headings.
- Period records may only override user-defined titles in header 1 and 2.
- The Period record(s) are produced only once per organization (Control 1–2). Use the FIRST–TIME–IND (value of 'F') test when writing them.
- The Period record(s) must be the first/lowest in the sorted extract file. To insure this, format the record normally with the Forms/Report Code literal first, followed by a organization number (Control 1–2) value. Then, space out the remaining positions of the sort key.
- No sort key options are used when formatting this extract record.

# Summary Reporting

CORPORATION 99	ACME MANUFACTURING	JOB SUMMARY REPORT			REPT	PAGE 1
DIVISION 9999	MIDWEST DIVISION				JSUM	TIME 10:50:40 DATE 11-02-1993
	JOB CODE	JOB-CODE EXTENT	JOB-TITLE	SALARY GRADE	OCCUPATION GROUP	ANNUAL SALARY
*JOB-CODE	10650	0001	ASST STATIONARY ENGINEER	S31	696	1,000,550.00
*JOB-CODE	15405	0001	SHIPPING/RECEIVING CLERK	S10	888	240,009.33
*JOB-CODE	17857	0001	ASSEMBLY LINE WORKER	S16	699	650,005.50
*JOB-CODE	18020	0001	LATHE MACHINE TOOL SETTER	S05	703	887,950.75
*CONTROL-1-2	999999					2,778,515.50
*REPORT CODE	XSUM					2,778,515.50

**Example: DEFINE-REPORT TOTALS-ONLY.**

```

:
:
:

```

```

OUTPUT '1JSUM' FORMS/REPORT-CODE
PRINT-GRAND-TOTAL DOUBLE-SPACE-BEFORE
CONTROL-1-2 DOUBLE-SPACE-BEFORE.
FIND-JOB-EFFECTIVE.
IF NOT FOUND
RETURN
ELSE
OUTPUT JOB-CODE BREAK-DEFAULT
SORT-LENGTH-21 JOB-CODE-EXTENT.
READ-TA-TABLE.
OUTPUT '0' JOB-CODE-EXTENT JOB-TITLE
SALARY-GRADE OCCUPATION-GROUP.
FIND-SALARY.
IF NOT FOUND
OUTPUT '000000000'
ELSE
OUTPUT ANNUAL-SALARY.
WRITE-EXTRACT.

```

SEQ	FIELD	-----PRINT-----		
NBR	NAME	POS.	LINE	TOTAL
00	JOB-CODE-EXTENT	029	01	P
05	JOB-TITLE	041	01	P
10	SALARY-GRADE	074	01	P
15	OCCUPATION-GROUP	080	01	P
20	ANNUAL-SALARY	092	01	T

## NOTES

## Summary reporting

### Summary report

A summary report is one that does not print detail for any given employee/company but produces totals based on a control break(s).

### TOTALS–ONLY

The TOTALS–ONLY option may be used to specify that a report only displays totals.

- This option must be coded in the DEFINE–REPORT verb in your extract program.
- Totals are determined by the sort criteria and report print position parameters.

### Example

The example is coded to display a summary of jobs and the total of the annual salary for each job.

- The combination of the TOTALS–ONLY verb and report print position parameters display only the data specified with a P, Y, or T Print Total options.
- Notice that the JOB–CODE field is not in the extract or Report Print Position (RTEDIT) form, however it displays in the report. This is due to the function the control break code automatically performs.

# 'First Middle Last' Name Conversion Subroutine

## FMLEDT

**Example:** MOVE EMPLOYEE-NAME TO LAST-FIRST.  
CALL 'FMLEDT'.  
MOVE PRINT-FIELD TO FIRST-LAST.  
PRINT FIRST-LAST.

**Result:** ROSS, JOY L. becomes JOY L. ROSS

# Access to Process Date and Time

## UPDATE-DATE-TIME

**Example:** IF EMPLOYEE-NUMBER EQUALS '999999999'  
UPDATE-DATE-TIME PRINT CURRENT-DATE CURRENT-TIME.

The execution date and time are refreshed and extracted.

**Result:**

---

## NOTES

---

## Report special formatting

### **FMLEDT**

The FMLEDT subroutine converts a name field, such as EMPLOYEE-NAME, from 'Last, First Middle' to 'First Middle Last' order.

### **Work fields used**

Be sure to follow exactly the example given. FMLEDT is a CALLED subroutine expecting EMPLOYEE-NAME to be moved to **LAST-FIRST**. The manipulated result is in PRINT-FIELD.

### **Update date and time**

The UPDATE-DATE-TIME verb is used to update the system maintained fields CURRENT-DATE and CURRENT-TIME.

Without issuing this verb, these date and time fields reflect the time and date the program **started** executing. Use this verb when it is significant to capture the **point-in-time** execution date and time.

## Section Summary

- **Labor and History Records**
- **Variable Report Headings**

---

### NOTES

---

## Section summary

### Labor and history records

- The \_\_\_\_\_ field on the Report Group form (RGMSTR) is used to pass Permanent Master, \_\_\_\_\_ and/or History records to your report.
- When a Source Record type of Labor/Hist/Emp is specified on a Report Group, you may use the \_\_\_\_\_, \_\_\_\_\_, or \_\_\_\_\_ statements to see which employee record has been passed to the report.

### Variable report headings

Variable heading information can be supplied to a report using a \_\_\_\_\_. The period extract record must only be created \_\_\_\_\_ time per organization number (Control 1–2).

The syntax for the header override is '*h*ppp' where:

*.* = \_\_\_\_\_  
*h* = \_\_\_\_\_  
*ppp* = \_\_\_\_\_

## **Section Summary, continued**

- **Summary Reporting**
- **Report Special Formatting**

---

### **NOTES**

---

## Section summary, continued

### Summary reporting

- To create a summary report, you must specify the DEFINE-REPORT option of \_\_\_\_\_, and define each summary field using a \_\_\_\_\_ in the Report Print Position Edit (RTEDIT) Total Column.

### Report special formatting

- A subroutine that reformats a name field into 'First Middle Last' order is \_\_\_\_\_. Before executing the subroutine, the name field must be MOVED to the \_\_\_\_\_ field.

# Section 7 Exercise 1

										1	1	1	1	1	1	1			
1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	0	0
1	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5	0	5
CORPORATION		99 ACME MANUFACTURING				HUMAN RESOURCE LISTING				REPT	PERIOD	FILE VERSION 00 PAGE				1			
DIVISION		9999 PRODUCTION CTL 1-2				CTRL 4: xxxx As Of: mm-dd-yyyy				X71R	PERIOD	TIME 15:59:50 DATE 11-11-2002							
EMPLOYEE NAME		CTRL	CTRL	CTRL	JOB	ANNUAL													
		FOUR	FIVE	SIX	CODE	SALARY													
ROBERT NORRIS		4040	5050	6060	15405	11,086.40													
:				:															
:				:															
:				:															
*JOB-CODE		70101		9,999,999,999		211,456.99		JOB TITLE...											

---

## NOTES

---

## Section 7 exercise 1

### Part 1 directions

Modify the report created in section 6, exercise 2 to include the following:

#### Fields:

Same fields as section 6 exercise 2 except the EMPLOYEE-NAME field should be displayed in 'First Middle Last' name order.

#### Headings:

Override the 2nd Title Heading to include the run-time parameters.

SubTitle: 'Control 4: xxxx As of: mm-dd-yyyy'

Where: xxxx is the Control 4 value from the run-time parameters.

*Note:* The results for this exercise shown in Appendix A reflects the use of a CTRL-FOUR of 4040.

## Section 7 Exercise 2

	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	1	1	1	1	1	1	1		
	1	5	0	0	5	0	0	5	0	0	5	0	0	5	0	0	5	0	0	5	0	0	5	0	0	5	0
CORPORATION	99	ACME MANUFACTURING	HUMAN RESOURCE LISTING										REPT	PERIOD	FILE VERSION	00	PAGE	1									
DIVISION	9999	PRODUCTION CTL 1-2	CTRL 4: 4444 As Of:01-01-2002										X72R	PERIOD	TIME	15:59:50	DATE	11-11-2002									
EMPLOYEE NAME		CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY																					
*JOB-CODE	70101			9,999,999,999		211,456.99	JOB TITLE...																				
*JOB-CODE	70102			9,999,999,999		115,336.99	JOB TITLE...																				
*JOB-CODE	70103			9,999,999,999		141,456.92	JOB TITLE...																				
*JOB-CODE	70104			9,999,999,999		151,435.44	JOB TITLE...																				

### NOTES

## Section 7 exercise 2

### Part 2 directions

Create a summary report from the report coded in section 7, exercise 1.

- Copy the report created in section 7, exercise 1 to a new name. Use an Object of **EL Rpt Components** (RPT) to copy all report components.
- Modify the report program so only the total information displays on the report. There are no additional changes to the report selection, headings, sort or detail.
- As you review the resulting report you might note some additional changes to the headers that may be desirable.

*Note:* The results for this exercise, shown in Appendix A, reflect the use of a CTRL-FOUR of 4040.

---

**NOTES**

---

## Section 8: Introduction to Special Print Options

---

### Table of Contents

Introduction.....	225
Special print option overview .....	227
New organization number (Control 1–2) is starting .....	237
Heading record are ready to print .....	239
Additional heading lines .....	241
Detail data with matching sort key is ready.....	243
Total data is ready .....	245
A detail line is ready to print .....	247
A total line is ready to print .....	249
The report is done .....	251
Section summary.....	253
Section 8 exercise .....	255

## Objectives

- **Identify the special print options program**
- **Define the work areas and fields used in the special print options program**
- **Use the special print options model to enhance report output**

---

### NOTES

---

## Introduction

### **Purpose**

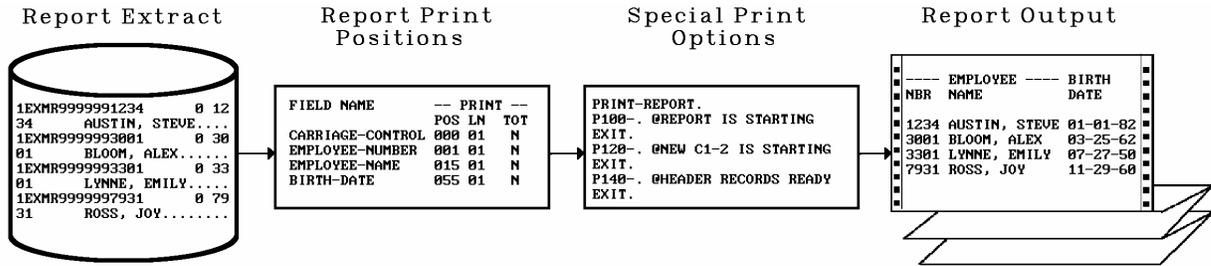
This section provides details about the special print option program. Examples are given to illustrate the entry points where standard print logic may be supplemented.

### **Objectives**

Upon completion of this section you will be able to:

- Identify the special print options program
- Define the work areas and fields used in the special print options program
- Use the special print options model to enhance report output

# Special Print Option Processing



## NOTES

## Special print option overview

### Special print option

The Special Print Options program is an optional CSL program that enhances the report output during the print phase of the report process.

The report output can be enhanced by the special print options by:

- Changing heading, detail and control break lines
- Suppressing lines not needed
- Adding lines
- Changing line advancement
- Performing additional calculations on the control break level
- Creating an extract file

# Special Print Option Program

**Report Extract Program Name = XXXXPT**

**Special Print Option Program Name = XXXXP**

---

## NOTES

---

## Special print option overview, continued

### Special print option program

The Special Print Options program contains several paragraphs that are performed automatically by the Report Print program (RTPRNT).

### Special print option name

Since the Special Print Option program is automatically executed by RTPRNT, its name must be associated with the Report Extract program's name.

- The first 5 characters of the Report Extract program is used as the Special Print Option program Name.

For example, if the Report Extract Program name is **XXXXPT**. The Special Print Options program must be named **XXXXP**. This name association is what causes automatic execution by the RTPRNT program.

### Special print option development

When developing a report that includes special print options, it is recommended that you first complete the coding and testing of the Report Extract program and Report Print Positions Edit (RTEDIT) before developing the Special Print Options program. This streamlines the task since the decisions made in the report extract and report print positions directly impact the Special Print Options program.

## Special Print Option Paragraphs

Paragraph	RTPRNT Process Timing
100	The report is starting
120	New Organization Number (Control 1-2) is starting
140	The header records are ready to print
160	The header records have been printed
180	Detail data with matching sort keys is ready
200	Total data is ready
220	A detail line is ready to print
240	A total line is ready to print
260	The report is done

---

### NOTES

---

## Special print option overview, continued

### Special print option paragraphs

The Special Print Option paragraphs are performed at specific times during the report print phase. Each paragraph is designed to allow you access to specific parts of the report output before and after the print line has been formatted.

General rules associated with the Special Print Options program include:

- All paragraph labels **MUST** be present, even if your Special Print Option program does not use them.
- Each paragraph must contain an EXIT that directs control back to the Report Print (RTPRNT) program.
- Additional paragraphs can be added to the program in the range of 100–999 (with the exception of the ones already used). Each new paragraph must contain an EXIT, so control can be properly returned to the performing paragraph.

### 5M–RP program

The 5M–RP program is a delivered model that you should use as a prototype whenever you need to create a Special Print Options program. The 5M–RP program contains all the required paragraphs, as well as comments providing direction for each paragraphs use.

## Work Areas and Fields

### Work Areas

- Pointer 8 – Sort Key, Report Detail, Report Totals
- Pointer 11 (SCREEN) – Report Headings
- Pointer 19 – Report Extract Record

### Line count and advancing

- Line Number (W8-02-158)
- LINE-COUNT
- LINE-ADVANCE

---

## NOTES

---

## Special print option overview, continued

### Work areas

The Special Print Options program makes use of working storage areas during processing.

- Paragraphs P140–P., P180–P., and P200–P. indicate that work area W8–40–160 must not be used. It is already being used to retain sort control information. So, do not overlay this area!
- Paragraph P200–P. documents that W8–15–164 contains the sort key field name. Your program for override control break processing can check this variable.
- The SCREEN Area is used for headings.
- F14–REC01–90 is the contents of the first 90 characters of the FILE14 input extract record. F14–REC91–150 is the last 60 characters of the FILE14 input extract record.

### Line counts and numbering

LINE–ADVANCE is used to define the number of blank lines to advance before printing (carriage control). If LINE–ADVANCE is adjusted, LINE–COUNT is automatically adjusted.

99 = Suppress the line

00 = Advance to Top of Page (no headings)

LINE–COUNT is used to keep count of output lines. Increment this when extra lines are added. This count must be kept accurate to properly handle page breaks.

Work field W8–02–158 contains the RTEDIT Line Number being processed. This can be tested in P220–P. and P240–P.

# P100-P. The Report is Starting

PRINT-REPORT

INITIAL-PRINT-LINE

WRITE

PRINT - REPORT.

INITIAL - PRINT - LINE.

WRITE filename.

**Example:**

```
PRINT-REPORT.  
P100-P. @ REPORT IS STARTING  
MOVE :0 TO PERM-01-V0.  
INITIAL-PRINT-LINE.  
MOVE '*** REPORT XXXRPT IS STARTING ***' TO  
W8-33-000.  
WRITE FILE03.  
INITIAL-PRINT-LINE.  
MOVE '00' TO LINE-ADVANCE.  
WRITE FILE03.  
EXIT.
```

**Result:**

A banner page is printed.

---

## NOTES

---

## Special print option overview, continued

### Print report

The PRINT-REPORT is a verb used in the Special Print Option program to read records into a table so the paragraphs can be performed. This verb must be the first command in the Special Print Options program.

### Initialize print line

The INITIAL-PRINT-LINE moves spaces to the first 132 positions of pointer 8. Report programs use this area of Pointer 8 to format a print line.

### Write

The WRITE verb is used to output a record for the specified file. Data is written from pointer 8 starting at displacement 000.

### P100-P.

Paragraph 100 is performed when the report is starting. This paragraph is typically used to:

- Initialize report wide work areas
- Print report banner pages
- Header labels for extract files (see Appendix B)

### Example

The example is coded to:

- Define the Special Print Option Report using the PRINT-REPORT verb.
- Initialize any report wide work areas.
- Initialize the print area (Pointer 8) using the INITIAL-PRINT-LINE verb.
- Move the banner message literal to the print area.
- Write the banner message to FILE03.
- Re-initialize the print area.
- Move '00' to LINE-ADVANCE to force the report to a new page before proceeding and write FILE03.

# P120-P. New Organization Number (Control 1-2) is Starting

**Example:**

```
P120-P. @ NEW C1-2 IS STARTING  
MOVE 'Y' TO W6-01-100.  
EXIT.
```

**Result:**

Work area is initialized.

---

## NOTES

---

## **New organization number (Control 1–2) is starting**

### **P120–P.**

Paragraph P120–P. is performed when a new organization number (Control 1–2) is starting. This paragraph is typically used to initialize organization number (Control 1–2) work areas.

### **Example**

The example is coded to initialize a switch in pointer 6.



## Heading record are ready to print

### **P140–P.**

The report heading information is stored in pointer 11 (SCREEN) when paragraph 140 is performed. Therefore, any changes to the four (4) report headings can be made by moving data to specific addresses of the SCREEN Work area.

- Header 1 starts in :1241 of SCREEN for 132 positions
- Header 2 starts in :1373 of SCREEN for 132 positions
- Header 3 starts in :1505 of SCREEN for 132 positions
- Header 4 starts in :1637 of SCREEN for 132 positions

### **Example**

To override a heading literal, a SET instruction is required. The example adds a sub–title and appends a literal of ‘PT’ to the report name in Header 2.

- SETting SCREEN to a specific address allows you to override the values in the heading.
- A PRINT statement moves literal data to the SCREEN (header) area.
- The SCREEN pointer is automatically advanced for the length of the PRINTed literal.

# P160-P.

## Additional Heading Lines

**Example:**

```

Adding additional headings.
P160-P. @ HEADER RECORDS HAVE BEEN PRINTED
@ DO NOT ALTER W8-40-160.
INITIAL-PRINT-LINE.
MOVE '===== ' TO W8-30-000.
MOVE W8-30-000 TO W8-30-029.
@MOVE W8-30-000 TO THE REST OF THE PRINT LINE POSITIONS.
WRITE FILE03.
INITIAL-PRINT-LINE.
MOVE F14-REC01-90 TO W8-90-800.
MOVE 'CTRL THREE: ' TO W8-12-000.
MOVE W8-04-811 TO W8-04-012. WRITE FILE03.
CALCULATE LINE-COUNT + :02 = LINE-COUNT.
EXIT.
    
```

**Result:**

A fifth and sixth heading is printed at the top of each page.

1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	0	0	1	1	1	1	1	1	1	
1..5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0...
CORPORATION	99	ACME	MANUFACTURING					EMPLOYEE	HEADCOUNT			REPT								PAGE		1					
DIVISION	9999	PRODUCTION	CTL 1-2									1N-R							TIME	11:53:56	DATE	10-13-2002					
EMPLOYEE NAME		EMPLOYEE	CTRL	CTRL	CTRL	CTRL		MAIL	HIRE	EMPLOYEE																	
		NUMBER	THREE	FOUR	FIVE	SIX		DISTRIBUTION	DATE	STATUS																	
=====																											
CTRL THREE: XXXX																											

## NOTES

## Additional heading lines

### **P160–P.**

Paragraph P160–P. allows you to create additional headings. This paragraph is executed after the standard headings 1–4 have been written to FILE03.

### **Example**

To add additional headings to a report, you will need to move literal or work area data to the print area (Pointer 8) and write the data to the print file.

- The **INITIAL–PRINT–LINE** macro verb moves 132 spaces to the first 132 positions of Pointer 8. This verb is used to initialize the print area before building the additional headings.
- When adding additional lines to the report, they must be added to the **LINE–COUNT** increment. As a general rule, add 1 to **LINE–COUNT** for each additional line that is written.

# P180-P.

## Detail Data With Matching Sort Keys is Ready

-----SORT-----  -----REPORT-----
1ANYR99999912345678900FIELD1...FIELD2
1ANYR99999912345678901FIELD3...FIELD4
1ANYR99999912345678902FIELD5...FIELD6
Record Type

**Example:** P180-P. @ DETAIL DATA WITH MATCHING SORT KEYS IS READY  
@ USE RECORD-TYPE:FIELD-NAME TO ACCESS DATA  
@ FROM THE EXTRACT RECORDS.  
@ W8-40-160 MAY BE ALTERED.  
IF 0:FIELD1 GREATER THAN 1:FIELD4  
CALCULATE 0:FIELD2 + 1:FIELD4 = 1:FIELD4.  
EXIT.

---

## NOTES

---

## Detail data with matching sort key is ready

### **P180–P.**

Paragraph P180–P. allows you to control the detail that is to be written to the report. This is especially helpful when more than one extract record type is written for a sort key group, since the records are grouped together at this time.

The system determines the location of the record type in the extract record based on the compiled extract program module. Where the sort key ends, the record type begins. All records in the process are read in from FILE14 until the sort key changes.

### **Example**

As the example indicates, Record Types 0–2 are available at the detail level for a matching sort key. Manipulations here can affect the printed detail line and the total data.

# P200-P.

## Total Data is Ready

**Example:**

```
P200-P. @ TOTAL DATA IS READY
@ USE RECORD-TYPE:FIELD-NAME TO ACCESS DATA
@ IN THE TOTAL COUNTERS.
@ DO NOT ALTER W8-40-160
@ W8-15-164 CONTAINS NAME OF LEVEL BREAK.
IF W8-03-164 EQUALS 'REP'
    CALCULATE 1:PERM-01-V2 / 1:PERM-02-V0 = 1:PERM-03-V2.
EXIT.
```

---

## NOTES

---

## Total data is ready

### P200–P.

Paragraph P200–P. allows you to control the totals that are written to the report.

- Paragraph P200–P. documents that W8–15–164 contains the sort key field name. Your program can check this variable by for override control break processing.

### Example

In this example the control break processed (name of field in W8–15–164) is checked for the grand total. At the grand total break (REPORT CODE) an average is calculated referring to the fields as RTEDIT extract file reference, that is, Record Type:Fieldname.

### Print position record preassignment

In the above example, the calculated result field, 1:PERM–03–V2 must be predefined in the print position record layout. There is no need for a corresponding field in the extract routine. The RP program creates it.

# P220-P. A Detail Line is Ready to Print

CORPORATION	99	ACME MANUFACTURING	SPOUSE/DEPENDENT CHANGES	REPT	FILE VERSION 00	PAGE	1			
DIVISION	9999	PRODUCTION CTL 1-2	BY PLAN	XK-R	TIME 09:14:46	DATE	10-23-2002			
PLAN	EMPLOYEE	DEPENDENT	DEP		BIRTH	F/T	INSURANCE			
ID	EMPLOYEE NAME	NUMBER	DATE	XREF	DEPENDENT NAME	SEX	DATE	STDT	RELATION	CARRIER
101	BLOOM, ALEXANDER	3001	01-01-1983	001	BLOOM, BRENDA	F	12-24-1940	N	WIFE	CONN.GENER
101	BLOOM, ALEXANDER	3001	01-01-1983	002	BLOOM, ABAGAIL	F	01-17-1915	N	MOTHER	
101	BLOOM, ALEXANDER	3001	01-01-1983	003	BLOOM, ELLEN	F	03-24-1972		DAUGHTER	

**Example:**

P160-P. @HEADER RECORDS HAVE BEEN PRINTED  
 MOVE SPACES TO W6-51-100WORK.  
 EXIT.

P220-P. @A DETAIL LINE IS READY TO PRINT  
 IF W8-51-000PRINT EQUALS W6-51-100WORK  
 MOVE SPACES TO W8-51-000PRINT  
 ELSE MOVE W8-51-000PRINT TO W6-51-100WORK.  
 EXIT.

**Result:**

Duplicate detail data in lines is suppressed.

CORPORATION	99	ACME MANUFACTURING	SPOUSE/DEPENDENT CHANGES	REPT	FILE VERSION 00	PAGE	1			
DIVISION	9999	PRODUCTION CTL 1-2	BY PLAN	XK-R	TIME 09:14:46	DATE	10-23-2002			
PLAN	EMPLOYEE	DEPENDENT	DEP		BIRTH	F/T	INSURANCE			
ID	EMPLOYEE NAME	NUMBER	DATE	XREF	DEPENDENT NAME	SEX	DATE	STDT	RELATION	CARRIER
101	BLOOM, ALEXANDER	3001	01-01-1983	001	BLOOM, BRENDA	F	12-24-1940	N	WIFE	CONN.GENER
			01-01-1983	002	BLOOM, ABAGAIL	F	01-17-1915	N	MOTHER	
			01-01-1983	003	BLOOM, ELLEN	F	03-24-1972		DAUGHTER	

## NOTES

## A detail line is ready to print

### P220–P.

Paragraph P220–P. is performed when the detail line is in the print area (Pointer 8) just before it is written to the output file.

- Work field W8–02–158 contains the RTEDIT Line Number being processed. The paragraph's timing is for each different Print Position Record line number.

### Example:

When detail lines are formatted and printed, there is no automatic group indication. You may wish to eliminate this duplication by using a few entry points in the Special Print Options program. The example is coded to check each detail line as it is processed to determine if the same data is to be printed.

- P160–P.

In the HEADER RECORDS HAVE BEEN PRINTED paragraph, a work area is cleared out for initialization.

- Initializing here insures that the first detail line of any new page will contain all detail data.

- P220–P.

In the DETAIL LINE IS READY TO PRINT paragraph, access to the formatted print line is available.

- The print line is in the first 132 positions of Pointer 8, **relative to zero**.
- The print area (W8) is compared to the work area (W6) and the print area is spaced if equal. Otherwise, the print data is moved to the work area to be compared again.

# P240-P. A Total Line is Ready to Print

## Standard Control Breaks:

*CONTROL-4-CODE	1221	60
*CONTROL-4-CODE	4444	7
*CONTROL-4-CODE	5123	3
*CONTROL-4-CODE	7931	15
*CONTROL-4-CODE	8490	2
*REPORT CODE	CB-R	95

## Override Control Breaks:

COST CENTER	1221	60
COST CENTER	4444	7
COST CENTER	5123	3
COST CENTER	7931	15
COST CENTER	8490	2
GRAND TOTAL COUNT		95

**Example:** P240-P. @A TOTAL LINE IS READY TO PRINT  
IF W8-15-000PRINT EQUALS '\*CONTROL-4-CODE'  
MOVE 'COST CENTER ' TO W8-15-000  
ELSE  
IF W8-07-000PRINT EQUALS '\*REPORT'  
MOVE 'GRAND TOTAL COUNT ' TO W8-21-000  
EXIT.

---

## NOTES

---

## A total line is ready to print

### P240-P.

Paragraph P240-P. is performed when the total line is in the print area (pointer 8) just before it is written to the output file. The TOTAL LINE IS READY TO PRINT paragraph is used to access the total print line.

- The print literal can be overridden with a MOVE to the print (W8) area.

### Example:

The example is coded to:

- Override the \*CONTROL-4-CODE with the more descriptive 'COST CENTER'
- Override the \*REPORT-CODE with 'GRAND TOTAL COUNT'

Warning: Make sure that your literal is the same length as your move length! If it is shorter, garbage is moved to the print line.

# P260-P. The Report is Done

**Example:**

```
P260-P. @ THE REPORT IS DONE  
INITIAL-PRINT-LINE.  
MOVE '*** REPORT XXXRPT IS COMPLETE ***' TO  
W8-33-000.  
MOVE '00' TO LINE-ADVANCE.  
WRITE FILE03.  
EXIT.
```

---

## NOTES

---

## The report is done

### **P260–P.**

Paragraph P260–P. is performed when the report is complete. This paragraph is typically used to:

- Print report trailer pages.
- Create trailer records for extract files.

☞ *Refer to Appendix B: Extract Record Files for additional information.*

### **Example**

The example is coded to:

- Initialize the print area (pointer 8) using the INITIAL–PRINT–LINE verb.
- Move the trailer message literal to the print area.
- Write the banner message to FILE03.

**Cyborg Scripting Language Report Customization - Participant Guide**

---

<b>Report Section</b>	<b>Report Print (RTPRNT)</b>	<b>Special Print Options</b>
Report Headings	<p>1st extract record is read from FILE14. First 5 characters are checked.</p> <p>Organization Number (Control 1-2) (FILE14) data is moved into working storage.</p> <p>The report headings are assembled in Pointer 11 (SCREEN).</p> <p>Report Headings are written to the print file.</p>	<p>P100-P – Report is Starting paragraph is executed.</p> <p>P120-P – New Organization Number (Control 1-2) is Starting paragraph is executed.</p> <p>P140-P – Header Records are Ready to Print paragraph is executed.</p> <p>P160-P – Header Records Have Been Printed paragraph is executed.</p>
Report Detail	<p>Detail extract records with a matching sort key are read into working storage.</p> <p>Detail data is assembled in Pointer 8 (Print area) by Print Position Record Line Number.</p> <p>Report Detail is written to the print file. The next record is read and assuming that no new Control Break is encountered, the detail for the matching sort key group is performed. New Organization Number (Control 1-2), Heading and Control Break logic is executed intermittently as directed by the input from FILE14.</p>	<p>P180-P – Detail Data With Matching Sort Keys are Ready Paragraph is executed.</p> <p>P220-P – Detail Line is Ready to Print paragraph is executed.</p>
Report Totals	<p>When a record with a new Sort Key is read, totaling occurs.</p> <p>Totals are assembled in Pointer 8 (Print area) by Print Position Record line number.</p> <p>Report Total is written to the print file. The next record is read and assuming that no new Control Break is encountered, the detail for the matching sort key group is performed. New Organization Number (Control 1-2), Heading and Control Break logic is executed intermittently as directed by the input from FILE14.</p> <p>When the Sort Key report name changes, the report is done.</p>	<p>P200-P – Total Data is Ready paragraph is executed.</p> <p>P240-P – A Total Line is ready to print –</p> <p>Pointer 8</p> <p>P260-P – The Report is Done paragraph is executed.</p>

---

**NOTES**

---

## Section summary

In this section, you learned how to override the report output using the Special Print Options program. The above table summarizes the relationship and timing between the Special Print options program and the printing of the report.

Be sure to review the above table thoroughly prior to proceeding to the exercise.

<u>Paragraph</u>	<u>Function</u>
100	The report is starting
120	New Organization Number (Control 1–2) is starting
140	The header records are ready to print
160	The header records have been printed
180	Detail data with matching sort keys is ready
220	A detail line is ready to print
200	Total data is ready
240	A total line is ready to print
260	The report is done

# Section 8 Exercise

CORPORATION	99	ACME MANUFACTURING	HUMAN RESOURCE LISTING				REPT	PERIOD	FILE VERSION	00	PAGE	1
DIVISION	9999	PRODUCTION CTL 1-2	RUNREP PARMS:01-01-2002 / 4444				CLER	PERIOD	TIME	11:27:28	DATE	11-18-2002
EMPLOYEE NAME			CTRL	CTRL	CTRL	JOB	ANNUAL					
			FOUR	FIVE	SIX	CODE	SALARY					
SUSAN A COMPTON			4444	5555	6666	10500	13,841.36					
KEVIN PRESCOTT							16,640.00					
JOHNS A HAYES							9,194.12					
*JOB-CODE	10500	TOTALS:			3		39,675.48			JOB TITLE...		
CHESTERON WARD			4444	5508	6608	10650	27,508.00					
*JOB-CODE	10650	TOTALS:			1		27,508.00			JOB TITLE...		
WILLIAM M. SPENSER			4444	5508	6608	18020	16,634.80					
*JOB-CODE	18020	TOTALS:			1		16,634.80			JOB TITLE...		
GARY D. WENDT			4444	5508	6608	18030	19,760.00					
*JOB-CODE	18030	TOTALS:			1		19,760.00			JOB TITLE...		
JEFFREY C. DANIELS			4444	5508	6608	18040	21,619.00					
*JOB-CODE	18040	TOTALS:			1		21,619.00			JOB TITLE...		

## NOTES

## Section 8 exercise

### Purpose

The purpose of this exercise is to give you practice working with the Special Print Options program logic. Using the report program from section 7, exercise 1 complete the following:

1. Copy the 5M–RP to the new program name (5 characters) and apply the following changes:
  - Suppress the detail line when the detail is for the same Ctrl 4–6 and Job Code.
  - For the \*JOB–CODE and \*REPORT CODE Control Breaks:
    - Move the literal **TOTALS:** to print position 036
  - For the \*REPORT CODE Control Break:
    - Override the total break description with the literal GRAND TOTAL SALARY
    - Blank out the Job Title
  - Schedule, run and view the report.

---

**NOTES**

---

## Appendix A: Exercise Answers

---

### Table of Contents

Section 2—exercise answers .....	258
Section 3—exercise answers .....	263
Section 4—exercise answers .....	264
Section 5—exercise 1 answers .....	268
Section 5—exercise 2 answers .....	269
Section 6—exercise 1 answers .....	275
Section 6—exercise 2 answers .....	279
Section 7—exercise 1 answers .....	283
Section 7—exercise 2 answers .....	287
Section 8—exercise answers .....	292

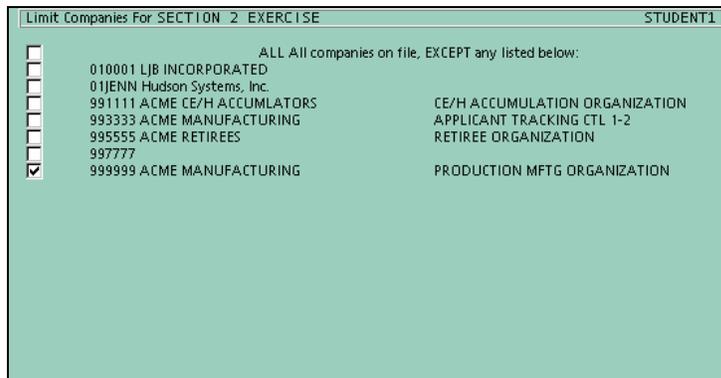
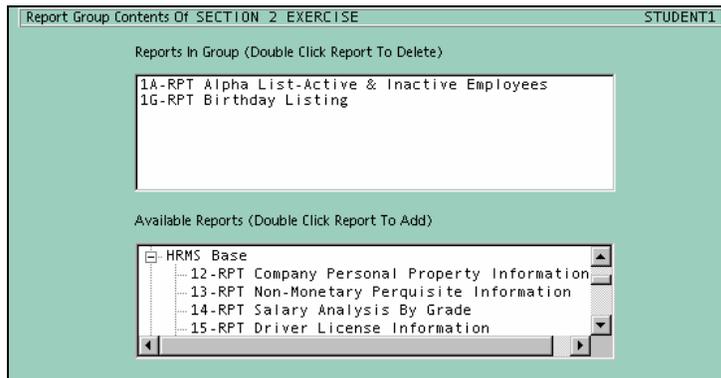
## Section 2—exercise answers

### Purpose

The purpose of this exercise is to give you practice using SUBMIT to initiate the report run and VIEW to see the output.

1. **Schedule the following report for batch submission (Be sure to review HELP documentation for input parameter options)**

- Reports:** Alpha List Active & Inactive Employees (1A-RPT)  
 Birthday Listing (1G-RPT)
- Organization Number** 999999  
**(Control 1-2):**



## Section 2—exercise answers, continued

2. Use Batch Job Initiator form (SUBMIT) to initiate the report process using the report schedule name you established in Step 1. Be sure to route your report output for online review.
  1. Select Tools | Reporting | Report Scheduling | Initiate Scheduled Reports from the menu bar.
  2. Type '1' in the Select Option field.
  3. Press ENTER.
  4. Type the six character Report Group name and type a >Y= to hold output for online review.
  5. Press ENTER.
  6. Follow the instructions on the form to either Log off and submit the job, or the job is submitted interactively for you.
2. Preview the report output using the VIEW program.

```

ACTION ==> V D-Delete, P-Print, V-View ID ==> 031
          ID# PGS  DATE   TIME
          030 0008 04-26-1999 14:32:20
          031 0010 04-26-1999 15:29:29
  
```

```

SIDE ==> L-Left, R-Right PAGE ==> 0002] LINE ==> 01] Pages=0010
CORPORATION 99 ACME MANUFACTURING ALPHABETIC LISTING OF ACTIVE
DIVISION 9999 PRODUCTION MFTG ORGANIZATION AND INACTIVE EMPLOYEES

EMPLOYEE-NAME EMPLOYEE CTRL CTRL CTRL CTRL
NUMBER THREE FOUR FIVE SIX
ADAMS, RICHARD 1117 3030 4040 5050 6060
ALSON, GEOFFERY 3003 3030 4040 5050 6060
ANDERSEN, KARI 6004 3333 4444 5555 6666
ANDERSON, DANIEL M 1616 3333 4444 5555 6666
ANDERSON, KAREN 3034 3388 4488 5508 6608
ANDREWS, HENRY A. 2013 3388 4488 5555 6666
AUSTIN, STEVEN 1234 3030 4040 5050 6060
BALDWIN, ALICE A 1043 3333 4444 5555 6666
BARNES, JOHNSON 2002 3388 4488 5508 6608
BARTHOLOW III, JONATHAN 1113 3333 4444 5555 6666
BEACHEM, JUDITH 6009 3030 4040 5050 6060
BENOWITZ, JAMES 3027 3333 4444 5508 6608
BETTS, J.T. 6003 3030 4040 5050 6060
BISHOP, MARIA 3021 3388 4488 5508 6608
  
```

## Section 2—exercise answers, continued

### Purpose

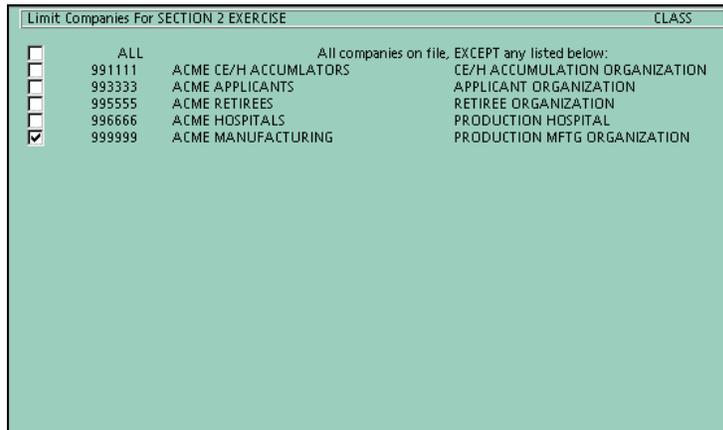
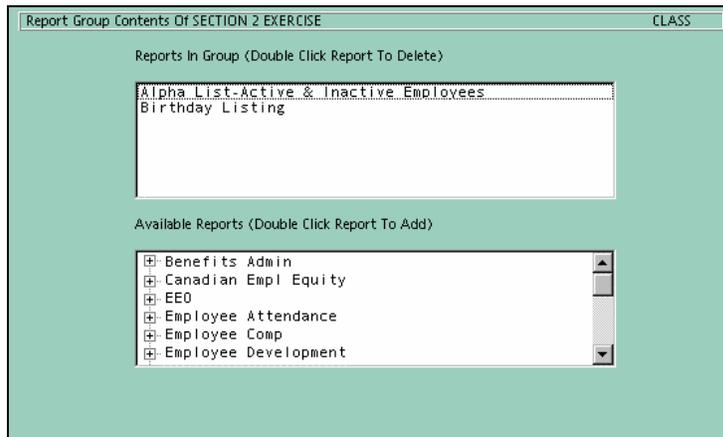
The purpose of this exercise is to give you practice using the Enhanced Reporting features to initiate and review a report.

1. **Schedule the following report for batch submission (Be sure to review HELP documentation for input parameter options):**

**Reports:** Alpha List Active & Inactive Employees (1A-RPT)

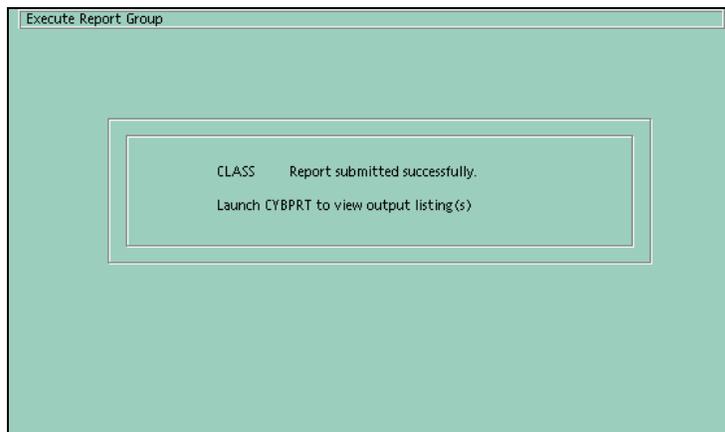
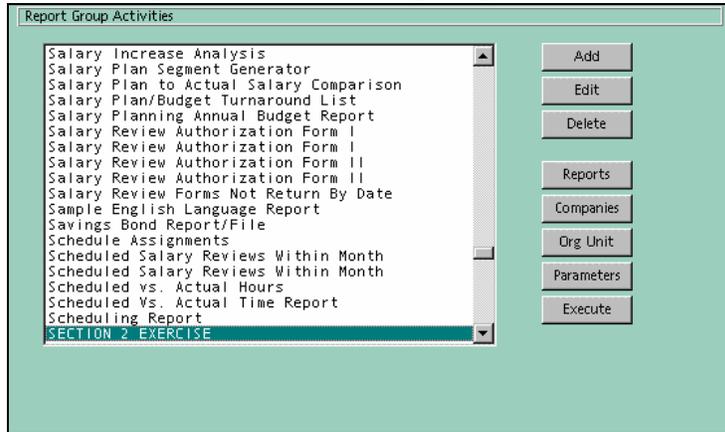
Birthday Listing (1G-RPT)

**Organization Number (Control 1-2):** 999999



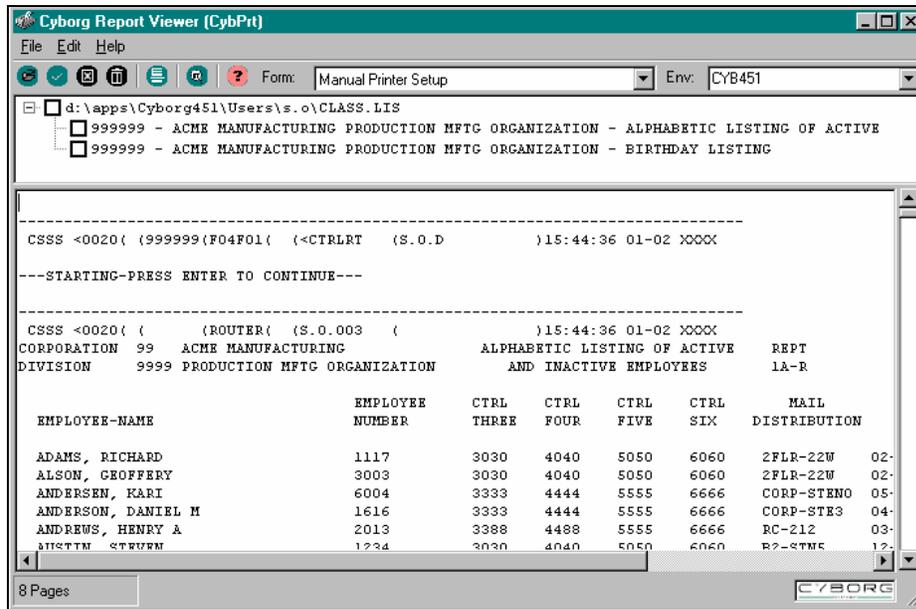
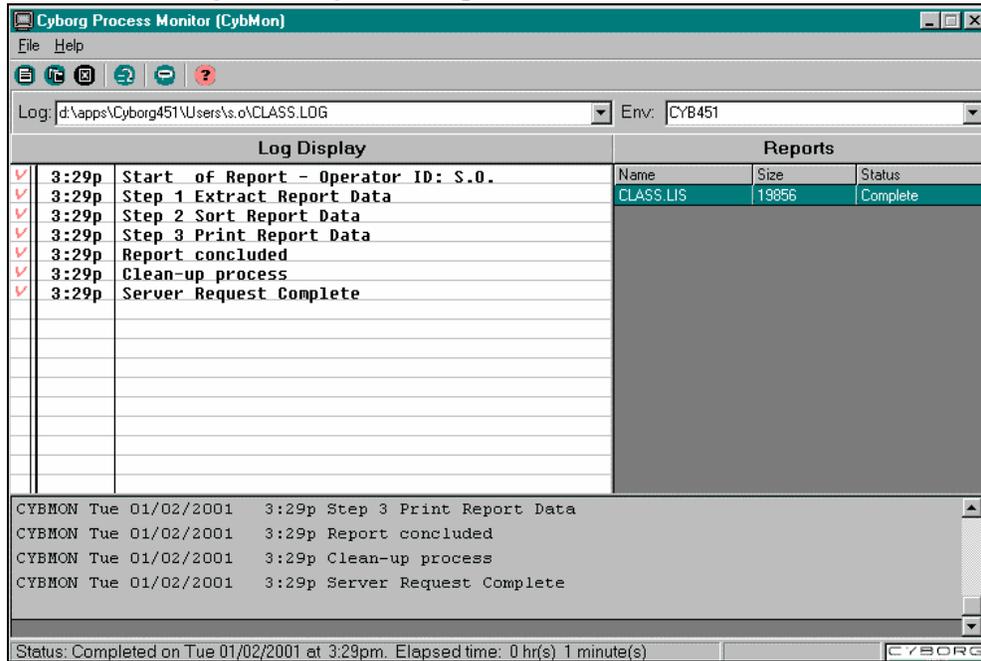
## Section 2—exercise answers, continued

2. Use the Execute button on the Report Group Activities form (RGMSTR) to initiate the report process.



## Section 2—exercise answers, continued

### 3. Review the report output using the Process Monitor.



## Section 3—exercise answers

### Purpose

The purpose of this exercise is to give you practice developing a report extract program.

### 1. Using the EDIT program, develop a report extract program.

#### Report Extract Program

```

P XCLRPT 00000 SECURITY ' ', @ BASIC PAYROLL DATA XPP
P XCLRPT 00001 @LAST MODIFIED ON: BY: AUTHOR: USER
P XCLRPT 00100 DEFINE-REPORT.
P XCLRPT 00200 HEADER-1 :57 'CYBORG CONVERSION'.
P XCLRPT 00300 HEADER-2 :57 'BASIC PAYROLL DATA'.
P XCLRPT 00400 HEADER-3 :33 'EMPLOYEE --- PAY FREQUENCY --'.
P XCLRPT 00500 HEADER-3 :68 '----- PAYMENT -----'.
P XCLRPT 00600 HEADER-3 :99 'SALARY EMPLOYMENT BIRTH'.
P XCLRPT 00700 HEADER-4 :1 'EMPLOYEE NAME'.
P XCLRPT 00800 HEADER-4 :33 'NUMBER'.
P XCLRPT 00900 HEADER-4 :46 'CODE TYPE CODE TYPE'.
P XCLRPT 01000 HEADER-4 :110 'DATE DATE'.
P XCLRPT 01100 P100-INITIALIZE.
P XCLRPT 01200 SET-EMP-PTRS-TO-1ST.
P XCLRPT 01300 SPACE-EXTRACT-RECORD.
P XCLRPT 01400 P200-SELECT.
P XCLRPT 01500 IF SPACES NOT EQUAL UNION-CODE RETURN.
P XCLRPT 01600 P300-SORT.
P XCLRPT 01700 PRINT '1XCLR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL
P XCLRPT 01800 SORT-LENGTH-21 CONTROL-1-2 EMPLOYEE-NUMBER.
P XCLRPT 01900 P400-OUTPUT.
P XCLRPT 02000 PRINT '1' EMPLOYEE-NAME EMPLOYEE-NUMBER
P XCLRPT 02100 PAY-FREQUENCY-CODE PAY-FREQUENCY
P XCLRPT 02200 PAYMENT-CODE PAYMENT-TYPE.
P XCLRPT 02300 OUTPUT SALARY EMPLOYMENT-DATE BIRTH-DATE.
P XCLRPT 02400 WRITE-EXTRACT.
P XCLRPT 02500 RETURN.

```



## Section 4—exercise answers, continued

3. **Schedule your report using the Schedule Report Groups forms (RGMSTR). Include only the Organization Number (Control 1-2) 999999 in your output.**
4. **Use Batch Job Initiator form (SUBMIT) to initiate the report process using the report schedule name you established in Step 1. Be sure to route your report output for online review.**
  1. Select Tools | Reporting | Initiate Scheduled Reports from the menu bar.  
Command: Type SUBMIT in the Screen field.
  2. Type '1' in the Select Option field.
  3. Press ENTER.
  4. Type the six character Report Group name and type a >Y= to hold output for online review.
  5. Press ENTER.
  6. Follow the instructions on the form to either Log off and submit the job, or the job is submitted interactively for you.
5. **Use the View Help Report (VIEW) program to preview your report's output.**
  1. Select Tools | Reporting | View Held Reports from the menu bar.
  2. Command: Type VIEW in the Screen field.
  3. Type 'V' in the Action Field. Type a Report ID in the ID Field.
  4. Press ENTER.
  5. Optionally, type 'L' to view the Left side of the report of 'R' for Right side of the report in the SIDE Field.
  6. Optionally, type a page number in the PAGE Field.
  7. Optionally, type a line number in the LINE Field.
  8. Press ENTER.

## Section 4—exercise answers, continued

### Report output

CORPORATION 99 ACME MANUFACTURING		CYBORG CONVERSION		REPT PERIOD	FILE VERSION 00	PAGE	1
DIVISION 9999 PRODUCTION CTL 1-2		BASIC PAYROLL DATA		XCLR PERIOD	TIME 13:17 DATE	08-30-2002	
EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY ---	--- PAYMENT ---	SALARY	EMPLOYMENT DATE	BIRTH DATE	
		CODE TYPE	CODE TYPE				
MEYER, JUNE	1001	1 Weekly	6 Salary-Auto/No OT Al	461.54	09-15-1984	07-11-1956	
MOORE, SAMUEL	1002	4 Monthly	6 Salary-Auto/No OT Al	4,950.00	10-01-1984	06-30-1945	
MUIR, LINDA	1003	2 Bi Weekly	5 Salary-Auto Paid	615.38	09-23-1984	01-03-1959	
MORSE, GORDAN	1004	4 Monthly	1 Hourly-TE Required	.00	08-30-1984	12-27-1948	
MORRIS, ROBERT	1005	2 Bi Weekly	1 Hourly-TE Required	.00	08-01-1984	08-24-1935	
MORITZ, KATHERINE C.	1007	4 Monthly	6 Salary-Auto/No OT Al	3,750.00	09-01-1984	12-20-1942	
MAURICE, STACY E.	1008	1 Weekly	5 Salary-Auto Paid	370.00	09-06-1984	01-17-1960	
MOREAU, GARDNER	1009	2 Bi Weekly	5 Salary-Auto Paid	825.00	09-01-1984	08-16-1958	
BALDWIN, ALICE A	1043	1 Weekly	1 Hourly-TE Required	.00	08-20-1983	09-08-1950	
JOHNSEN, RICH DANIEL	1112	4 Monthly	6 Salary-Auto/No OT Al	5,583.33	12-19-1979	11-07-1932	
BARTHOLOW III, JONATHAN	1113	4 Monthly	6 Salary-Auto/No OT Al	3,166.67	03-12-1982	02-16-1938	
WELKER, GEORGE W	1114	4 Monthly	6 Salary-Auto/No OT Al	2,333.33	05-15-1983	05-07-1939	
ADAMS, RICHARD	1117	3 Semi Monthly	5 Salary-Auto Paid	1,333.33	02-19-1984	02-01-1939	
MANNING, WILLIAM Z.	1165	3 Semi Monthly	2 Salary-TE Required	800.00	10-22-1977	09-23-1950	
BARKER, MARTINN A	1184	3 Semi Monthly	5 Salary-Auto Paid	1,375.00	10-03-1983	09-11-1946	
HAMMER, JAMES B.	1236	1 Weekly	2 Salary-TE Required	352.35	07-22-1983	11-05-1935	
JORDAN, WILLIAM M.	1257	2 Bi Weekly	5 Salary-Auto Paid	440.00	01-01-1981	02-11-1940	
ISLEY, JEANETTE J.	1432	1 Weekly	1 Hourly-TE Required	.00	03-17-1983	12-16-1947	
MAGUIRE, HENRY S.	1578	1 Weekly	2 Salary-TE Required	222.00	02-28-1984	05-29-1946	
ANDERSON, DANIEL M	1616	3 Semi Monthly	5 Salary-Auto Paid	1,068.75	04-24-1984	07-17-1948	
AYERS, CHESTER D	1755	2 Bi Weekly	2 Salary-TE Required	760.00	02-05-1980	12-02-1946	
HANCOCK, STEVEN W.	1764	3 Semi Monthly	2 Salary-TE Required	800.00	04-07-1984	01-17-1942	
COLLINS, ANNA MARIE	1848	3 Semi Monthly	5 Salary-Auto Paid	1,416.67	09-26-1983	03-14-1939	
GRIMES, THEODORE J	1975	2 Bi Weekly	5 Salary-Auto Paid	1,923.08	06-01-1979	10-11-1940	
REYNOLDS, BRENDA	2001	1 Weekly	5 Salary-Auto Paid	.00	04-15-1985	07-11-1956	
BARNES, JOHNSON	2002	4 Monthly	6 Salary-Auto/No OT Al	8,463.94	05-01-1985	06-30-1945	
CMEYLA, JANE	2003	2 Bi Weekly	5 Salary-Auto Paid	513.08	05-23-1985	01-03-1959	
KWONG, STEVEN S.	2004	4 Monthly	6 Salary-Auto/No OT Al	1,637.97	06-30-1985	12-27-1948	
BROWN, WILLIAM R	2005	2 Bi Weekly	6 Salary-Auto/No OT Al	603.65	07-01-1985	08-24-1935	
COSTELLO, SUSANNE	2006	1 Weekly	6 Salary-Auto/No OT Al	.00	07-10-1985	02-14-1930	
LANNON, PATRICE	2007	4 Monthly	6 Salary-Auto/No OT Al	3,683.33	09-01-1985	12-20-1942	
HALL, RHONDA D.	2008	1 Weekly	6 Salary-Auto/No OT Al	.00	09-06-1985	01-17-1960	
CREMMINS, ALAN EDWARD	2009	2 Bi Weekly	6 Salary-Auto/No OT Al	1,308.32	12-01-1985	08-16-1958	
PITARO, JOSEPH C.	2011	1 Weekly	1 Hourly-TE Required	.00	02-15-1986	07-01-1967	
WARD, CHESTERON	2012	1 Weekly	1 Hourly-TE Required	.00	03-01-1986	09-13-1951	
ANDREWS, HENRY A.	2013	2 Bi Weekly	2 Salary-TE Required	650.42	03-09-1986	04-20-1959	
GRANT, KEITH L.	2014	1 Weekly	3 Salary-TE/No OT Allw	.00	03-15-1986	04-04-1961	
MARGOLIS, DAVID	2015	3 Semi Monthly	3 Salary-TE/No OT Allw	1,687.50	04-01-1986	12-20-1956	
SHEA, JEFFERY B.	2016	1 Weekly	2 Salary-TE Required	.00	04-11-1986	03-13-1954	
BLOOM, ALEXANDER	3001	3 Semi Monthly	6 Salary-Auto/No OT Al	1,583.33	03-12-1978	02-16-1938	
WALSH, THEODORE	3002	4 Monthly	6 Salary-Auto/No OT Al	2,333.33	05-15-1983	05-07-1939	
ALSON, GEOFFERY	3003	3 Semi Monthly	5 Salary-Auto Paid	1,333.33	02-19-1984	02-01-1939	
MARSH, PAUL J.	3004	3 Semi Monthly	2 Salary-TE Required	800.00	03-22-1985	09-23-1950	
SWALTER, STEVEN Y	3005	1 Weekly	2 Salary-TE Required	352.35	07-22-1980	11-05-1935	
WARREN, MICHAEL	3006	3 Semi Monthly	5 Salary-Auto Paid	1,068.75	09-25-1985	07-17-1948	
HILLERY, THOMAS	3007	3 Semi Monthly	2 Salary-TE Required	800.00	04-07-1984	01-17-1942	
GRIFFITH, BERNARD	3008	2 Bi Weekly	5 Salary-Auto Paid	1,923.08	06-01-1984	10-11-1940	
TEACHEN, JUDITH	3009	3 Semi Monthly	5 Salary-Auto Paid	1,416.67	09-26-1983	03-14-1939	
RUNYON, BRENDA	3010	1 Weekly	6 Salary-Auto/No OT Al	461.54	02-01-1975	07-11-1955	
LAUGHLIN, SANDRA T.	3011	4 Monthly	6 Salary-Auto/No OT Al	3,750.00	09-01-1984	12-20-1942	
SCHAEFER, JOANNA S.	3012	1 Weekly	5 Salary-Auto Paid	370.00	09-06-1984	01-17-1960	
CARLILE, WILLIAM E.	3013	2 Bi Weekly	5 Salary-Auto Paid	825.00	09-01-1984	08-16-1958	
SULLIVAN, MIKE M.	3014	1 Weekly	2 Salary-TE Required	222.00	02-28-1984	05-29-1946	

CORPORATION 99 ACME MANUFACTURING		SOLUTION SERIES CONVERSION		REPT PERIOD	FILE VERSION 00	PAGE 2
DIVISION 9999 PRODUCTION CTL 1-2		BASIC PAYROLL DATA		XCLR PERIOD	TIME 13:17 DATE 08-30-2002	
EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY --	----- PAYMENT -----	SALARY	EMPLOYMENT DATE	BIRTH DATE
		CODE TYPE	CODE TYPE			
LLEWELYN, STEVE	3015	2 Bi Weekly	5 Salary-Auto Paid	926.92	08-12-1976	07-22-1938
SANTANA, LOUISE	3016	1 Weekly	2 Salary-TE Required	362.98	06-15-1982	05-17-1955
STENMAN, SAMANTHA	3017	3 Semi Monthly	5 Salary-Auto Paid	831.25	10-14-1982	01-08-1953
THEISSEN, LEONARD	3018	2 Bi Weekly	6 Salary-Auto/No OT Al	1,192.26	08-01-1981	06-15-1951
HARRIS, CECELIA	3019	2 Bi Weekly	5 Salary-Auto Paid	615.00	03-19-1980	02-04-1949
CHOU, LO	3020	1 Weekly	2 Salary-TE Required	726.00	05-01-1981	06-25-1953
BISHOP, MARIA	3021	1 Weekly	6 Salary-Auto/No OT Al	414.98	07-07-1976	09-12-1933
WILSON, BARBARA	3025	2 Bi Weekly	5 Salary-Auto Paid	800.00	06-15-1975	11-14-1926
LEWIS, JAMES X.	3026	3 Semi Monthly	6 Salary-Auto/No OT Al	1,416.66	09-20-1962	07-22-1923
BENOWITZ, JAMES	3027	2 Bi Weekly	5 Salary-Auto Paid	1,540.00	01-15-1966	04-11-1932
CORTEZ, MARIA	3028	3 Semi Monthly	6 Salary-Auto/No OT Al	1,020.83	03-15-1965	12-12-1924
DUNBAR, WALLCOTT A.	3029	2 Bi Weekly	2 Salary-TE Required	996.00	03-15-1965	11-01-1920
PATEL, LAURENCE	3030	2 Bi Weekly	5 Salary-Auto Paid	1,665.96	02-11-1966	05-25-1923
EMORY, TODD	3031	2 Bi Weekly	5 Salary-Auto Paid	.00	06-15-1973	09-20-1949
OWENS, LORNA	3032	2 Bi Weekly	2 Salary-TE Required	.00	01-22-1981	03-31-1936
YOUNG, J.T.	3033	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	03-11-1985	04-22-1956
ANDERSON, KAREN	3034	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	05-16-1974	10-17-1950
LITTLE, DORA	6002	2 Bi Weekly	2 Salary-TE Required	.00	01-22-1981	03-31-1936
BETTS, J.T.	6003	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	03-11-1985	04-22-1956
ANDERSEN, KARI	6004	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	05-16-1974	10-17-1950
SANDERS, STEVEN S	6005	1 Weekly	2 Salary-TE Required	.00	07-22-1980	11-05-1935
WINTER, MICHAEL	6006	3 Semi Monthly	5 Salary-Auto Paid	.00	09-25-1985	07-17-1948
MILLER, THOMAS S	6007	3 Semi Monthly	2 Salary-TE Required	.00	04-07-1984	01-17-1942
GRIFFITHS, ROBERT	6008	2 Bi Weekly	5 Salary-Auto Paid	.00	06-01-1984	10-11-1940
BEACHEM, JUDITH	6009	3 Semi Monthly	5 Salary-Auto Paid	.00	09-26-1983	03-14-1939
ITTER, BRENDA	6010	1 Weekly	6 Salary-Auto/No OT Al	.00	02-01-1975	07-11-1955

## Section 5—exercise 1 answers

### Purpose

The purpose of this exercise is to give you practice coding sort logic for multiple control breaks. Directions assume an employee count at the control breaks indicated. Use the space below.

### Sort logic

```
SET-EMP-PTRS-TO-1ST.  
PRINT '1ANYR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL TRIPLE-SPACE-BEFORE  
CONTROL-1-2 NO-PRINT-SUBTOTAL  
PAY-FREQUENCY-CODE NO-PRINT-SUBTOTAL NEW-PAGE-BEFORE  
CONTROL-4-CODE DOUBLE-SPACE-BEFORE DOUBLE-SPACE-AFTER  
SORT-LENGTH-41  
EMPLOYEE-NAME-15  
EMPLOYEE-NUMBER.
```

## Section 5—exercise 2 answers

### Report extract program

```

P X5-RPT 00000 SECURITY 'PP'. @ Detail with Totals - Exercise 5          XPP
P X5-RPT 00001 @LAST MODIFIED ON: 09-29-99 BY: USER AUTHOR: USER
P X5-RPT 00100 @This is the second report exercise.
P X5-RPT 00200 DEFINE-REPORT.
P X5-RPT 00300 HEADER-1 :059 'CLASS REPORT'.
P X5-RPT 00400 HEADER-2 :055 'PAY REQUENCY TOTALS'.
P X5-RPT 00500 HEADER-4 :001 'EMPLOYEE NAME'.
P X5-RPT 00600 HEADER-3 :033 'EMPLOYEE'.
P X5-RPT 00700 HEADER-4 :033 'NUMBER'.
P X5-RPT 00800 HEADER-3 :046 '--- PAY FREQUENCY ---'.
P X5-RPT 00900 HEADER-4 :046 'CODE          TYPE'.
P X5-RPT 01000 HEADER-3 :069 '----- PAYMENT -----'.
P X5-RPT 01100 HEADER-4 :069 'CODE          TYPE'.
P X5-RPT 01200 HEADER-4 :099 'SALARY'.
P X5-RPT 01300 HEADER-3 :110 'EMPLOYMENT      BIRTH'.
P X5-RPT 01400 HEADER-4 :110 '    DATE          DATE'.
P X5-RPT 01500 P100-INITIALIZE. @ INITIALIZE COUNTERS HERE IF NEEDED.
P X5-RPT 01600     SET-EMP-PTRS-TO-1ST.
P X5-RPT 01700     MOVE :1 TO PERM-01-V0.
P X5-RPT 01800 P200-SELECT. @RECORD SELECTION LOGIC GOES HERE
P X5-RPT 01900     IF SPACES NOT EQUAL UNION-CODE RETURN.
P X5-RPT 02000 P300-SORT.
P X5-RPT 02100     PRINT '1X5-R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL
P X5-RPT 02200             CONTROL-1-2 NO-PRINT-SUBTOTAL
P X5-RPT 02300             PAY-FREQUENCY-CODE DOUBLE-SPACE-BEFORE
P X5-RPT 02400             DOUBLE-SPACE-AFTER SORT-LENGTH-42
P X5-RPT 02500             EMPLOYEE-NAME-20 EMPLOYEE-NUMBER.
P X5-RPT 02600 P400-OUTPUT.
P X5-RPT 02700     PRINT '00' EMPLOYEE-NAME EMPLOYEE-NUMBER
P X5-RPT 02800             PAY-FREQUENCY-CODE PAY-FREQUENCY PAYMENT-CODE
P X5-RPT 02900             PAYMENT-TYPE.
P X5-RPT 03000     OUTPUT SALARY EMPLOYMENT-DATE BIRTH-DATE PERM-01-V0.
P X5-RPT 03100     WRITE-EXTRACT.

```



## Section 5—exercise 2 answers, continued

## Report output

CORPORATION 99 ACME MANUFACTURING		CLASS REPORT		REPT PERIOD	FILE VERSION 00	PAGE 1
DIVISION 9999	PRODUCTION CTL 1-2	PAY FREQUENCY	TOTALS	X5-R PERIOD	TIME 13:54	DATE 08-30-2002
EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY --- CODE TYPE	----- PAYMENT ----- CODE TYPE	SALARY	EMPLOYMENT DATE	BIRTH DATE
BALDWIN, ALICE A	1043	1 Weekly	1 Hourly-TE Required	.00	08-20-1983	09-08-1950
BISHOP, MARIA	3021	1 Weekly	6 Salary-Auto/No OT Al	414.98	07-07-1976	09-12-1933
CHOU, LO	3020	1 Weekly	2 Salary-TE Required	726.00	05-01-1981	06-25-1953
COSTELLO, SUSANNE	2006	1 Weekly	6 Salary-Auto/No OT Al	.00	07-10-1985	02-14-1930
GRANT, KEITH L.	2014	1 Weekly	3 Salary-TE/No OT Allw	.00	03-15-1986	04-04-1961
HALL, RHONDA D.	2008	1 Weekly	6 Salary-Auto/No OT Al	.00	09-06-1985	01-17-1960
HAMMER, JAMES B.	1236	1 Weekly	2 Salary-TE Required	352.35	07-22-1983	11-05-1935
ISLEY, JEANETTE J.	1432	1 Weekly	1 Hourly-TE Required	.00	03-17-1983	12-16-1947
MAGUIRE, HENRY S.	1578	1 Weekly	2 Salary-TE Required	222.00	02-28-1984	05-29-1946
MAURICE, STACY E.	1008	1 Weekly	5 Salary-Auto Paid	370.00	09-06-1984	01-17-1960
MEYER, JUNE	1001	1 Weekly	6 Salary-Auto/No OT Al	461.54	09-15-1984	07-11-1956
PITARO, JOSEPH C.	2011	1 Weekly	1 Hourly-TE Required	.00	02-15-1986	07-01-1967
REYNOLDS, BRENDA	2001	1 Weekly	5 Salary-Auto Paid	.00	04-15-1985	07-11-1956
RITTER, BRENDA	6010	1 Weekly	6 Salary-Auto/No OT Al	.00	02-01-1975	07-11-1955
RUNYON, BRENDA	3010	1 Weekly	6 Salary-Auto/No OT Al	461.54	02-01-1975	07-11-1955
SANDERS, STEVEN S	6005	1 Weekly	2 Salary-TE Required	.00	07-22-1980	11-05-1935
SANTANA, LOUISE	3016	1 Weekly	2 Salary-TE Required	362.98	06-15-1982	05-17-1955
SCHAEFER, JOANNA S.	3012	1 Weekly	5 Salary-Auto Paid	370.00	09-06-1984	01-17-1960
SHEA, JEFFERY B.	2016	1 Weekly	2 Salary-TE Required	.00	04-11-1986	03-13-1954
SULLIVAN, MIKE M.	3014	1 Weekly	2 Salary-TE Required	222.00	02-28-1984	05-29-1946
SWALTER, STEVEN Y	3005	1 Weekly	2 Salary-TE Required	352.35	07-22-1980	11-05-1935
WARD, CHESTERON	2012	1 Weekly	1 Hourly-TE Required	.00	03-01-1986	09-13-1951
*PAY-FREQUENCY-C 1		22		4,315.74		
ANDREWS, HENRY A.	2013	2 Bi Weekly	2 Salary-TE Required	650.42	03-09-1986	04-20-1959
AYERS, CHESTER D	1755	2 Bi Weekly	2 Salary-TE Required	760.00	02-05-1980	12-02-1946
BENOWITZ, JAMES	3027	2 Bi Weekly	5 Salary-Auto Paid	1,540.00	01-15-1966	04-11-1932
BROWN, WILLIAM R	2005	2 Bi Weekly	6 Salary-Auto/No OT Al	603.65	07-01-1985	08-24-1935

Cyborg Scripting Language Report Customization - Participant Guide

Report output, continued

CORPORATION 99 ACME MANUFACTURING		CLASS REPORT		REPT PERIOD	FILE VERSION 00	PAGE 2
DIVISION 9999 PRODUCTION CTL 1-2		PAY FREQUENCY TOTALS		X5-R PERIOD	TIME 13:54 DATE 08-30-2002	
EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY --	----	PAYMENT -----	SALARY	EMPLOYMENT BIRTH
		CODE TYPE		CODE TYPE		DATE DATE
CARLILE, WILLIAM E.	3013	2 Bi Weekly		5 Salary-Auto Paid	825.00	09-01-1984 08-16-1958
CMEYLA, JANE	2003	2 Bi Weekly		5 Salary-Auto Paid	513.08	05-23-1985 01-03-1959
CREMMINS, ALAN EDWARD	2009	2 Bi Weekly		6 Salary-Auto/No OT Al	1,308.32	12-01-1985 08-16-1958
DUNBAR, WALLCOTT A.	3029	2 Bi Weekly		2 Salary-TE Required	996.00	03-15-1965 11-01-1920
EMORY, TODD	3031	2 Bi Weekly		5 Salary-Auto Paid	.00	06-15-1973 09-20-1949
GRIFFITH, BERNARD	3008	2 Bi Weekly		5 Salary-Auto Paid	1,923.08	06-01-1984 10-11-1940
GRIFFITHS, ROBERT	6008	2 Bi Weekly		5 Salary-Auto Paid	.00	06-01-1984 10-11-1940
GRIMES, THEODORE J	1975	2 Bi Weekly		5 Salary-Auto Paid	1,923.08	06-01-1979 10-11-1940
HARRIS, CECELIA	3019	2 Bi Weekly		5 Salary-Auto Paid	615.00	03-19-1980 02-04-1949
JORDAN, WILLIAM M.	1257	2 Bi Weekly		5 Salary-Auto Paid	440.00	01-01-1981 02-11-1940
LITTLE, DORA	6002	2 Bi Weekly		2 Salary-TE Required	.00	01-22-1981 03-31-1936
LLEWELYN, STEVE	3015	2 Bi Weekly		5 Salary-Auto Paid	926.92	08-12-1976 07-22-1938
MOREAU, GARDNER	1009	2 Bi Weekly		5 Salary-Auto Paid	825.00	09-01-1984 08-16-1958
MORRIS, ROBERT	1005	2 Bi Weekly		1 Hourly-TE Required	.00	08-01-1984 08-24-1935
MUIR, LINDA	1003	2 Bi Weekly		5 Salary-Auto Paid	615.38	09-23-1984 01-03-1959
OWENS, LORNA	3032	2 Bi Weekly		2 Salary-TE Required	.00	01-22-1981 03-31-1936
PATEL, LAURENCE	3030	2 Bi Weekly		5 Salary-Auto Paid	1,665.96	02-11-1966 05-25-1923
THEISSEN, LEONARD	3018	2 Bi Weekly		6 Salary-Auto/No OT Al	1,192.26	08-01-1981 06-15-1951
WILSON, BARBARA	3025	2 Bi Weekly		5 Salary-Auto Paid	800.00	06-15-1975 11-14-1926
*PAY-FREQUENCY-C 2		23			18,123.15	
ADAMS, RICHARD	1117	3 Semi Monthly		5 Salary-Auto Paid	1,333.33	02-19-1984 02-01-1939
ALSON, GEOFFERY	3003	3 Semi Monthly		5 Salary-Auto Paid	1,333.33	02-19-1984 02-01-1939
ANDERSEN, KARI	6004	3 Semi Monthly		6 Salary-Auto/No OT Al	.00	05-16-1974 10-17-1950
ANDERSON, DANIEL M	1616	3 Semi Monthly		5 Salary-Auto Paid	1,068.75	04-24-1984 07-17-1948
ANDERSON, KAREN	3034	3 Semi Monthly		6 Salary-Auto/No OT Al	.00	05-16-1974 10-17-1950
BARKER, MARTINN A	1184	3 Semi Monthly		5 Salary-Auto Paid	1,375.00	10-03-1983 09-11-1946
BEACHEM, JUDITH	6009	3 Semi Monthly		5 Salary-Auto Paid	.00	09-26-1983 03-14-1939

Report output, continued

CORPORATION 99 ACME MANUFACTURING		CLASS REPORT		REPT PERIOD	FILE VERSION 00	PAGE 3
DIVISION 9999 PRODUCTION CTL 1-2		PAY FREQUENCY TOTALS		X5-R PERIOD	TIME 13:54 DATE 08-30-2002	
EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY -- CODE TYPE	----- PAYMENT ----- CODE TYPE	SALARY	EMPLOYMENT DATE	BIRTH DATE
BETTS, J.T.	6003	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	03-11-1985	04-22-1956
BLOOM, ALEXANDER	3001	3 Semi Monthly	6 Salary-Auto/No OT Al	1,583.33	03-12-1978	02-16-1938
COLLINS, ANNA MARIE	1848	3 Semi Monthly	5 Salary-Auto Paid	1,416.67	09-26-1983	03-14-1939
CORTEZ, MARIA	3028	3 Semi Monthly	6 Salary-Auto/No OT Al	1,020.83	03-15-1965	12-12-1924
HANCOCK, STEVEN W.	1764	3 Semi Monthly	2 Salary-TE Required	800.00	04-07-1984	01-17-1942
HILLERY, THOMAS	3007	3 Semi Monthly	2 Salary-TE Required	800.00	04-07-1984	01-17-1942
LEWIS, JAMES X.	3026	3 Semi Monthly	6 Salary-Auto/No OT Al	1,416.66	09-20-1962	07-22-1923
MANNING, WILLIAM Z.	1165	3 Semi Monthly	2 Salary-TE Required	800.00	10-22-1977	09-23-1950
MARGOLIS, DAVID	2015	3 Semi Monthly	3 Salary-TE/No OT Allw	1,687.50	04-01-1986	12-20-1956
MARSH, PAUL J.	3004	3 Semi Monthly	2 Salary-TE Required	800.00	03-22-1985	09-23-1950
MILLER, THOMAS S	6007	3 Semi Monthly	2 Salary-TE Required	.00	04-07-1984	01-17-1942
STENMAN, SAMANTHA	3017	3 Semi Monthly	5 Salary-Auto Paid	831.25	10-14-1982	01-08-1953
TEACHEN, JUDITH	3009	3 Semi Monthly	5 Salary-Auto Paid	1,416.67	09-26-1983	03-14-1939
WARREN, MICHAEL	3006	3 Semi Monthly	5 Salary-Auto Paid	1,068.75	09-25-1985	07-17-1948
WINTER, MICHAEL	6006	3 Semi Monthly	5 Salary-Auto Paid	.00	09-25-1985	07-17-1948
YOUNG, J.T.	3033	3 Semi Monthly	6 Salary-Auto/No OT Al	.00	03-11-1985	04-22-1956
*PAY-FREQUENCY-C 3		23		18,752.07		
BARNES, JOHNSON	2002	4 Monthly	6 Salary-Auto/No OT Al	8,463.94	05-01-1985	06-30-1945
BARTHOLOW III, JONATHAN	1113	4 Monthly	6 Salary-Auto/No OT Al	3,166.67	03-12-1982	02-16-1938
JOHNSEN, RICH DANIEL	1112	4 Monthly	6 Salary-Auto/No OT Al	5,583.33	12-19-1979	11-07-1932
KWONG, STEVEN S.	2004	4 Monthly	6 Salary-Auto/No OT Al	1,637.97	06-30-1985	12-27-1948
LANNON, PATRICE	2007	4 Monthly	6 Salary-Auto/No OT Al	3,683.33	09-01-1985	12-20-1942
LAUGHLIN, SANDRA T.	3011	4 Monthly	6 Salary-Auto/No OT Al	3,750.00	09-01-1984	12-20-1942
MOORE, SAMUEL	1002	4 Monthly	6 Salary-Auto/No OT Al	4,950.00	10-01-1984	06-30-1945
MORITZ, KATHERINE C.	1007	4 Monthly	6 Salary-Auto/No OT Al	3,750.00	09-01-1984	12-20-1942
MORSE, GORDAN	1004	4 Monthly	1 Hourly-TE Required	.00	08-30-1984	12-27-1948
WALSH, THEODORE	3002	4 Monthly	6 Salary-Auto/No OT Al	2,333.33	05-15-1983	05-07-1939

**Report output, continued**

CORPORATION 99 ACME MANUFACTURING	CLASS REPORT	REPT PERIOD	FILE VERSION 00	PAGE 4
DIVISION 9999 PRODUCTION CTL 1-2	PAY FREQUENCY TOTALS	X5-R PERIOD	TIME 13:54 DATE 08-30-2002	

EMPLOYEE NAME	EMPLOYEE NUMBER	--- PAY FREQUENCY -- CODE TYPE	----- PAYMENT ----- CODE TYPE	SALARY	EMPLOYMENT DATE	BIRTH DATE
WELKER, GEORGE W	1114	4 Monthly	6 Salary-Auto/No OT A1	2,333.33	05-15-1983	05-07-1939
*PAY-FREQUENCY-C 4		11		39,651.90		
*REPORT CODE X5-R		79		80,842.86		

## Section 6—exercise 1 answers

## Report extract program

```

P X6ARPT 00000 SECURITY ' '. @ CLASS REPORT EXAMPLE XPP
P X6ARPT 00001 @LAST MODIFIED ON: BY: AUTHOR: USER
P X6ARPT 00002 @PARMS=GRPRSC
P X6ARPT 00100 DEFINE-REPORT ALLOCATE-10.
P X6ARPT 00200 HEADER-1 :52 'HUMAN RESOURCE LISTING'.
P X6ARPT 00300 HEADER-3 :37 'CTRL CTRL CTRL JOB'.
P X6ARPT 00400 HEADER-4 :01 'EMPLOYEE NAME'.
P X6ARPT 00500 HEADER-4 :37 'FOUR FIVE SIX CODE'.
P X6ARPT 00600 HEADER-3 :79 'ANNUAL'.
P X6ARPT 00700 HEADER-4 :79 'SALARY'.
P X6ARPT 00800 P100-INITIALIZE.
P X6ARPT 00900 SET-EMP-PTRS-TO-1ST.
P X6ARPT 00950 IF FIRST-TIME-IND NOT EQUAL 'F' GO TO P200-SELECT.
P X6ARPT 01000 IF SPACES EQUAL W6-06-036
P X6ARPT 01100 MOVE CURRENT-DATE-CYYMDD TO SPECIAL-DATE
P X6ARPT 01200 MOVE 'Y' TO FIRST-TIME-IND.
P X6ARPT 01300 P200-SELECT.
P X6ARPT 01500 FIND CTRL-FOUR STARTING WITH SPECIAL-DATE.
P X6ARPT 01600 MATCH-SEGMENT-CODE.
P X6ARPT 01700 IF NOT FOUND RETURN.
P X6ARPT 01800 IF SPACES EQUALS W6-04-042
P X6ARPT 01900 GO TO P300-SORT.
P X6ARPT 02000 IF W6-04-042 NOT EQUAL CTRL-FOUR RETURN.
P X6ARPT 02100 P300-SORT.
P X6ARPT 02200 PRINT '1X6AR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL
P X6ARPT 02250 DOUBLE-SPACE-BEFORE
P X6ARPT 02300 CONTROL-1-2 NO-PRINT-SUBTOTAL
P X6ARPT 02400 CTRL-FOUR DOUBLE-SPACE-BEFORE DOUBLE-SPACE-AFTER
P X6ARPT 02500 SORT-LENGTH-33
P X6ARPT 02600 CTRL-FIVE CTRL-SIX EMPLOYEE-NUMBER.
P X6ARPT 02650 P400-OUTPUT.
P X6ARPT 02700 PRINT '0' EMPLOYEE-NAME CTRL-FOUR CTRL-FIVE CTRL-SIX.
P X6ARPT 02800 FIND JOB-CODE STARTING WITH SPECIAL-DATE.
P X6ARPT 02900 MATCH-SEGMENT-CODE.
P X6ARPT 03000 IF FOUND PRINT JOB-CODE
P X6ARPT 03100 ELSE SPACE-OVER :06.
P X6ARPT 03200 MOVE SPECIAL-DATE TO SALARY-AS-OF-DATE.
P X6ARPT 03300 FIND-SALARY-AS-OF.
P X6ARPT 03400 IF FOUND OUTPUT ANNUAL-SALARY
P X6ARPT 03500 ELSE OUTPUT '000000000'.
P X6ARPT 03700 WRITE-EXTRACT.
P X6ARPT 03800 RETURN.

```

## Section 6—exercise 1 answers, continued

### Print positions layout

SEQ	FIELD	-----PRINT-----		
NBR	NAME	POS.	LINE	TOTAL
10	EMPLOYEE-NAME	001	01	N
20	CTRL-FOUR	036	01	N
30	CTRL-FIVE	046	01	N
40	CTRL-SIX	056	01	N
50	JOB-CODE	064	01	N
60	ANNUAL-SALARY	074	01	Y

## Section 6—exercise 1 answers, continued

## Report output

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 1  
 DIVISION 9999 PRODUCTION CTL 1-2 X6AR PERIOD TIME 14:09 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY
MORRIS, ROBERT	4040	5050	6060	15405	11,086.40
JONES, JERRY	4040	5050	6060	17857	13,302.64
JOHNSEN, RICH DANIEL	4040	5050	6060	68000	68,499.96
SWEENEY, BARBARA	4040	5050	6060	15405	10,201.88
ADAMS, RICHARD	4040	5050	6060	47214	38,221.44
BARKER, MARTINN A	4040	5050	6060	35050	33,000.00
AUSTIN, STEVEN	4040	5050	6060	17857	8,926.84
HAMMER, JAMES B.	4040	5050	6060	22200	21,172.84
JOHNSON, WALTER D	4040	5050	6060	17857	7,881.64
JORDAN, WILLIAM M.	4040	5050	6060	20002	11,440.00
CACH, ROBERT	4040	5050	6060	15405	9,056.32
LYNDEN, ANNETTE C.	4040	5050	6060	17857	8,710.00
MAGUIRE, HENRY S.	4040	5050	6060	20002	14,099.80
COLLINS, ANNA MARIE	4040	5050	6060	47214	39,657.60
GRIMES, THEODORE J	4040	5050	6060	68000	62,999.82
PITARO, JOSEPH C.	4040	5050	6060	15405	8,840.00
GRANT, KEITH L.	4040	5050	6060	18020	19,760.00
SHEA, JEFFERY B.	4040	5050	6060	18020	20,696.00
ALSON, GEOFFERY	4040	5050	6060	47214	31,999.92
SWALTER, STEVEN Y	4040	5050	6060	22200	18,322.20
GRIFFITH, BERNARD	4040	5050	6060	47214	50,000.08
TEACHEN, JUDITH	4040	5050	6060	47214	36,720.00
SULLIVAN, MIKE M.	4040	5050	6060	20002	11,544.00
BETTS, J.T.	4040	5050	6060	47214	32,000.00
SANDERS, STEVEN S	4040	5050	6060	22200	18,322.20
GRIFFITHS, ROBERT	4040	5050	6060	47214	50,000.08
BEACHEM, JUDITH	4040	5050	6060	47214	34,000.08
*CTRL-FOUR 4040					690,461.74
WARD, CHESTERON	4444	5508	6608	10650	27,508.00
WENDT, GARY D.	4444	5508	6608	18030	19,760.00
DANIELS, JEFFREY C.	4444	5508	6608	18040	21,619.00
SPENSER, WILLIAM M.	4444	5508	6608	18020	16,634.80
BENOWITZ, JAMES	4444	5508	6608	10600	40,040.00
DUNBAR, WALLCOTT A.	4444	5508	6608	18040	25,896.00
EMORY, TODD	4444	5508	6608	10600	40,040.00
YOUNG, J.T.	4444	5508	6608	18040	23,904.00
BALDWIN, ALICE A	4444	5555	6666	20002	17,246.84
BARTHOLOW III, JONATHAN	4444	5555	6666	35050	39,900.00
WELKER, GEORGE W	4444	5555	6666	35050	31,831.80
COMPTON, SUSAN A	4444	5555	6666	10500	13,841.36
MANNING, WILLIAM Z.	4444	5555	6666	35050	25,099.92
PRESCOTT, KEVIN	4444	5555	6666	10500	16,640.00
ISLEY, JEANETTE J.	4444	5555	6666	20500	16,372.20
ANDERSON, DANIEL M	4444	5555	6666	35050	28,817.52
AYERS, CHESTER D	4444	5555	6666	22200	19,760.00
HANCOCK, STEVEN W.	4444	5555	6666	22200	19,200.00
HAYES, JOHNS A	4444	5555	6666	10500	9,194.12
BLOOM, ALEXANDER	4444	5555	6666	35050	37,999.92
WALSH, THEODORE	4444	5555	6666	35050	27,999.96
MARSH, PAUL J.	4444	5555	6666	22200	15,600.00
WARREN, MICHAEL	4444	5555	6666	35050	25,650.00

# Cyborg Scripting Language Report Customization - Participant Guide

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 2  
 DIVISION 9999 PRODUCTION CTL 1-2 X6AR PERIOD TIME 14:09 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY
HILLERY, THOMAS	4444	5555	6666	22200	19,200.00
WILSON, BARBARA	4444	5555	6666	20500	20,800.00
BYERS, TODD	4444	5555	6666	35050	32,000.00
LITTLE, DORA	4444	5555	6666	35050	24,999.96
ANDERSEN, KARI	4444	5555	6666	22200	15,600.00
WINTER, MICHAEL	4444	5555	6666	35050	25,650.00
MILLER, THOMAS S	4444	5555	6666	22200	19,200.00
*CTRL-FOUR 4444					718,005.40
MEYER, JUNE	4488	5508	6608	35050	25,000.00
MOORE, SAMUEL	4488	5508	6608	68000	60,000.00
MUIR, LINDA	4488	5508	6608	22200	16,000.00
MORSE, GORDAN	4488	5508	6608	10500	20,799.60
MERTZ, LYNNE C.	4488	5508	6608	17857	13,832.00
MORITZ, KATHERINE C.	4488	5508	6608	47214	45,000.00
MAURICE, STACY E.	4488	5508	6608	20500	19,240.00
MOREAU, GARDNER	4488	5508	6608	22200	21,450.00
MOHR, MICHAEL T.	4488	5508	6608	15405	11,089.52
REYNOLDS, BRENDA	4488	5508	6608	35050	18,650.00
BARNES, JOHNSON	4488	5508	6608	68000	87,180.84
CMEYLA, JANE	4488	5508	6608	19260	13,340.00
KWONG, STEVEN S.	4488	5508	6608	10500	19,655.64
BROWN, WILLIAM R.	4488	5508	6608	15405	15,694.90
COSTELLO, SUSANNE	4488	5508	6608	19270	18,280.08
LANNON, PATRICE	4488	5508	6608	47214	44,200.00
HALL, RHONDA D.	4488	5508	6608	20500	1,662,970.40
CREMMINS, ALAN EDWARD	4488	5508	6608	10600	34,016.32
PENDARVIS, MARTIN M.	4488	5508	6608	18020	16,380.00
RUNYON, BRENDA	4488	5508	6608	35050	20,000.00
LAUGHLIN, SANDRA T.	4488	5508	6608	47214	45,000.00
SCHAEFER, JOANNA S.	4488	5508	6608	20500	19,240.00
CARLILE, WILLIAM E.	4488	5508	6608	22200	21,450.00
LLEWELYN, STEVE	4488	5508	6608	21840	24,099.92
SANTANA, LOUISE	4488	5508	6608	30650	18,874.96
STENMAN, SAMANTHA	4488	5508	6608	21830	19,950.00
THEISSEN, LEONARD	4488	5508	6608	30660	30,998.76
HARRIS, CECELIA	4488	5508	6608	20002	15,990.00
CHOU, LO	4488	5508	6608	30670	37,752.00
BISHOP, MARIA	4488	5508	6608	34020	21,578.96
LEWIS, JAMES X.	4488	5508	6608	30660	33,999.84
CORTEZ, MARIA	4488	5508	6608	30650	24,499.92
PATEL, LAURENCE	4488	5508	6608	30670	43,314.96
OWENS, LORNA	4488	5508	6608	30650	26,541.58
ANDERSON, KAREN	4488	5508	6608	30670	39,983.04
RITTER, BRENDA	4488	5508	6608	35050	15,000.00
ANDREWS, HENRY A.	4488	5555	6666	19260	16,900.00
MARGOLIS, DAVID	4488	5555	6666	30650	40,500.00
*CTRL-FOUR 4488					2,678,453.24
*REPORT CODE X6AR					4,086,920.38

## Section 6—exercise 2 answers

### Report extract program

```

P X6BRPT 00000 SECURITY ' '. @ CLASS REPORT EXAMPLE XPP
P X6BRPT 00001 @LAST MODIFIED ON: BY: AUTHOR: USER
P X6BRPT 00002 @PARMS=GRPRSC
P X6BRPT 00100 DEFINE-REPORT ALLOCATE-10.
P X6BRPT 00200 HEADER-1 :52 'HUMAN RESOURCE LISTING'.
P X6BRPT 00300 HEADER-4 :04 'EMPLOYEE NAME'.
P X6BRPT 00400 HEADER-3 :36 'CTRL CTRL CTRL'.
P X6BRPT 00500 HEADER-4 :36 'FOUR FIVE SIX '.
P X6BRPT 00600 HEADER-3 :65 'JOB ANNUAL'.
P X6BRPT 00700 HEADER-4 :65 'CODE SALARY'.
P X6BRPT 00800 P100-INITIALIZE.
P X6BRPT 00900 SET-EMP-PTRS-TO-1ST. SPACE-EXTRACT-RECORD.
P X6BRPT 00910 MOVE :1 TO PERM-01-V0.
P X6BRPT 00950 IF FIRST-TIME-IND NOT EQUAL 'F' GO TO P200-SELECT.
P X6BRPT 01000 IF SPACES EQUAL W6-06-036
P X6BRPT 01100 MOVE CURRENT-DATE-CYYMDD TO SPECIAL-DATE
P X6BRPT 01200 MOVE 'Y' TO FIRST-TIME-IND.
P X6BRPT 01300 P200-SELECT.
P X6BRPT 01410 FIND RESULTING-EMP-STATUS STARTING WITH SPECIAL-DATE. @LZC
P X6BRPT 01420 MATCH-SEGMENT-CODE.
P X6BRPT 01430 IF NOT FOUND RETURN.
P X6BRPT 01440 IF RESULTING-EMP-STATUS NOT EQUAL '0' RETURN.
P X6BRPT 01500 FIND CTRL-FOUR STARTING WITH SPECIAL-DATE. @ LZR
P X6BRPT 01600 MATCH-SEGMENT-CODE.
P X6BRPT 01700 IF NOT FOUND RETURN.
P X6BRPT 01710 MOVE-PLACE-TO-HOLD1. @ LZR
P X6BRPT 01800 IF SPACES EQUALS W6-04-042
P X6BRPT 01900 GO TO P250-CONTINUE.
P X6BRPT 02000 IF W6-04-042 NOT EQUAL CTRL-FOUR RETURN.
P X6BRPT 02010 P250-CONTINUE.
P X6BRPT 02020 FIND JOB-CODE STARTING WITH SPECIAL-DATE. @ LZD
P X6BRPT 02030 MATCH-SEGMENT-CODE.
P X6BRPT 02040 IF NOT-FOUND RETURN.
P X6BRPT 02050 MOVE-PLACE-TO-HOLD2. @ LZD
P X6BRPT 02060 MOVE SPECIAL-DATE TO SALARY-AS-OF-DATE. @ LZF
P X6BRPT 02070 FIND-SALARY-AS-OF.
P X6BRPT 02080 IF NOT-FOUND RETURN.
P X6BRPT 02090 MOVE-PLACE-TO-HOLD3. @ LZF
P X6BRPT 02100 P300-SORT.
P X6BRPT 02200 OUTPUT '1X6BR' FORMS/REPORT-CODE PRINT-GRAND-TOTAL
P X6BRPT 02250 TRIPLE-SPACE-BEFORE
P X6BRPT 02300 CONTROL-1-2 NO-PRINT-SUBTOTAL.
P X6BRPT 02350 RESET-TO-HOLD1-PLACE. @ LZR
P X6BRPT 02400 PRINT CTRL-FOUR NO-PRINT-SUBTOTAL NEW-PAGE-BEFORE.
P X6BRPT 02450 RESET-TO-HOLD2-PLACE. @ LZD

```

## Cyborg Scripting Language Report Customization - Participant Guide

---

```
P X6BRPT 02460 PRINT JOB-CODE DOUBLE-SPACE-BEFORE DOUBLE-SPACE-AFTER
P X6BRPT 02500 SORT-LENGTH-31 EMPLOYEE-NUMBER.
P X6BRPT 02650 P400-OUTPUT.
P X6BRPT 02700 OUTPUT '0' EMPLOYEE-NAME JOB-CODE. @ LZD
P X6BRPT 02800 READ-TA-TABLE.
P X6BRPT 02810 IF STAT-KEY GREATER THAN '01'
P X6BRPT 02820 SPACE-OVER :30
P X6BRPT 02830 ELSE
P X6BRPT 02840 OUTPUT JOB-TITLE.
P X6BRPT 02900 RESET-TO-HOLD1-PLACE. @ LZR
P X6BRPT 03000 PRINT CTRL-FOUR CTRL-FIVE CTRL-SIX.
P X6BRPT 03100 RESET-TO-HOLD3-PLACE. @ LZF
P X6BRPT 03200 OUTPUT ANNUAL-SALARY PERM-01-V0.
P X6BRPT 03700 WRITE-EXTRACT.
P X6BRPT 03800 RETURN.
```

## Section 6—exercise 2 answers, continued

### Print positions layout

SEQ	FIELD	-----PRINT-----		
NBR	NAME	POS.	LINE	TOTAL
10	EMPLOYEE-NAME	001	01	N
15	JOB-CODE	065	01	N
17	JOB-TITLE	103	01	P
20	CTRL-FOUR	036	01	N
30	CTRL-FIVE	046	01	N
40	CTRL-SIX	056	01	N
60	ANNUAL-SALARY	074	01	Y
80	PERM-01-VO	050	01	T

## Section 6—exercise 2 answers, continued

### Report output

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 1  
 DIVISION 9999 PRODUCTION CTL 1-2 X6BR PERIOD TIME 14:30 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
MORRIS, ROBERT	4040	5050	6060	15405	11,086.40	
SWEENEY, BARBARA	4040	5050	6060	15405	10,201.88	
PITARO, JOSEPH C.	4040	5050	6060	15405	8,840.00	
*JOB-CODE	15405			3	30,128.28	SHIPPING/RECEIVING CLERK
JONES, JERRY	4040	5050	6060	17857	13,302.64	
AUSTIN, STEVEN	4040	5050	6060	17857	8,926.84	
JOHNSON, WALTER D	4040	5050	6060	17857	7,881.64	
LYNDEN, ANNETTE C.	4040	5050	6060	17857	8,710.00	
*JOB-CODE	17857			4	38,821.12	ASSEMBLY LINE WORKER
GRANT, KEITH L.	4040	5050	6060	18020	19,760.00	
SHEA, JEFFERY B.	4040	5050	6060	18020	20,696.00	
*JOB-CODE	18020			2	40,456.00	LATHE MACHINE TOOL SETTER
MAGUIRE, HENRY S.	4040	5050	6060	20002	14,099.80	
SULLIVAN, MIKE M.	4040	5050	6060	20002	11,544.00	
*JOB-CODE	20002			2	25,643.80	CLERK/TYPIST, SENIOR
HAMMER, JAMES B.	4040	5050	6060	22200	21,172.84	
SWALTER, STEVEN Y	4040	5050	6060	22200	18,322.20	
SANDERS, STEVEN S	4040	5050	6060	22200	18,322.20	
*JOB-CODE	22200			3	57,817.24	ACCOUNTING CLERK
ADAMS, RICHARD	4040	5050	6060	47214	38,221.44	
COLLINS, ANNA MARIE	4040	5050	6060	47214	39,657.60	
ALSON, GEOFFERY	4040	5050	6060	47214	31,999.92	
GRIFFITH, BERNARD	4040	5050	6060	47214	50,000.08	
TEACHEN, JUDITH	4040	5050	6060	47214	36,720.00	
BETTS, J.T.	4040	5050	6060	47214	32,000.00	
GRIFFITHS, ROBERT	4040	5050	6060	47214	50,000.08	
BEACHEM, JUDITH	4040	5050	6060	47214	34,000.08	
*JOB-CODE	47214			8	312,599.20	PURCHASING MANAGER

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 2  
 DIVISION 9999 PRODUCTION CTL 1-2 X6BR PERIOD TIME 14:30 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
*REPORT CODE	X6BR			22	505,465.64	PURCHASING MANAGER

## Section 7—exercise 1 answers

### Report extract program

```

P X71RPT 00000 SECURITY ' '. @ HUMAN RESOURCE LISTING XPP
P X71RPT 00001 @LAST MODIFIED ON: BY: AUTHOR: USER
P X71RPT 00002 @PARMS=GRPRSC
P X71RPT 00100 DEFINE-REPORT ALLOCATE-10.
P X71RPT 00200 HEADER-1 :52 'HUMAN RESOURCE LISTING'.
P X71RPT 00300 HEADER-3 :37 'CTRL CTRL CTRL JOB'.
P X71RPT 00400 HEADER-4 :01 'EMPLOYEE NAME'.
P X71RPT 00500 HEADER-4 :37 'FOUR FIVE SIX CODE'.
P X71RPT 00600 HEADER-3 :79 'ANNUAL'.
P X71RPT 00700 HEADER-4 :79 'SALARY'.
P X71RPT 00800 P100-INITIALIZE.
P X71RPT 00900 SET-EMP-PTRS-TO-1ST.
P X71RPT 01000 IF FIRST-TIME-IND NOT EQUAL 'F' GO TO P200-SELECT.
P X71RPT 01100 MOVE :1 TO PERM-01-V0.
P X71RPT 01200 IF SPACES EQUAL SPECIAL-DATE
P X71RPT 01300 MOVE CURRENT-DATE-CYYMDD TO SPECIAL-DATE
P X71RPT 01400 MOVE 'Y' TO FIRST-TIME-IND.
P X71RPT 01410 PRINT '1X71R' CONTROL-1-2. SPACE-OVER :20.
P X71RPT 01420 PRINT '.2049CTRL 4: ' W6-04-042 ' As Of: '
P X71RPT 01425 SPECIAL-DATE '..'.
P X71RPT 01430 WRITE-EXTRACT.
P X71RPT 01500 P200-SELECT.
P X71RPT 01700 FIND CTRL-FOUR STARTING WITH SPECIAL-DATE.
P X71RPT 01800 MATCH-SEGMENT-CODE.
P X71RPT 01900 IF NOT FOUND RETURN.
P X71RPT 02000 MOVE-PLACE-TO-HOLD1. @ LOCATION SEGMENT
P X71RPT 02100 IF SPACES EQUALS W6-04-042
P X71RPT 02200 GO TO P210-CHECK-STATUS.
P X71RPT 02300 IF W6-04-042 NOT EQUAL CTRL-FOUR RETURN.
P X71RPT 02400 P210-CHECK-STATUS.
P X71RPT 02500 FIND RESULTING-EMP-STATUS STARTING WITH SPECIAL-DATE.
P X71RPT 02600 MATCH-SEGMENT-CODE.
P X71RPT 02700 IF NOT FOUND RETURN.
P X71RPT 02800 IF RESULTING-EMP-STATUS NOT EQUAL '0' RETURN.
P X71RPT 02900 P220-FIND-JOB-AND-SALARY.
P X71RPT 03000 FIND JOB-CODE STARTING WITH SPECIAL-DATE.
P X71RPT 03100 MATCH-SEGMENT-CODE.
P X71RPT 03200 IF NOT FOUND RETURN.
P X71RPT 03300 MOVE-PLACE-TO-HOLD2. @ JOB SEGMENT
P X71RPT 03400 MOVE SPECIAL-DATE TO SALARY-AS-OF-DATE.
P X71RPT 03500 FIND-SALARY-AS-OF.
P X71RPT 03600 IF NOT FOUND RETURN.
P X71RPT 03700 MOVE-PLACE-TO-HOLD3. @ SALARY SEGMENT.
P X71RPT 03800 P300-SORT.
P X71RPT 03900 PRINT '1X71R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL

```

```
P X71RPT 04000    TRIPLE-SPACE-BEFORE
P X71RPT 04100    CONTROL-1-2 NO-PRINT-SUBTOTAL
P X71RPT 04150    RESET-TO-HOLD1-PLACE.           @ LOCATION SEGMENT
P X71RPT 04200    PRINT CTRL-FOUR NO-PRINT-SUBTOTAL NEW-PAGE-BEFORE
P X71RPT 04300    RESET-TO-HOLD2-PLACE.           @ JOB SEGMENT
P X71RPT 04310    PRINT JOB-CODE DOUBLE-SPACE-BEFORE DOUBLE-SPACE-AFTER
P X71RPT 04400    SORT-LENGTH-31
P X71RPT 04500    EMPLOYEE-NUMBER.
P X71RPT 04600    P400-OUTPUT.
P X71RPT 04700    PRINT '0'.
P X71RPT 04710    MOVE EMPLOYEE-NAME TO LAST-FIRST.
P X71RPT 04720    CALL 'FMLED'T'.
P X71RPT 04730    MOVE PRINT-FIELD TO FIRST-LAST.
P X71RPT 04740    PRINT FIRST-LAST JOB-CODE.
P X71RPT 04800    READ-TA-TABLE. PRINT JOB-TITLE.
P X71RPT 04900    RESET-TO-HOLD1-PLACE.
P X71RPT 05000    PRINT CTRL-FOUR CTRL-FIVE CTRL-SIX.
P X71RPT 05100    RESET-TO-HOLD3-PLACE.
P X71RPT 05200    OUTPUT ANNUAL-SALARY PERM-01-V0.
P X71RPT 05600    WRITE-EXTRACT.
P X71RPT 05700    RETURN.
```

## Section 7—exercise 1 answers, continued

### Purpose

Modify the report created in Section 6.

### Print positions layout

SEQ	FIELD	----PRINT----		
NBR	NAME	POS.	LINE	TOTAL
10	EMPLOYEE-NAME	001	01	N
15	JOB-CODE	065	01	N
17	JOB-TITLE	103	01	P
20	CTRL-FOUR	036	01	N
30	CTRL-FIVE	046	01	N
40	CTRL-SIX	056	01	N
60	ANNUAL-SALARY	074	01	Y
80	PERM-01-VO	050	01	T

# Section 7—exercise 1 answers, continued

## Report output

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 1  
 DIVISION 9999 PRODUCTION CTL 1-2 CTRL 4: 4040 As Of: 08-30-2002 X71R PERIOD TIME 14:53 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
ROBERT MORRIS	4040	5050	6060	15405	11,086.40	
BARBARA SWEENEY	4040	5050	6060	15405	10,201.88	
JOSEPH C. PITARO	4040	5050	6060	15405	8,840.00	
*JOB-CODE	15405				30,128.28	SHIPPING/RECEIVING CLERK
JERRY JONES	4040	5050	6060	17857	13,302.64	
STEVEN AUSTIN	4040	5050	6060	17857	8,926.84	
WALTER D JOHNSON	4040	5050	6060	17857	7,881.64	
ANNETTE C. LYNDEN	4040	5050	6060	17857	8,710.00	
*JOB-CODE	17857				38,821.12	ASSEMBLY LINE WORKER
KEITH L. GRANT	4040	5050	6060	18020	19,760.00	
JEFFERY B. SHEA	4040	5050	6060	18020	20,696.00	
*JOB-CODE	18020				40,456.00	LATHE MACHINE TOOL SETTER
HENRY S. MAGUIRE	4040	5050	6060	20002	14,099.80	
MIKE M. SULLIVAN	4040	5050	6060	20002	11,544.00	
*JOB-CODE	20002				25,643.80	CLERK/TYPIST, SENIOR
JAMES B. HAMMER	4040	5050	6060	22200	21,172.84	
STEVEN Y SWALTER	4040	5050	6060	22200	18,322.20	
STEVEN S SANDERS	4040	5050	6060	22200	18,322.20	
*JOB-CODE	22200				57,817.24	ACCOUNTING CLERK
RICHARD ADAMS	4040	5050	6060	47214	38,221.44	
ANNA MARIE COLLINS	4040	5050	6060	47214	39,657.60	
GEOFFERY ALSON	4040	5050	6060	47214	31,999.92	
BERNARD GRIFFITH	4040	5050	6060	47214	50,000.08	
JUDITH TEACHEN	4040	5050	6060	47214	36,720.00	
J. T. BETTS	4040	5050	6060	47214	32,000.00	
ROBERT GRIFFITHS	4040	5050	6060	47214	50,000.08	
JUDITH BEACHEM	4040	5050	6060	47214	34,000.08	
*JOB-CODE	47214				312,599.20	PURCHASING MANAGER

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 2  
 DIVISION 9999 PRODUCTION CTL 1-2 CTRL 4: 4040 As Of: 08-30-2002 X71R PERIOD TIME 14:53 DATE 08-30-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
*REPORT CODE	X71R				505,465.64	PURCHASING MANAGER

## Section 7—exercise 2 answers

### Purpose

Create a summary report from the report coded in section 7, exercise 1.

### Report extract program

```

P X72RPT 00000 SECURITY ' '. @ HUMAN RESOURCE LISTING XPP
P X72RPT 00001 @LAST MODIFIED ON: BY: AUTHOR: USER
P X72RPT 00002 @PARMS=GRPRSC
P X72RPT 00100 DEFINE-REPORT ALLOCATE-10 TOTALS-ONLY.
P X72RPT 00200 HEADER-1 :52 'HUMAN RESOURCE LISTING'.
P X72RPT 00300 HEADER-3 :37 'CTRL CTRL CTRL JOB'.
P X72RPT 00400 HEADER-4 :01 'EMPLOYEE NAME'.
P X72RPT 00500 HEADER-4 :37 'FOUR FIVE SIX CODE'.
P X72RPT 00600 HEADER-3 :79 'ANNUAL'.
P X72RPT 00700 HEADER-4 :79 'SALARY'.
P X72RPT 00800 P100-INITIALIZE.
P X72RPT 00900 SET-EMP-PTRS-TO-1ST.
P X72RPT 01000 IF FIRST-TIME-IND NOT EQUAL 'F' GO TO P200-SELECT.
P X72RPT 01100 MOVE :1 TO PERM-01-V0.
P X72RPT 01200 IF SPACES EQUAL SPECIAL-DATE
P X72RPT 01300 MOVE CURRENT-DATE-CYYMDD TO SPECIAL-DATE
P X72RPT 01400 MOVE 'Y' TO FIRST-TIME-IND.
P X72RPT 01410 PRINT '1X72R' CONTROL-1-2. SPACE-OVER :20.
P X72RPT 01420 PRINT '.2049CTRL 4: ' W6-04-042 ' As Of: '
P X72RPT 01425 SPECIAL-DATE '...'.
P X72RPT 01430 WRITE-EXTRACT.
P X72RPT 01500 P200-SELECT.
P X72RPT 01700 FIND CTRL-FOUR STARTING WITH SPECIAL-DATE.
P X72RPT 01800 MATCH-SEGMENT-CODE.
P X72RPT 01900 IF NOT FOUND RETURN.
P X72RPT 02000 MOVE-PLACE-TO-HOLD1. @ LOCATION SEGMENT
P X72RPT 02100 IF SPACES EQUALS W6-04-042
P X72RPT 02200 GO TO P210-CHECK-STATUS.
P X72RPT 02300 IF W6-04-042 NOT EQUAL CTRL-FOUR RETURN.
P X72RPT 02400 P210-CHECK-STATUS.
P X72RPT 02500 FIND RESULTING-EMP-STATUS STARTING WITH SPECIAL-DATE.
P X72RPT 02600 MATCH-SEGMENT-CODE.
P X72RPT 02700 IF NOT FOUND RETURN.
P X72RPT 02800 IF RESULTING-EMP-STATUS NOT EQUAL '0' RETURN.
P X72RPT 02900 P220-FIND-JOB-AND-SALARY.
P X72RPT 03000 FIND JOB-CODE STARTING WITH SPECIAL-DATE.
P X72RPT 03100 MATCH-SEGMENT-CODE.
P X72RPT 03200 IF NOT FOUND RETURN.
P X72RPT 03300 MOVE-PLACE-TO-HOLD2. @ JOB SEGMENT
P X72RPT 03400 MOVE SPECIAL-DATE TO SALARY-AS-OF-DATE.
P X72RPT 03500 FIND-SALARY-AS-OF.
P X72RPT 03600 IF NOT FOUND RETURN.
P X72RPT 03700 MOVE-PLACE-TO-HOLD3. @ SALARY SEGMENT.

```

## Cyborg Scripting Language Report Customization - Participant Guide

---

```
P X72RPT 03800 P300-SORT.
P X72RPT 03900 PRINT '1X72R' FORMS/REPORT-CODE PRINT-GRAND-TOTAL
P X72RPT 04000 TRIPLE-SPACE-BEFORE
P X72RPT 04100 CONTROL-1-2 NO-PRINT-SUBTOTAL
P X72RPT 04150 RESET-TO-HOLD1-PLACE. @ LOCATION SEGMENT
P X72RPT 04200 PRINT CTRL-FOUR NO-PRINT-SUBTOTAL NEW-PAGE-BEFORE
P X72RPT 04300 RESET-TO-HOLD2-PLACE. @ JOB SEGMENT
P X72RPT 04310 PRINT JOB-CODE DOUBLE-SPACE-BEFORE DOUBLE-SPACE-AFTER
P X72RPT 04400 SORT-LENGTH-31
P X72RPT 04500 EMPLOYEE-NUMBER.
P X72RPT 04600 P400-OUTPUT.
P X72RPT 04700 PRINT '0'.
P X72RPT 04710 MOVE EMPLOYEE-NAME TO LAST-FIRST.
P X72RPT 04720 CALL 'FMLED'T'.
P X72RPT 04730 MOVE PRINT-FIELD TO FIRST-LAST.
P X72RPT 04740 PRINT FIRST-LAST JOB-CODE.
P X72RPT 04800 READ-TA-TABLE. PRINT JOB-TITLE.
P X72RPT 04900 RESET-TO-HOLD1-PLACE.
P X72RPT 05000 PRINT CTRL-FOUR CTRL-FIVE CTRL-SIX.
P X72RPT 05100 RESET-TO-HOLD3-PLACE.
P X72RPT 05200 OUTPUT ANNUAL-SALARY PERM-01-V0.
P X72RPT 05600 WRITE-EXTRACT.
P X72RPT 05700 RETURN.
```

## Section 7—exercise 2 answers, continued

### Print positions layout

SEQ	FIELD	-----PRINT-----		
NBR	NAME	POS.	LINE	TOTAL
10	EMPLOYEE-NAME	001	01	N
15	JOB-CODE	065	01	N
17	JOB-TITLE	103	01	P
20	CTRL-FOUR	036	01	N
30	CTRL-FIVE	046	01	N
40	CTRL-SIX	056	01	N
60	ANNUAL-SALARY	074	01	Y
80	PERM-01-VO	050	01	T

## Section 7—exercise 2 answers, continued

### Report output

CORPORATION 99 ACME MANUFACTURING  
 DIVISION 9999 PRODUCTION CTL 1-2

HUMAN RESOURCE LISTING  
 CTRL 4: As Of:07-15-2002

REPT PERIOD  
 X72R PERIOD

FILE VERSION 00 PAGE 1  
 TIME 14:59:09 DATE 12-28-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
*JOB-CODE 15405			3		30,128.28	SHIPPING/RECEIVING CLERK
*JOB-CODE 17857			4		38,821.12	ASSEMBLY LINE WORKER
*JOB-CODE 18020			2		40,456.00	LATHE MACHINE TOOL SETTER
*JOB-CODE 20002			2		25,643.80	CLERK/TYPIST, SENIOR
*JOB-CODE 22200			3		57,817.24	ACCOUNTING CLERK
*JOB-CODE 47214			8		312,599.20	PURCHASING MANAGER

CORPORATION 99 ACME MANUFACTURING  
 DIVISION 9999 PRODUCTION CTL 1-2

HUMAN RESOURCE LISTING  
 CTRL 4: As Of:07-15-2002

REPT PERIOD  
 X72R PERIOD

FILE VERSION 00 PAGE 2  
 TIME 14:59:09 DATE 12-28-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY	
*JOB-CODE 10500			3		39,675.48	MAINTENANCE ENGINEER, SENIOR
*JOB-CODE 10650			1		27,508.00	ASSISTANT STATIONARY ENGINEER
*JOB-CODE 18020			1		16,634.80	LATHE MACHINE TOOL SETTER
*JOB-CODE 18030			1		19,760.00	GRINDING MACHINE SETTER
*JOB-CODE 18040			1		21,619.00	DRILLING MACHINE TOOL OPERATOR
*JOB-CODE 20002			1		17,246.84	CLERK/TYPIST, SENIOR
*JOB-CODE 22200			5		88,800.00	ACCOUNTING CLERK
*JOB-CODE 35050			10		299,949.08	ACCOUNTANT, CLASS II

Appendix A: Exercise Answers

CORPORATION	99	ACME MANUFACTURING	HUMAN RESOURCE LISTING				REPT PERIOD	FILE VERSION 00	PAGE	3
DIVISION	9999	PRODUCTION CTL 1-2	CTRL 4:	As Of:07-15-2002	X72R	PERIOD	TIME 14:59:09	DATE	12-28-2002	
EMPLOYEE NAME			CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY			
*JOB-CODE	10500				2		40,455.24		MAINTENANCE ENGINEER, SENIOR	
*JOB-CODE	10600				1		34,016.32		STATIONARY ENGINEER	
*JOB-CODE	15405				2		26,784.42		SHIPPING/RECEIVING CLERK	
*JOB-CODE	17857				1		13,832.00		ASSEMBLY LINE WORKER	
*JOB-CODE	18020				1		16,380.00		LATHE MACHINE TOOL SETTER	
*JOB-CODE	19260				2		30,240.00		BILLING CLERK	
*JOB-CODE	19270				1		18,280.08		TRAFFIC CLERK	
*JOB-CODE	20002				1		15,990.00		CLERK/TYPIST, SENIOR	
*JOB-CODE	20500				3		1,701,450.40		SECRETARY, SENIOR	
*JOB-CODE	21830				1		19,950.00		LABOR RELATIONS SPECIALIST	
*JOB-CODE	21840				1		24,099.92		OCUPATIONL SAF & HLTH INSPECTR	
*JOB-CODE	22200				2		42,900.00		ACCOUNTING CLERK	
*JOB-CODE	30650				2		59,374.96		FIELD CONTACT TECHNICIAN	
*JOB-CODE	30660				1		30,998.76		WHOLESALE TRADE SALES WORKER	
*JOB-CODE	30670				1		37,752.00		SALES ENGINEER	
*JOB-CODE	34020				1		21,578.96		SUPERVISOR, GENERAL OFFICE	
*JOB-CODE	35050				4		78,650.00		ACCOUNTANT, CLASS II	
*JOB-CODE	47214				3		134,200.00		PURCHASING MANAGER	
*JOB-CODE	68000				2		147,180.84		EXECUTIVE SALES MGR	

CORPORATION	99	ACME MANUFACTURING	HUMAN RESOURCE LISTING				REPT PERIOD	FILE VERSION 00	PAGE	4
DIVISION	9999	PRODUCTION CTL 1-2	CTRL 4:	As Of:07-15-2002	X72R	PERIOD	TIME 14:59:09	DATE	12-28-2002	
EMPLOYEE NAME			CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY			
*REPORT CODE	X72R				77		3,530,772.74		EXECUTIVE SALES MGR	

## Section 8—exercise answers

### Special print options program

```
P X71RP 00000 SECURITY 'SC'. @ SAMPLE FOR PRINT PROGRAM NSC
P X71RP 00001 @LAST MODIFIED ON: 03-15-94 BY: USER AUTHOR: XXX
P X71RP 00030 PRINT-REPORT.
P X71RP 10000 P100-P. @ REPORT IS STARTING
P X71RP 11999 EXIT.
P X71RP 12000 P120-P. @ NEW C1-2 IS STARTING
P X71RP 13999 EXIT.
P X71RP 14000 P140-P. @ HEADER RECORDS ARE READY TO PRINT
P X71RP 14001 @ SET SCREEN TO :1241 FOR START OF HEADER LINE 1
P X71RP 14002 @ :1373 - LINE 2, :1505 - LINE 3, :1637 - LINE 4
P X71RP 14005 @ DO NOT ALTER W8-40-160.
P X71RP 15999 EXIT.
P X71RP 16000 P160-P. @ HEADER RECORDS HAVE BEEN PRINTED
P X71RP 16010 MOVE SPACES TO W6-35-100COMPARE.
P X71RP 17999 EXIT.
P X71RP 18000 P180-P. @ DETAIL DATA WITH MATCHING SORT KEYS IS READY
P X71RP 18001 @ USE RECORD-TYPE:FIELD-NAME TO ACCESS DATA
P X71RP 18002 @ FROM THE EXTRACT RECORDS.
P X71RP 18005 @ W8-40-160 MAY BE ALTERED.
P X71RP 19999 EXIT.
P X71RP 20000 P200-P. @ TOTAL DATA IS READY
P X71RP 20001 @ USE RECORD-TYPE:FIELD-NAME TO ACCESS DATA
P X71RP 20002 @ IN THE TOTAL COUNTERS.
P X71RP 20005 @ DO NOT ALTER W8-40-160.
P X71RP 20006 @ W8-15-164 CONTAINS NAME OF LEVEL BREAK.
P X71RP 21999 EXIT.
P X71RP 22000 P220-P. @ A DETAIL LINE IS READY TO PRINT - POINTER 8
P X71RP 22002 IF W8-35-035 EQUAL W6-35-100
P X71RP 22004 MOVE SPACES TO W8-35-035COMPARE
P X71RP 22006 ELSE MOVE W8-35-035PRINT TO W6-35-100COMPARE.
P X71RP 23999 EXIT.
P X71RP 24000 P240-P. @ A TOTAL LINE IS READY TO PRINT - POINTER 8
P X71RP 24006 MOVE 'TOTALS:' TO W8-07-035.
P X71RP 24008 IF W8-07-000 EQUALS '*REPORT'
P X71RP 24012 MOVE SPACES TO W8-30-102JOBTITLE
P X71RP 24014 MOVE 'GRAND TOTAL SALARY ' TO W8-21-000.
P X71RP 25999 EXIT.
P X71RP 26000 P260-P. @ THE REPORT IS DONE
P X71RP 27999 EXIT.
```

## Section 8—exercise answers, continued

## Report output

CORPORATION 99 ACME MANUFACTURING		HUMAN RESOURCE LISTING				REPT PERIOD	FILE VERSION 00	PAGE 1
DIVISION 9999 PRODUCTION CTL 1-2		CTRL 4:		As Of:07-15-2002		X71R PERIOD	TIME 14:04:26 DATE 03-15-2002	
EMPLOYEE NAME		CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY		
ROBERT MORRIS		4040	5050	6060	15405	11,086.40		
BARBARA SWEENEY						10,201.88		
JOSEPH C. PITARO						8,840.00		
*JOB-CODE	15405	TOTALS:		3		30,128.28	SHIPPING/RECEIVING EX8RK	
JERRY JONES		4040	5050	6060	17857	13,302.64		
STEVEN AUSTIN						8,926.84		
WALTER D JOHNSON						7,881.64		
ANNETTE C. LYNDEN						8,710.00		
*JOB-CODE	17857	TOTALS:		4		38,821.12	ASSEMBLY LINE WORKER	
KEITH L. GRANT		4040	5050	6060	18020	19,760.00		
JEFFERY B. SHEA						20,696.00		
*JOB-CODE	18020	TOTALS:		2		40,456.00	LATHE MACHINE TOOL SETTER	
HENRY S. MAGUIRE		4040	5050	6060	20002	14,099.80		
MIKE M. SULLIVAN						11,544.00		
*JOB-CODE	20002	TOTALS:		2		25,643.80	EX8RK/TYPIST, SENIOR	
JAMES B. HAMMER		4040	5050	6060	22200	21,172.84		
STEVEN Y SWALTER						18,322.20		
STEVEN S SANDERS						18,322.20		
*JOB-CODE	22200	TOTALS:		3		57,817.24	ACCOUNTING EX8RK	
RICHARD ADAMS		4040	5050	6060	47214	38,221.44		
ANNA MARIE COLLINS						39,657.60		
GEOFFERY ALSON						31,999.92		
BERNARD GRIFFITH						50,000.08		
JUDITH TEACHEN						36,720.00		
J. T. BETTS						32,000.00		
ROBERT GRIFFITHS						50,000.08		
JUDITH BEACHEM						34,000.08		
*JOB-CODE	47214	TOTALS:		8		312,599.20	PURCHASING MANAGER	

# Cyborg Scripting Language Report Customization - Participant Guide

CORPORATION 99 ACME MANUFACTURING		HUMAN RESOURCE LISTING				REPT PERIOD	FILE VERSION 00	PAGE 2
DIVISION 9999 PRODUCTION CTL 1-2		CTRL 4: As Of:07-15-2002				X71R PERIOD	TIME 14:04:26	DATE 03-15-2002
EMPLOYEE NAME		CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY		
SUSAN A COMPTON		4444	5555	6666	10500	13,841.36		
KEVIN PRESCOTT						16,640.00		
JOHNS A HAYES						9,194.12		
*JOB-CODE	10500	TOTALS:			3	39,675.48		MAINTENANCE ENGINEER, SENIOR
CHESTERON WARD		4444	5508	6608	10650	27,508.00		
*JOB-CODE	10650	TOTALS:			1	27,508.00		ASSISTANT STATIONARY ENGINEER
WILLIAM M. SPENSER		4444	5508	6608	18020	16,634.80		
*JOB-CODE	18020	TOTALS:			1	16,634.80		LATHE MACHINE TOOL SETTER
GARY D. WENDT		4444	5508	6608	18030	19,760.00		
*JOB-CODE	18030	TOTALS:			1	19,760.00		GRINDING MACHINE SETTER
JEFFREY C. DANIELS		4444	5508	6608	18040	21,619.00		
*JOB-CODE	18040	TOTALS:			1	21,619.00		DRILLING MACHINE TOOL OPERATOR
ALICE A BALDWIN		4444	5555	6666	20002	17,246.84		
*JOB-CODE	20002	TOTALS:			1	17,246.84		EX8RK/TYPIST, SENIOR
STEVEN W. HANCOCK		4444	5555	6666	22200	19,200.00		
PAUL J. MARSH						15,600.00		
THOMAS HILLERY						19,200.00		
KARI ANDERSEN						15,600.00		
THOMAS S MILLER						19,200.00		
*JOB-CODE	22200	TOTALS:			5	88,800.00		ACCOUNTING EX8RK
JONATHAN BARTHOLOW III		4444	5555	6666	35050	39,900.00		
GEORGE W WELKER						31,831.80		
WILLIAM Z. MANNING						25,099.92		
DANIEL M ANDERSON						28,817.52		
ALEXANDER BLOOM						37,999.92		
THEODORE WALSH						27,999.96		
MICHAEL WARREN						25,650.00		
TODD BYERS						32,000.00		
DORA LITTLE						24,999.96		
MICHAEL WINTER						25,650.00		
*JOB-CODE	35050	TOTALS:			10	299,949.08		ACCOUNTANT, CLASS II

Appendix A: Exercise Answers

CORPORATION 99 ACME MANUFACTURING		HUMAN RESOURCE LISTING				REPT PERIOD	FILE VERSION 00	PAGE 3
DIVISION 9999 PRODUCTION CTL 1-2		CTRL 4: As Of:07-15-2002				X71R PERIOD	TIME 14:04:26	DATE 03-15-2002
EMPLOYEE NAME		CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY		
GORDAN MORSE		4488	5508	6608	10500	20,799.60		
STEVEN S. KWONG						19,655.64		
*JOB-CODE	10500	TOTALS:			2	40,455.24		MAINTENANCE ENGINEER, SENIOR
ALAN EDWARD CREMMINS		4488	5508	6608	10600	34,016.32		
*JOB-CODE	10600	TOTALS:			1	34,016.32		STATIONARY ENGINEER
MICHAEL T. MOHR		4488	5508	6608	15405	11,089.52		
WILLIAM R BROWN						15,694.90		
*JOB-CODE	15405	TOTALS:			2	26,784.42		SHIPPING/RECEIVING EX8RK
LYNNE C. MERTZ		4488	5508	6608	17857	13,832.00		
*JOB-CODE	17857	TOTALS:			1	13,832.00		ASSEMBLY LINE WORKER
MARTIN M. PENDARVIS		4488	5508	6608	18020	16,380.00		
*JOB-CODE	18020	TOTALS:			1	16,380.00		LATHE MACHINE TOOL SETTER
JANE CMEYLA		4488	5508	6608	19260	13,340.00		
HENRY A. ANDREWS		4488	5555	6666	19260	16,900.00		
*JOB-CODE	19260	TOTALS:			2	30,240.00		BILLING EX8RK
SUSANNE COSTELLO		4488	5508	6608	19270	18,280.08		
*JOB-CODE	19270	TOTALS:			1	18,280.08		TRAFFIC EX8RK
CECELIA HARRIS		4488	5508	6608	20002	15,990.00		
*JOB-CODE	20002	TOTALS:			1	15,990.00		EX8RK/TYPIST, SENIOR
STACY E. MAURICE		4488	5508	6608	20500	19,240.00		
RHONDA D. HALL						1,662,970.40		
JOANNA S. SCHAEFER						19,240.00		
*JOB-CODE	20500	TOTALS:			3	1,701,450.40		SECRETARY, SENIOR
SAMANTHA STENMAN		4488	5508	6608	21830	19,950.00		
*JOB-CODE	21830	TOTALS:			1	19,950.00		LABOR RELATIONS SPECIALIST
STEVE LLEWELYN		4488	5508	6608	21840	24,099.92		
*JOB-CODE	21840	TOTALS:			1	24,099.92		OCUPATIONL SAF & HLTH INSPECTR
GARDNER MOREAU		4488	5508	6608	22200	21,450.00		
WILLIAM E. CARLILE						21,450.00		
*JOB-CODE	22200	TOTALS:			2	42,900.00		ACCOUNTING EX8RK

# Cyborg Scripting Language Report Customization - Participant Guide

CORPORATION 99 ACME MANUFACTURING HUMAN RESOURCE LISTING REPT PERIOD FILE VERSION 00 PAGE 4  
 DIVISION 9999 PRODUCTION CTL 1-2 CTRL 4: As Of:07-15-2002 X71R PERIOD TIME 14:04:26 DATE 03-15-2002

EMPLOYEE NAME	CTRL	CTRL	CTRL	JOB	ANNUAL	
	FOUR	FIVE	SIX	CODE	SALARY	
DAVID MARGOLIS	4488	5555	6666	30650	40,500.00	
LOUISE SANTANA	4488	5508	6608	30650	18,874.96	
*JOB-CODE 30650	TOTALS:		2		59,374.96	FIELD CONTACT TECHNICIAN
LEONARD THEISSEN	4488	5508	6608	30660	30,998.76	
*JOB-CODE 30660	TOTALS:		1		30,998.76	WHOLESALE TRADE SALES WORKER
LO CHOU	4488	5508	6608	30670	37,752.00	
*JOB-CODE 30670	TOTALS:		1		37,752.00	SALES ENGINEER
MARIA BISHOP	4488	5508	6608	34020	21,578.96	
*JOB-CODE 34020	TOTALS:		1		21,578.96	SUPERVISOR, GENERAL OFFICE
JUNE MEYER	4488	5508	6608	35050	25,000.00	
BRENDA REYNOLDS					18,650.00	
BRENDA RUNYON					20,000.00	
BRENDA RITTER					15,000.00	
*JOB-CODE 35050	TOTALS:		4		78,650.00	ACCOUNTANT, CLASS II
KATHERINE C. MORITZ	4488	5508	6608	47214	45,000.00	
PATRICE LANNON					44,200.00	
SANDRA T. LAUGHLIN					45,000.00	
*JOB-CODE 47214	TOTALS:		3		134,200.00	PURCHASING MANAGER
SAMUEL MOORE	4488	5508	6608	68000	60,000.00	
JOHNSON BARNES					87,180.84	
*JOB-CODE 68000	TOTALS:		2		147,180.84	EXECUTIVE SALES MGR

---

**Appendix A: Exercise Answers**

CORPORATION 99 ACME MANUFACTURING  
DIVISION 9999 PRODUCTION CTL 1-2

HUMAN RESOURCE LISTING REPT PERIOD  
CTRL 4: As Of:07-15-2002 X71R PERIOD

FILE VERSION 00 PAGE 5  
TIME 14:04:26 DATE 03-15-2002

EMPLOYEE NAME	CTRL FOUR	CTRL FIVE	CTRL SIX	JOB CODE	ANNUAL SALARY
GRAND TOTAL SALARY	TOTALS:		77		3,530,772.74

---

**NOTES**

---

## Appendix B: Extract Record Files

---

### Table of Contents

Extract record files.....	301
Extract record files—the report extract step.....	301
Extract record files—special print options technique 1.....	303
Extract record files—special print options technique 2.....	305

# Creating an Extract Only Report

**Example:**

```
DEFINE-REPORT.  
SET-EMP-PTRS-TO-FIRST.  
P100-DUMMY-SORT.  
  PRINT '1ANYR' FORMS/REPORT-CODE NO-PRINT-GRAND-TOTAL  
    SORT-LENGTH-11 CONTROL-1-2.  
  
SET SCREEN TO :1601. @ FIRST POSITION OF EXTRACT  
OUTPUT EMPLOYEE-NUMBER EMPLOYEE-NAME HED-AMOUNT-YTD.  
WRITE-EXTRACT.
```

---

## NOTES

---

## Extract record files

### Extract techniques

A report extract program can be executed for the sole purpose of the FILE15 extract file. The first example that follows is a simple extract program for that purpose. Other guidelines are included in the explanation.

The second example is from the RP program execution. The FILE10 extract is created in the report run in addition to the formatted FILE03 report file. The P180-P. paragraph logic is executed to create the output file. Two different techniques are given.

## Extract record files—the report extract step

### Extract data

It may be desirable to execute the report process for the sole purpose of creating a 150-position extract file (FILE15).

### Benefits

Using the report process to extract data incorporates some automatic features. Recall that an extract report program executed by the REPORT utility has the following capabilities:

- Automatic read of the Company and Employee records
- Labor/History records can be read
- Control File data can be extracted
- Run-Time parameters are operable/Company scheduling is viable

### Technique

To accomplish this, it is necessary to:

- Execute only the REPORT portion of the report process (exclude the Sort and RTPRNT steps).
- Exclude RTEDIT record setup (not required).
- Create a 'Dummy' sort key.

### Dummy sort

To satisfy the report rules, a sort key must be defined for the report. However, the sort key area may be reused for extracting data using the SET verb. The **SET SCREEN TO :1601** statement sets the output position pointer to the first position of the extract area. Any PRINT, OUTPUT or SPACE-OVER commands overlay the (sort key) data in the work area.

# Special Print Options Extract using Pointer 8

```
14L-R999999234-33-9001 A0BLOOM, ALEXANDER          9999993333444455556666003799992
14L-R999999234-33-9001 B 086673000000000234-33-9001 380216780312780312221J2001M201
14L-R999999234-33-9001 C 221J20219L31219L31219L31                01
14L-R999999300-31-3031 A0EMORY, TODD              99999933334444550866080040040001
14L-R999999300-31-3031 B 080002000000000300-31-3031 490920730615730615226G1701M201
14L-R999999300-31-3031 C 226G17224L31224L31224L31                01
14L-R999999300-32-3032 A0OWENS, LORNA            99999933884488550866080026541581
14L-R999999300-32-3032 B 086672000000000300-32-3032 360331810122810122218L1001F201
14L-R999999300-32-3032 C 218L10216L31216L31216L31                01
```

**Example:** P180-P. @ DETAIL DATA WITH MATCHING SORT KEYS IS READY.  
INITIAL-80.  
MOVE A:EMPLOYEE-NAME-25 TO W8-25-000.  
MOVE A:CONTROL-1-2 TO W8-06-025.  
MOVE A:CTRL-THREE TO W8-04-031.  
MOVE A:CTRL-FOUR TO W8-04-035.  
MOVE A:CTRL-FIVE TO W8-04-039.  
MOVE A:CTRL-SIX TO W8-04-043.  
MOVE A:ANNUAL-SALARY TO W8-09-047.  
WRITE FILE10.  
INITIAL-80.  
@Move B:FIELDS to Pointer 8.  
WRITE FILE10.  
INITIAL-80.  
@Move C:FIELDS to Pointer 8.  
WRITE FILE10.  
EXIT.

---

## NOTES

---

## Extract record files—special print options technique 1

### Technique 1

The **P180-DETAIL-DATA-READY** paragraph is the only entry point needed for this purpose. Recall that the **4L-RPT** program was used in a previous example for multiple extracts per employee. The extract file has records **A**, **B**, and **C**.

### Example

- Data in this paragraph is accessed by Record Type; Fieldname, for example, **A:EMPLOYEE-NAME**.
- The record is built in the W8 I/O area (000-079). When written to FILE10, it is written from here.
- The W8 Area is initialized first with INITIAL-80.
- The extract fields are moved to the output area by length and displacement. Unused positions have initially been spaced.
- The output file is written to. Then, the second (B:) and third (C:) records are initialized, built and written.

# Special Print Options Extract using Pointer 11 (SCREEN)

## WRITE-FILE10

**Example:** P180-P. @ DETAIL DATA WITH MATCHING SORT KEYS IS  
READY.  
SPACE-EXTRACT-RECORD.  
OUTPUT A:EMPLOYEE-NAME-25.  
OUTPUT A:CONTROL-1-2.  
OUTPUT A:CTRL-THREE.  
OUTPUT A:CTRL-FOUR.  
OUTPUT A:CTRL-FIVE.  
OUTPUT A:CTRL-SIX.  
OUTPUT A:ANNUAL-SALARY.  
WRITE-FILE10.  
@OUTPUT B:FIELDS.  
WRITE-FILE10.  
@OUTPUT C:FIELDS.  
WRITE-FILE10.  
EXIT.

---

## NOTES

---

## Extract record files—special print options technique 2

### Technique 2

An alternate method for building an extract file is to build the extract data in the SCREEN area. This method is easier since there is no need to keep track of the individual move lengths.

- Clear SCREEN starting at :1601 and set the SCREEN pointer to :1601.
- Use OUTPUT or PRINT verbs to move the extract record data to the SCREEN area. These verbs provide automatic placement in the output work area.
- Issue a WRITE-FILE10 verb to create the file. WRITE-FILE10 moves data from the SCREEN area to the first 80 characters of the I/O Pointer 8 area and then writes it to FILE10.
- The SCREEN pointer need only be initialized once. The WRITE-FILE10 macro verb sets SCREEN to :1601 after the WRITE operation.
- SPACE-EXTRACT initializes the SCREEN Work area if 150 or less characters are moved for the output record.
- WRITE-EXTRACT can optionally be used to write 150 characters to FILE15.

---

**NOTES**

---

## Appendix C: Segment Layout Reports

---

### Table of Contents

E segment—basic employee .....	308
F segment—employee name and address .....	316
G segment—employee labor splits .....	318
H segment—employee HEDs .....	319
H segment—employee HEDs (labor/history record).....	322
J segment—employee taxes .....	323
J segment—employee taxes (history record).....	330
L segment—employee spouse/dependent.....	331
L segment—employment status.....	332
L segment—employee salary.....	334
L segment—employee location .....	335
L segment—employee training.....	336

## E segment—basic employee

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
=====				
1 3	NEW-EMPL-LENGTH	003	9(006)	
4 20	EMPLOYEE-MATCH	017	X(017)	
11 20	EMP-NO	010	X(010)	
21 22	H/L-MULTIPLE-NUMBER	002	X(002)	
23 25	NEW-EMPL-SEGMENT	003	X(003)	

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
4 9	NEW-EMPL-C1-2	006	X(006)	
10 10	FILLER	001	X(001)	
11 20	APPLICANT-NUMBER	010	X(010)	
11 20	EMPLOYEE-NUMBER	010	X(010)	
11 16	EMPLOYEE-NO	006	X(006)	

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	E
2 3	SEGMENT-CODE	002	X(002)	E
4 63	E-SEGMENT	060	X(060)	J
4 15	SOCIAL-INSURANCE-NBR	012	X(012)	
4 15	SOCIAL-SECURITY-NBR	012	X(012)	J
4 6	SIN-1-3	003	9(003)	J
4 6	SSN-1-3	003	9(003)	J
7 7	SIN-4	001	X(001)	J
7 7	SSN-4	001	X(001)	J
8 10	SIN-5-7	003	9(003)	J
8 9	SSN-5-6	002	9(002)	J
10 10	SSN-7	001	X(001)	J
11 14	SSN-8-11	004	9(004)	J
11 11	SIN-8	001	X(001)	J
12 14	SIN-9-11	003	9(003)	J
15 15	FILLER	001	X(001)	
16 16	PAY-FREQUENCY-CODE	001	X(001)	PP29
16 16	PAY-FREQUENCY	001	X(015)	D PP29
17 17	PAYMENT-CODE	001	9(001)	PP40
17 17	PAYMENT-TYPE	001	X(020)	D PP40
18 19	STATUS-CODE	002	X(002)	J PE049
18 19	STATUS	002	X(020)	D PE049
20 20	SEX-CODE	001	X(001)	PP41
21 22	LANGUAGE-PREFERENCE	002	X(002)	PP54
21 22	RACE-CODE	002	X(002)	HR22
21 22	EEO-RACE	002	X(015)	D HR22
23 27	UNION-CODE	005	X(005)	PP429
23 27	UNION	005	X(015)	D PP429
23 25	APPLICANT-SOURCE-CD	003	9(003)	HR21

## E segment—basic employee, continued

## SEGMENT LAYOUT REPORT

## E SEGMENT

## BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
23 25	ET-APPLICANT-SOURCE	003	9(003)	HR21
23 25	APPLICANT-SOURCE	003	9(020)	D HR21
26 27	AGENCY-ID	002	X(002)	PP589
26 27	ET-AGENCY-ID	002	X(002)	PP589
26 27	AGENCY-NAME	002	X(020)	D PP589
28 32	WORKERS-COMP-CODE	005	X(005)	
33 38	BIRTH-DATE	006	9(006)	
33 36	BIRTH-YY-MM	004	9(004)	
33 34	BIRTH-YEAR	002	9(002)	
35 38	BIRTH-MONTH-DAY	004	9(004)	
35 36	BIRTH-MONTH	002	9(002)	
37 38	BIRTH-DAY	002	9(002)	
39 40	EMPLOYMENT-CODE	002	X(002)	PE02

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
39 40	EMPLOYMENT-SOURCE	002	X(020)	D PE02
41 46	EMPLOYMENT-DATE	006	9(006)	
41 46	ET-APPL-DATE	006	9(006)	
41 46	ORIGINAL-APPL-DATE	006	9(006)	
47 48	TERM-CODE	002	X(002)	PP479
47 48	TERMINATION-CODE	002	X(002)	PP479
47 48	TERM-CAUSE	002	X(020)	D PP479
49 54	APPL-STATUS-AS-OF	006	9(006)	
49 54	DATE-OF-TERMINATION	006	9(006)	
49 54	ET-APPL-STATUS	006	9(006)	
55 55	APPL-MILITARY-STATUS	001	X(001)	HR07
55 55	ET-MILITARY-STATUS	01	X(001)	HR07
55 55	SHIFT-CODE	001	X(001)	PP43
55 55	APPLICANT-MILITARY	001	X(020)	D HR07
56 58	EDUCATION-LEVEL-CODE	003	9(003)	HR31
56 58	ET-ED-LEVEL	003	9(003)	HR31
56 58	LEVEL-OF-EDUCATION	003	9(020)	D HR31
56 56	SPLIT-CODE	001	9(001)	PP44
57 60	JOB-CATEGORY-CODE	004	X(004)	HR01
57 60	JOB-CATEGORY	004	X(020)	D HR01
57 58	JOB-CATEGORY-2	002	X(002)	EO09
59 60	FILLER	002	X(002)	
61 61	USER-FIELD-E	001	X(001)	
62 64	CURRENT-APPL-STATUS	003	9(003)	AT01
62 64	ET-CURRENT-STATUS	003	9(003)	AT01
62 64	APPLICANT-STATUS	003	9(020)	D AT01
62 62	COMMISSION-FLAG	001	X(001)	PP24
62 62	FAIR-LABOR-CODE	001	X(001)	PP24
63 64	PAY-PDS-TO-STOP-PAY	002	9(002)	

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
63 64	PERIOD-OVERRIDE	002	9(002)	
63 64	PERIODS-TO-STOP-PAY	002	9(002)	
65 66	EMPLOYEE-SECURITY	002	X(002)	
67 70	KEY-FIELDS	004	X(004)	
67 67	KEY-FIELD-1	001	X(001)	
68 68	KEY-FIELD-2	001	X(001)	
69 69	KEY-FIELD-3	001	X(001)	
70 70	KEY-FIELD-4	001	X(001)	
71 71	CLEAR-ANNIV	001	X(001)	

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT  
E SEGMENT  
BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	E
2 3	SEGMENT-CODE	002	X(002)	EA
1 40	HISTORY-LABOR-FLAG	040	X(040)	
2 3	E-CARD-CODE	002	X(002)	
4 10	HISTORY-HOURS	007	9(007)	
11 17	HISTORY-RATE	007	9(007)	
18 25	HISTORY-SALARY	008	9(008)	
26 34	E-ROUTING-NBR	009	X(009)	
26 34	ROUTING-NBR	009	9(009)	
35 40	CHANGE-DATE	006	9(006)	
41 42	RECORD-GROUP	002	X(002)	
41 41	ACTION-CODE	001	X(001)	
42 42	FILLER	001	X(001)	
43 44	TO-DATE-BATCH	002	X(002)	
43 43	TO-DATE-CODE	001	X(001)	
44 44	ADJ-BATCH	001	X(001)	
45 48	MASTER-NUMBER	004	9(004)	
49 50	BANK-CD	002	9(002)	
49 50	E-BANK-CODE	002	X(002)	
51 58	E-RECON-NBR	008	9(008)	
51 58	RECON-NUMBER	008	(008)	
59 59	E-RECON-CLEAR	001	X(001)	
59 59	RECON-CLEAR-CODE	001	X(001)	
60 65	E-PERIOD-DATE	006	9(006)	
60 65	PERIOD-END-DATE	006	9(006)	
66 71	E-RECON-DATE	006	9(006)	
66 71	PAYMNT-DATE	006	9(006)	

## E segment—basic employee, continued

SEGMENT LAYOUT REPORT

E SEGMENT

BASIC EMPLOYEE (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	E
2 3	SEGMENT-CODE	002	X(002)	EB
4 4	EB-TRF-CODE	001	9(001)	J PR22
4 4	TRANS-CODE	001	X(001)	
4 4	EB-TRF-CODE-DESC	001	9(020)	D PR22
5 6	EB-CTL1	002	X(002)	
5 6	TRANS-001	002	X(002)	
7 10	EB-CTL2	004	X(004)	
7 10	TRANS-002	004	X(004)	
11 20	EB-NEW-NUMBER	010	X(010)	
11 20	TRANS-003	010	X(010)	

## F segment—employee name and address

SEGMENT LAYOUT REPORT  
 F SEGMENT  
 EMPLOYEE NAME AND ADDRESS

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	F
1 124	F-SEGMENT	124	X(124)	
1 3	F-SEGMENT-TYPE/CODE	003	X(003)	
1 1	F-SEGMENT-TYPE	001	X(001)	
2 4	NAME-CODE	003	9(003)	K
5 34	EMPLOYEE	030	X(030)	
5 34	EMPLOYEE-NAME	030	X(030)	
5 34	NAME	030	X(030)	
5 34	W2-NAME	030	X(030)	
5 33	NAME-29	029	X(029)	
5 29	EMPLOYEE-NAME-25	025	X(025)	
5 24	EMPLOYEE-NAME-20	020	X(020)	
5 19	EMPLOYEE-NAME-15	015	X(015)	
5 19	NAME-FIRST-15	015	X(015)	
5 14	EMPLOYEE-NAME-10	010	X(010)	
5 9	EMPLOYEE-NAME-5	005	X(005)	
10 34	FILLER	025	X(025)	
35 64	ADDRESS	030	X(030)	
35 64	ADDRESS-LINE-ONE	030	X(030)	
35 35	W2-STATUTORY	001	X(001)	W200
36 36	W2-DECEASED	001	X(001)	W200
37 37	W2-PENSION	001	X(001)	W200
38 38	W2-LEGAL-REP	001	X(001)	W200
39 39	W2-DEFERRED	001	X(001)	W200
40 40	W2-MQGE-STATUS	001	X(001)	W200
41 48	FILLER	008	X(008)	
49 49	W2-TAX-AMT	001	X(001)	W200
50 50	FILLER	001	X(001)	
51 51	W2-IRA/SEP	001	X(001)	W200
52 52	W2-DIST-BOX7	001	X(001)	W210

## F segment—employee name and address, continued

SEGMENT LAYOUT REPORT  
F SEGMENT  
EMPLOYEE NAME AND ADDRESS

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
53 53	W2-DIST-2ND-BOX7	001	X(001)	W218
54 63	FILLER	010	X(010)	
64 64	W2-REPLACE	001	X(001)	W200
65 94	ADDRESS-2	030	X(030)	
65 94	ADDRESS-LINE-TWO	030	X(030)	
95 119	CITY-AND-STATE	025	X(025)	
95 119	CITY/STATE	025	X(025)	
95 118	CITY/PROVINCE	024	X(024)	
95 113	CITY	019	X(019)	
114 115	STATE	002	X(002)	
114 115	STATE	002	X(002)	
116 124	ZIP-CODE+4	009	9(009)	
119 124	POSTAL-CODE	006	X(006)	
120 124	ZIP-CODE	005	9(005)	

## G segment—employee labor splits

SEGMENT LAYOUT REPORT

G SEGMENT

EMPLOYEE LABOR SPLITS

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	G
1 34	G-SEGMENT	034	X(034)	
1 3	G-SEGMENT-TYPE/CODE	003	X(003)	
1 1	G-SEGMENT-TYPE	001	X(001)	
2 3	JOB-APPLIED-SEQ-NBR	002	9(002)	K
2 3	LOCATION-NUMBER	002	9(002)	K
4 9	DATE-FIRST-CONSIDER	006	9(006)	
4 8	LOCATION-PERCENT	005	9(005)	
9 24	CONTROLS-3-THRU-6	016	X(016)	
9 20	CONTROLS-3-THRU-5	012	9(012)	
9 12	CONTROL-3-CODE	004	X(004)	PR019
9 12	GROUPS	004	X(004)	PR019
9 12	CONTROL-3	004	X(015)	D PR019
9 12	COST-CENTER	004	X(015)	D PR019
10 15	JOB-CODE-CONSIDER	006	X(006)	
13 16	CONTROL-4-CODE	004	X(004)	PR029
13 16	CONTROL-4	004	X(015)	D PR029
16 19	JOB-EXTENT-CONSIDER	004	9(004)	
17 20	CONTROL-5-CODE	004	X(004)	PR039
17 20	CONTROL-5	004	X(015)	D PR039
20 23	ET-JOB-EEOC	004	X(004)	EO069
20 23	JOB'S-EEOC-LOCATION	004	X(004)	EO069
20 23	JOB-EEOC-LOCATION	004	X(020)	D EO069
21 24	CONTROL-6-CODE	004	X(004)	PR049
21 24	CONTROL-6	004	X(015)	D PR049
24 27	DEPT-APPLIED-FOR	004	X(004)	HR439
24 27	DEPARTMENT-APPLIED	004	X(020)	D HR439
25 34	FUNCTION-CODE	010	X(010)	PR059
25 34	FUNCTION	010	X(015)	D PR059
28 29	OUTCOME-REASON	002	9(002)	AT02
28 29	OUTCOME	002	9(020)	D AT02

## H segment—employee HEDs

## SEGMENT LAYOUT REPORT

## H SEGMENT

## EMPLOYEE HEDs

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	H
1 101	H-SEGMENT	101	X(101)	
1 1	H-SEGMENT-TYPE	001	X(001)	
2 4	HED-NUMBER	003	9(003)	K
2 4	HED-NUMBER-2	003	9(003)	K
5 6	FREQUENCY-CODE	002	9(002)	
5 6	HED-DEDUCT-FREQ-CD	002	9(002)	PP08
5 6	HED-EARNING-FREQ-CD	002	9(002)	PP07
5 6	HED-DEDUCTION-FREQ	002	9(015)	D PP08
5 6	HED-EARNING-FREQ	002	9(015)	D PP07
7 8	HED-DEDUCT-ARREAR-CD	002	X(002)	PP10
7 8	HED-EARNING-TYPE-CD	002	X(002)	PP09
7 8	TYPE-CODE	002	9(002)	
7 8	HED-DEDUCTION-ARREAR	002	X(015)	D PP10
7 8	HED-EARNING-TYPE	002	X(015)	D PP09
9 10	HED-DEDUCT-METHOD-CD	002	X(002)	PP12
9 10	HED-EARNING-METHOD-C	002	X(002)	PP11
9 10	METHOD-CODE	002	X(002)	
9 10	HED-DEDUCTION-METHOD	002	X(015)	D PP12
9 10	HED-EARNING-METHOD	002	X(015)	D PP11
11 17	AMOUNT	007	9(007)	
11 17	AMOUNT-PERCENT	007	9(007)	
11 17	PAY-RATE	007	9(007)	
18 18	START-CODE	001	X(001)	PP48
18 18	START-DESCRIPTION	001	X(015)	D PP48
19 24	START-FIELD	006	9(006)	
25 25	STOP-CODE	001	X(001)	PP49
25 25	STOP-DESCRIPTION	001	X(015)	D PP49
26 31	STOP-FIELD	006	9(006)	

## H segment—employee HEDs, continued

SEGMENT LAYOUT REPORT

H SEGMENT

EMPLOYEE HEDs (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
32 32	ONE-TIME-CODE	001	X(001)	PP50
32 32	ONE-TIME-DESCRIPTION	001	X(015)	D PP50
33 39	ONE-TIME-AMOUNT	007	9(007)	
40 46	AMOUNT-1	007	9(007)	
40 46	AMOUNT-ONE	007	9(007)	
40 46	NORMAL-HOURS	007	9(007)	
40 40	DIRECT-DEPOSIT-CODE	001	X(001)	PR08
40 40	DIRECT-DEPOSIT-EDIT	001	9(015)	D PR08
41 49	EMPLOYEE-BANK-NUMBER	009	9(009)	PR07
41 49	EMPLOYEE-BANK-NAME	009	9(020)	D PR07
47 54	AMOUNT-2	008	9(008)	
47 54	AMOUNT-TWO	008	9(008)	
47 54	SALARY	008	9(008)	

## H segment—employee HEDs, continued

## SEGMENT LAYOUT REPORT

## H SEGMENT

## EMPLOYEE HEDs (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
50 50	BANK-ACCOUNT	001	9(001)	PR06
50 50	BANK-ACCOUNT-CODE	001	9(001)	PR06
50 50	BANK-ACCOUNT-TYPE	001	9(015)	D PR06
51 67	BANK-ACCOUNT-NUMBER	017	X(017)	
55 56	USER-CODE	002	X(002)	
55 55	USER-CODE2	001	X(001)	
56 56	FILLER	001	X(001)	
57 67	USER-NUMBER	011	X(011)	
68 68	FILLER	001	X(001)	
69 69	H-CHANGE	001	X(001)	
70 73	HED-AMOUNT-CUR	004	9(009)	COMPUTATIONAL
74 77	HED-HOURS-CUR	004	9(007)	COMPUTATIONAL
78 81	HED-AMOUNT-MTD	004	9(009)	COMPUTATIONAL
82 85	HED-HOURS-MTD	004	9(007)	COMPUTATIONAL
86 89	HED-AMOUNT-QTD	004	9(009)	COMPUTATIONAL
90 93	HED-HOURS-QTD	004	9(007)	COMPUTATIONAL
94 97	HED-AMOUNT-YTD	004	9(009)	COMPUTATIONAL
98 101	HED-HOURS-YTD	004	9(007)	COMPUTATIONAL

## H segment—employee HEDs (labor/history record)

SEGMENT LAYOUT REPORT

H SEGMENT

EMPLOYEE HEDs (Labor/History Record)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	H
1 1	H-SEGMENT-CODE-L/H	001	X(001)	
1 1	HED-SEG-L/H	001	X(001)	
2 4	HED-NBR-L/H	003	9(003)	
5 8	HED-AMOUNT-L/H	004	9(009)	COMPUTATIONAL
5 8	HED-AMOUNT-L/H-S	004	9(007)	COMPUTATIONAL
9 12	HED-HOURS-L/H	004	9(007)	COMPUTATIONAL

## J segment—employee taxes

## SEGMENT LAYOUT REPORT

## J SEGMENT

## EMPLOYEE TAXES

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	J
1 1	J-SEGMENT-TYPE	001	X(001)	
2 8	ET-TAX-ID	007	X(007)	K
2 8	TAX-ID	007	X(007)	K
2 2	ET-TAX-TYPE	001	9(001)	P PR15
2 2	TAX-TYPE	001	9(001)	P PR15
2 2	TAX-TYPE-EDIT	001	9(015)	D PR15
3 8	TAX-CITY	006	X(006)	
3 8	TAX-CODE	006	X(006)	K
3 8	TAX-COUNTY	006	X(006)	
3 4	TAX-CODE-2	002	X(002)	K
3 4	TAX-FED-NBR	002	9(002)	
3 4	TAX-STATE-NBR	002	X(002)	
5 8	FED-UNIT	004	9(004)	
5 8	FICA-UNIT	004	9(004)	
5 8	STATE-UNIT	004	9(004)	
5 8	UNIT-NUMBER	004	X(004)	
9 9	TAX-METHOD-CODE	001	X(001)	PR09
9 9	TAX-METHOD	001	X(015)	D PR09
10 10	RESIDENT-CODE	001	X(001)	PR10
10 10	RESIDENT-STATUS	001	X(015)	D PR10
11 12	TAX-DEPENDENTS	002	9(002)	
13 13	TAX-MARITAL-CODE	001	9(001)	PR11
13 13	TAX-MARITAL-STAT-13	001	9(013)	D PR11
13 13	TAX-MARITAL-STATUS	001	9(015)	D PR11
14 20	TAX-AMOUNT-PERCENT	007	9(007)	
21 21	CPP/QPP-UIC-CODE	001	9(001)	PR12
21 21	UNEMPLOYMENT-CODE	001	9(001)	PR12
21 21	UNEMPLOYMENT	001	9(015)	D PR12
22 26	EXEMPTION-AMOUNT	005	9(005)	

## J segment—employee taxes, continued

SEGMENT LAYOUT REPORT

J SEGMENT

EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
22 26	PLEDGE-PERCENT	005	9(005)	
27 27	RECIPROCAL-CODE	001	X(001)	PR13
27 27	TAX-RECIPROCAL	001	X(015)	D PR13
28 32	K3-HD-AMOUNT	005	9(005)	
28 29	RECIPROCAL-STATE	002	X(002)	
30 30	TAX-TABLE	001	9(001)	
31 31	WORK-FLAG-CODE	001	9(001)	PR14
31 31	WORK-FLAG	001	9(015)	D PR14
32 34	FILLER	003	X(003)	
35 35	J-CHANGE	001	X(001)	
36 39	EMPLOYER-WAGES-CUR	004	9(009)	COMPUTATIONAL
36 39	ET-TAX-WORK-CUR	004	9(009)	COMPUTATIONAL

**J segment—employee taxes, continued**

## SEGMENT LAYOUT REPORT

## J SEGMENT

## EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
36 39	TAX-WORK-CUR	004	9(009)	COMPUTATIONAL
36 39	TAXABLE-BENEFITS-CUR	004	9(009)	COMPUTATIONAL
40 43	EMPLOYEE-WAGES-CUR	004	9(009)	COMPUTATIONAL
40 43	ET-TAX-RES-CUR	004	9(009)	COMPUTATIONAL
40 43	TAX-RESIDENT-CUR	004	9(009)	COMPUTATIONAL
44 47	ET-TAX-WAGES-CUR	004	9(009)	COMPUTATIONAL
44 47	TAXABLE-WAGES-CUR	004	9(009)	COMPUTATIONAL
44 47	TOTAL-PAY-CUR	004	9(009)	COMPUTATIONAL
48 51	EMPLOYER-TAX-CUR	004	9(007)	COMPUTATIONAL
48 51	ET-TAX-WITHHELD-CUR	004	9(009)	COMPUTATIONAL
48 51	TAX-WITHHELD-CUR	004	9(009)	COMPUTATIONAL
52 55	C/QPP-UIC-WAGES-CUR	004	9(009)	COMPUTATIONAL
52 55	ET-TAX-UNEMPLOY-CUR	004	9(009)	COMPUTATIONAL
52 55	TAX-UNEMPLOY-CUR	004	9(009)	COMPUTATIONAL
56 59	C/QPP-UIC-DED-CUR	004	9(005)	COMPUTATIONAL
56 59	EMPLOYEE-TAX-CUR	004	9(007)	COMPUTATIONAL
56 59	ET-TAX-DIS-CUR	004	9(005)	COMPUTATIONAL
56 59	TAX-DISABILITY-CUR	004	9(005)	COMPUTATIONAL
56 59	TAX-EIC-CUR	004	9(005)	COMPUTATIONAL
60 63	ET-TAX-PREM-CUR	004	9(009)	COMPUTATIONAL
60 63	TAX-PREMIUM-CUR	004	9(009)	COMPUTATIONAL
64 67	ET-TAX-WEEKS-CUR	004	9(004)	COMPUTATIONAL
64 67	TAX-WEEKS-CUR	004	9(004)	COMPUTATIONAL
68 71	EMPLOYER-WAGES-MTD	004	9(009)	COMPUTATIONAL
68 71	ET-TAX-WORK-MTD	004	9(009)	COMPUTATIONAL
68 71	TAX-WORK-MTD	004	9(009)	COMPUTATIONAL
68 71	TAXABLE-BENEFITS-MTD	004	9(009)	COMPUTATIONAL
72 75	EMPLOYEE-WAGES-MTD	004	9(009)	COMPUTATIONAL
72 75	ET-TAX-RES-MTD	004	9(009)	COMPUTATIONAL

## J segment—employee taxes, continued

SEGMENT LAYOUT REPORT

J SEGMENT

EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
72 75	TAX-RESIDENT-MTD	004	9(009)	COMPUTATIONAL
76 79	ET-TAX-WAGES-MTD	004	9(009)	COMPUTATIONAL
76 79	TAXABLE-WAGES-MTD	004	9(009)	COMPUTATIONAL
76 79	TOTAL-PAY-MTD	004	9(009)	COMPUTATIONAL
80 83	EMPLOYER-TAX-MTD	004	9(007)	COMPUTATIONAL
80 83	ET-TAX-WITHHELD-MTD	004	9(009)	COMPUTATIONAL
80 83	TAX-WITHHELD-MTD	004	9(009)	COMPUTATIONAL
84 87	C/QPP-UIC-WAGES-MTD	004	9(009)	COMPUTATIONAL
84 87	ET-TAX-UNEMPLOY-MTD	004	9(009)	COMPUTATIONAL
84 87	TAX-UNEMPLOY-MTD	004	9(009)	COMPUTATIONAL
88 91	C/QPP-UIC-DED-MTD	004	9(005)	COMPUTATIONAL
88 91	EMPLOYEE-TAX-MTD	004	9(007)	COMPUTATIONAL

**J segment—employee taxes, continued**
 SEGMENT LAYOUT REPORT  
 J SEGMENT  
 EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
88 91	ET-TAX-DIS-MTD	004	9(005)	COMPUTATIONAL
88 91	TAX-DISABILITY-MTD	004	9(005)	COMPUTATIONAL
88 91	TAX-EIC-MTD	004	9(005)	COMPUTATIONAL
92 95	ET-TAX-PREM-MTD	004	9(009)	COMPUTATIONAL
92 95	TAX-PREMIUM-MTD	004	9(009)	COMPUTATIONAL
96 99	ET-TAX-WEEKS-MTD	004	9(004)	COMPUTATIONAL
96 99	TAX-WEEKS-MTD	004	9(004)	COMPUTATIONAL
100 103	EMPLOYER-WAGES-QTD	004	9(009)	COMPUTATIONAL
100 103	ET-TAX-WORK-QTD	004	9(009)	COMPUTATIONAL
100 103	TAX-WORK-QTD	004	9(009)	COMPUTATIONAL
100 103	TAXABLE-BENEFITS-QTD	004	9(009)	COMPUTATIONAL
104 107	EMPLOYEE-WAGES-QTD	004	9(009)	COMPUTATIONAL
104 107	ET-TAX-RES-QTD	004	9(009)	COMPUTATIONAL
104 107	TAX-RESIDENT-QTD	004	9(009)	COMPUTATIONAL
108 111	ET-TAX-WAGES-QTD	004	9(009)	COMPUTATIONAL
108 111	TAXABLE-WAGES-QTD	004	9(009)	COMPUTATIONAL
108 111	TOTAL-PAY-QTD	004	9(009)	COMPUTATIONAL
112 115	EMPLOYER-TAX-QTD	004	9(007)	COMPUTATIONAL
112 115	ET-TAX-WITHHELD-QTD	004	9(009)	COMPUTATIONAL
112 115	TAX-WITHHELD-QTD	004	9(009)	COMPUTATIONAL
116 119	C/QPP-UIC-WAGES-QTD	004	9(009)	COMPUTATIONAL
116 119	ET-TAX-UNEMPLOY-QTD	004	9(009)	COMPUTATIONAL
116 119	TAX-UNEMPLOY-QTD	004	9(009)	COMPUTATIONAL
120 123	C/QPP-UIC-DED-QTD	004	9(005)	COMPUTATIONAL
120 123	EMPLOYEE-TAX-QTD	004	9(007)	COMPUTATIONAL
120 123	ET-TAX-DIS-QTD	004	9(005)	COMPUTATIONAL
120 123	TAX-DISABILITY-QTD	004	9(005)	COMPUTATIONAL
120 123	TAX-EIC-QTD	004	9(005)	COMPUTATIONAL
124 127	ET-TAX-PREM-QTD	004	9(009)	COMPUTATIONAL

## J segment—employee taxes, continued

SEGMENT LAYOUT REPORT  
 J SEGMENT  
 EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
124 127	TAX-PREMIUM-QTD	004	9(009)	COMPUTATIONAL
128 131	ET-TAX-WEEKS-QTD	004	9(004)	COMPUTATIONAL
128 131	TAX-WEEKS-QTD	004	9(004)	COMPUTATIONAL
132 135	EMPLOYER-WAGES-YTD	004	9(009)	COMPUTATIONAL
132 135	ET-TAX-WORK-YTD	004	9(009)	COMPUTATIONAL
132 135	TAX-WORK-YTD	004	9(009)	COMPUTATIONAL
132 135	TAXABLE-BENEFITS-YTD	004	9(009)	COMPUTATIONAL
136 139	EMPLOYEE-WAGES-YTD	004	9(009)	COMPUTATIONAL
136 139	ET-TAX-RES-YTD	004	9(009)	COMPUTATIONAL
136 139	TAX-RESIDENT-YTD	004	9(009)	COMPUTATIONAL
140 143	ET-TAX-WAGES-YTD	004	9(009)	COMPUTATIONAL
140 143	TAXABLE-WAGES-YTD	004	9(009)	COMPUTATIONAL

**J segment—employee taxes, continued**
 SEGMENT LAYOUT REPORT  
 J SEGMENT  
 EMPLOYEE TAXES (Cont'd)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
140 143	TOTAL-PAY-YTD	004	9(009)	COMPUTATIONAL
144 147	EMPLOYER-TAX-YTD	004	9(007)	COMPUTATIONAL
144 147	ET-TAX-WITHHELD-YTD	004	9(009)	COMPUTATIONAL
144 147	TAX-WITHHELD-YTD	004	9(009)	COMPUTATIONAL
148 151	C/QPP-UIC-WAGES-YTD	004	9(009)	COMPUTATIONAL
148 151	ET-TAX-UNEMPLOY-YTD	004	9(009)	COMPUTATIONAL
148 151	TAX-UNEMPLOY-YTD	004	9(009)	COMPUTATIONAL
152 155	C/QPP-UIC-DED-YTD	004	9(005)	COMPUTATIONAL
152 155	EMPLOYEE-TAX-YTD	004	9(007)	COMPUTATIONAL
152 155	ET-TAX-DIS-YTD	004	9(005)	COMPUTATIONAL
152 155	TAX-DISABILITY-YTD	004	9(005)	COMPUTATIONAL
152 155	TAX-EIC-YTD	004	9(005)	COMPUTATIONAL
156 159	EST-ANNUAL-INCOME	004	9(009)	COMPUTATIONAL
156 159	ET-TAX-PREM-YTD	004	9(009)	COMPUTATIONAL
156 159	TAX-PREMIUM-YTD	004	9(009)	COMPUTATIONAL
160 163	ET-TAX-WEEKS-YTD	004	9(004)	COMPUTATIONAL
160 163	TAX-WEEKS-YTD	004	9(004)	COMPUTATIONAL

## J segment—employee taxes (history record)

SEGMENT LAYOUT REPORT  
 J SEGMENT  
 EMPLOYEE TAXES (History Record)

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	J
1 1	TAX-SEG-HIS	001	X(001)	
2 8	TAX-NBR-HIS	007	X(007)	
2 4	TAX-NBR-HIS-3	003	X(003)	
5 8	FILLER	004	X(004)	
9 12	EMPLOYER-WAGES-HIS	004	9(009)	COMPUTATIONAL
9 12	TAX-WORK-HIS	004	9(009)	COMPUTATIONAL
13 16	EMPLOYEE-WAGES-HIS	004	9(009)	COMPUTATIONAL
13 16	TAX-RESIDENT-HIS	004	9(009)	COMPUTATIONAL
17 20	TAXABLE-WAGES-HIS	004	9(009)	COMPUTATIONAL
17 20	TOTAL-PAY-HIS	004	9(009)	COMPUTATIONAL
21 24	EMPLOYER-TAX-HIS	004	9(007)	COMPUTATIONAL
21 24	TAX-WITHHELD-HIS	004	9(009)	COMPUTATIONAL
25 28	TAX-UNEMPLOY-HIS	004	9(009)	COMPUTATIONAL
29 32	EMPLOYEE-TAX-HIS	004	9(007)	COMPUTATIONAL
29 32	TAX-DISABILITY-HIS	004	9(005)	COMPUTATIONAL
29 32	TAX-EIC-HIS	004	9(005)	COMPUTATIONAL
33 36	TAX-PREMIUM-HIS	004	9(009)	COMPUTATIONAL
33 36	TAX-PREMIUM-HIS-X	004	9(008)	COMPUTATIONAL
37 40	TAX-WEEKS-HIS	004	9(004)	COMPUTATIONAL
37 40	TAX-WEEKS-HIS-X	004	9(004)	COMPUTATIONAL

## L segment—employee spouse/dependent

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYEE SPOUSE/DEPENDENT

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE		001	X(001) L
2 3	SEGMENT-CODE		002	X(002) O1
4 6	OTHER-KEY		003	X(003) K
7 8	RELATIONSHIP-CODE		002	X(002) HR28
7 8	RELATIONSHIP		002	X(015) D HR28
9 38	OTHER-NAME		030	X(030)
9 33	OTHER-NAME-25		025	X(025)
9 28	OTHER-NAME-20		020	X(020)
9 23	OTHER-NAME-15		015	X(015)
24 38	FILLER		015	X(015)
39 50	OTHER-SOC-INSURANCE		012	X(012)
39 50	OTHER-SOC-SECURITY		012	X(012)
51 51	OTHER-SEX-CODE		001	X(001) HR29
51 51	OTHER-SEX		001	X(010) D HR29
52 57	OTHER-BIRTH-DATE		006	9(006)
58 60	OTHER-AREA-CODE		003	9(003)
61 68	OTHER-PHONE-NUMBER		008	X(008)
69 69	FULL-TIME-STUDENT		001	X(001) HR30
69 69	STUDENT-STATUS		001	X(020) D HR30

## L segment—employment status

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYMENT STATUS

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	L
2 3	SEGMENT-CODE	002	X(002)	ZC
4 9	ACTIVITY-DATE	006	9(006)	P
4 5	ACTIVITY-YEAR	002	9(002)	P
6 9	FILLER	004	X(004)	
10 12	ACTIVITY-CODE	003	X(003)	K HR09
10 12	ACTIVITY	003	X(020)	D HR09
10 12	ACTIVITY-10	003	X(010)	D HR09
10 12	ACTIVITY-15	003	X(015)	D HR09
10 12	ACTIVITY-FIRST15	003	X(015)	D HR09
10 10	ACTIVITY-CODE-1	001	X(001)	
11 12	FILLER	002	X(002)	
13 14	RESULTING-EMP-STATUS	002	X(002)	
13 14	EMPLOYEE-STATUS	002	X(020)	D HR10
13 14	EMPLOYEE-STATUS-10	002	X(010)	D HR10
13 14	EMPLOYEE-STATUS-15	002	X(015)	D HR10
13 13	EMPLOYEE-STATUS-1	001	X(001)	
13 13	RESULTING-STATUS-1	001	X(001)	
14 14	RESULTING-STATUS-2	001	X(001)	
15 20	EXPECTED-RETURN	006	9(006)	
15 20	EXPECTED-RETURN-DATE	006	9(006)	
21 26	LAST-DAY-WORKED	006	9(006)	
27 32	LAST-DAY-PAID	006	9(006)	
33 38	HISTORY-C1-2	006	X(006)	
33 34	CTRL-ONE	002	X(002)	
35 38	CTRL-TWO	004	X(004)	
39 44	ADJUSTED-SENIORITY	006	9(006)	
45 50	SEPARATION-DATE	006	9(006)	
51 56	HR-PARITY-REQ-DATE	006	9(006)	
57 62	HR-LEAVE-SPAN	006	9(006)	
63 68	ORIGINAL-HIRE-DATE	006	9(006)	
63 64	ORIGINAL-HIRE-YEAR	002	9(002)	
65 66	ORIGINAL-HIRE-MONTH	002	9(002)	
67 68	ORIGINAL-HIRE-DAY	002	9(002)	

## L segment—employment status, continued

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYMENT STATUS

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	L
2 3	SEGMENT-CODE	002	X(002)	ZD
4 9	JOB-EFFECTIVE	006	9(006)	P
10 10	KEY-SEPARATOR	001	9(001)	P HR70
10 10	KEY-SEPARATOR-DESC	001	9(020)	D HR70
11 13	CHANGE-TYPE	003	X(003)	K HR33
11 13	CHANGE	003	X(020)	D HR33
11 13	CHANGE-FIRST15	003	X(015)	D HR33
14 19	JOB-CODE	006	X(006)	
20 23	JOB-CODE-EXTENT	004	9(004)	
24 28	JOB-UNION-CODE	005	X(005)	PP429
24 28	JOB-UNION	005	X(015)	D PP429

## L segment—employee salary

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYEE SALARY

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	L
2 3	SEGMENT-CODE	002	X(002)	ZF
4 5	INCUMBENCY-CODE	002	X(002)	P
6 11	SALARY-DATE	006	9(006)	P
6 11	SALARY-EFFECTIVE	006	9(006)	P
12 12	SALARY-SEQUENCE	001	X(001)	P
12 12	ZF-KEY-SEPARATOR	001	9(001)	P HR70
12 12	ZF-KEY-SEPARATOR-DES	001	9(020)	D HR70
13 15	SALARY-CHANGE-TYPE	003	X(003)	K HR11
13 15	SALARY-CHANGE	003	X(020)	D HR11
13 15	SALARY-CHANGE-15	003	X(015)	D HR11
16 22	HOURLY-RATE	007	9(007)	
23 27	HOURS-PER-PERIOD	005	9(005)	
28 28	PAYMENT-FREQUENCY	001	X(001)	PP29
28 28	FREQUENCY-TYPE	001	X(020)	D PP29
28 28	PAY-FREQUENCY-10	001	X(010)	D PP29
29 36	SALARY-PER-PERIOD	008	9(008)	
37 45	ANNUAL-SALARY	009	9(009)	
46 53	ANNUAL-AMOUNT-CHANGE	008	9(008)	
54 57	PERCENT-CHANGE	004	9(004)	
58 60	MONTHS-SINCE	003	9(003)	
61 63	COMPA-RATIO	003	9(003)	
64 68	POSITION-IN-RANGE	005	9(005)	
64 68	RANGE-PENETRATION	005	9(005)	
69 69	RED/GREEN-INDICATOR	001	X(001)	HR14
69 69	RANGE-INDICATOR	001	X(020)	D HR14
70 70	SALARY-CALC-METHOD	001	X(001)	

## L segment—employee location

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYEE LOCATION

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	L
2 3	SEGMENT-CODE	002	X(002)	ZR
4 9	LOCATION-EFFECTIVE	006	9(006)	P
10 10	ZR-KEY-SEPARATOR	001	9(001)	P HR70
10 10	ZR-KEY-SEPARATOR-DES	001	9(020)	D HR70
11 13	LOCATION-CHANGE-TYPE	003	X(003)	K HR65
11 13	LOCATION-CHANGE	003	X(020)	D HR65
14 17	EEO-ESTABLISHMENT	004	X(004)	EO069
14 17	EEO-ESTABLISH	004	X(020)	D EO069
14 17	EEO-ESTABLISH-15	004	X(015)	D EO069
14 17	ESTABLISHMENT	004	X(020)	D EO069
18 33	CTRL-3-THRU-6	016	X(016)	
18 29	CTRL-3-THRU-5	012	X(012)	
18 21	CTRL-THREE	004	X(004)	HR439
18 21	CONTROL-THREE	004	X(020)	D HR439
18 21	CONTROL-THREE-15	004	X(015)	D HR439
22 25	CTRL-FOUR	004	X(004)	HR449
22 25	DEPART-MENT	004	X(004)	HR449
22 25	DEPARTMENT	004	X(004)	HR449
22 25	DEPARTMENT-TOTAL	004	X(004)	HR449
22 25	CONTROL-FOUR	004	X(020)	D HR449
22 25	CONTROL-FOUR-15	004	X(015)	D HR449
26 29	CTRL-FIVE	004	X(004)	HR459
26 29	CONTROL-FIVE	004	X(020)	D HR459
26 29	CONTROL-FIVE-15	004	X(015)	D HR459
30 33	CTRL-SIX	004	X(004)	HR469
30 33	CONTROL-SIX	004	X(020)	D HR469
30 33	CONTROL-SIX-15	004	X(015)	D HR469
34 34	GEOGRAPHIC-RANGE-IND	001	X(001)	HR66
34 34	GEOGRAPHIC-RANGE	001	X(020)	D HR66
35 44	MAIL-DISTRIBUTE-DATA	010	X(010)	

## L segment—employee training

SEGMENT LAYOUT REPORT  
L SEGMENT  
EMPLOYEE TRAINING

POSITIONS	FIELD NAME	LENGTH	PIC	COMMENTS
1 1	SEGMENT-TYPE	001	X(001)	L
2 3	SEGMENT-CODE	002	X(002)	Z3
4 6	TRAINING-COURSE-CODE	003	X(003)	P MP02
4 6	TRAINING-COURSE	003	X(020)	D MP02
7 8	TRAINING-YEAR	002	9(002)	P
9 9	TRAINING-COURSE-NO	001	X(001)	K
10 10	TRAIN-COMPLETION-CD	001	X(001)	MP06
10 10	TRAINING-COMPLETE-CD	001	X(001)	MP06
10 10	TRAIN-COMPLETION	001	X(020)	D MP06
11 13	TRAINING-TYPE-CODE	003	X(003)	MP03
11 13	TRAINING-TYPE	003	X(020)	D MP03
14 17	COURSE-LOCATION-CODE	004	X(004)	MP049
14 17	COURSE-LOCATION	004	X(020)	D MP049
18 19	COURSE-INSTRUCTOR-CD	002	X(002)	MP059
18 19	COURSE-INSTRUCTOR	002	X(020)	D MP059
20 25	COURSE-START-DATE	006	9(006)	
26 31	COURSE-COMPLETION-DT	006	9(006)	
32 35	TRAINING-RATING	004	9(004)	
36 37	TRAINING-CREDIT	002	9(002)	
38 43	TRAINING-COSTS	006	9(006)	
44 49	TRAINING-COMPANY-PD	006	9(006)	
50 65	TRAINING-CONTROL-3-6	016	X(016)	
50 53	TRAINING-CONTROL-3	004	X(004)	HR439
54 57	TRAINING-CONTROL-4	004	X(004)	HR449
58 61	TRAINING-CONTROL-5	004	X(004)	HR459
62 65	TRAINING-CONTROL-6	004	X(004)	HR469
66 66	COURSE-EVALUATION-CD	001	X(001)	MP09
66 66	COURSE-EVALUATION	001	X(020)	D MP09

## Appendix D: FIND-Verbs and Tables

---

### Table of Contents

FIND- verbs .....	338
HR base module tables.....	339
Benefits tables.....	340

## FIND- verbs

FIND- Macro	Segment Code	Program	Form Usage
FIND-ABSENCE	LVA	93-SCR	Excused/Unexcused Absences
FIND-ACTIVITY	LZC	01-SCR, 04-SCR 08-SCR, 09-SCR 95-SCR, 96-SCR	Employment Status
FIND-BASIC-DATA	LZA	03-SCR	Telephone and New Hire
FIND-CURRENT-PH	LPH	PH-SCR	Salary
FIND-DISABILITY	LRD	DA-SCR	Disciplinary Action
FIND-EMERGENCY-DATA	LZM	17-SCR	Disability/Work Restrictions
FIND-GRIEVANCE	LVG	GT-SCR	Grievance Recording/Status
FIND-JOB-EFFECTIVE	LZD	05-SCR	Job Assignments
FIND-LOCATION	LZR	05CSCR	Location Information
FIND-NEXT-APPRAISAL	LZS	50-SCR	Next Scheduled Performance Appraisal
FIND-NEXT-REVIEW	LZQ	43-SCR	Scheduled Salary Review
FIND-POSITION	LRS	001SCR	Position Assignments
FIND-PRIOR-JOB	LZ6	006SCR	Applicant Work History
FIND-RATING	LZG	49-SCR	Performance Appraisals
FIND-RECORD-DATE	LZB	02-SCR	Personal and Identification Data
FIND-SALARY-AS-OF	LZF	40-SCR	Salary Information
FIND-SALARY-PLAN	LZP	45-SCR	Salary Planning Information
FIND-SCHEDULE *	LTS	TAASCR	Schedule Assignments
FIND-TERMINATION	LZK	97-SCR	Exit Interview Data

*Note: A program-supplied date moved to WORK-DATE is required before executing this verb. The screen programs listed access the given segments but do not necessarily use the FIND- macro.*

## HR base module tables

Table Identifier	Table Use	Table Read Verb	Program Example	Prior Access Required
TA	Job Codes	READ-TA-TABLE	1N-RPT	Employee Segment: Job (LZD)
TB	Salary Grades	READ-TB-TABLE	11-RPT	Employee Salary Segment: (LZF)
TC	Job Points	READ-TC-TABLE	5I-RPT	Employee Job Segment: (LZD)
TE	Occupation Groups	READ-TE-TABLE	71-RPT	READ-TA-TABLE and Employee Segment: Job (LZD)
TF	Activity Validation	READ-TF-TABLE	01-SCR	Employee Segment: Activity (LZC)
TG	System Options	READ-TG-TABLE	05CSCR GG-SCR	Related Segments: (No prior access required) Labor Distribution (G) or Location (LZR) or Salary (LZF) or HED 001 (H)
TH	Salary Planning	READ-TH <sub>x</sub> -TABLE	5H-RPT	Employee Segments: Job (LZD)
TX	EEO Locations	READ-TX-TABLE	05CSCR	Employee Segments: Location (LZR)

*Note: In all cases, a TZAX record (See AX-SCR) is automatically retrieved and supplies the Control Number value.*

## Benefits tables

Table Identifier	Table Use	Table Read Verb	Program Example	Related Data Elements
TK	Plan Rules	READ-TK-TABLE	55-SCR 4X-RPT	Segments: LQ1 or LQ4
TK	Plan Rules	READ-TK-TABLE-BATCH	4I-RPT	RUNREP Parameters Required: Plan ID/Date
TL	Eligibility Rules	READ-TL-TABLE	55ELIG	Segments: LQ1 or LQ4 and READ-TK-TABLE
TM	Plan Factors	READ-TM-TABLE	55-HED	Segments: LQ1 or LQ4 and READ-TK-TABLE
TN	Participation Rules	READ-TN-TABLE	64SERV	Segment: LQ4 and READ-TK-TABLE
TO	Considered Hours/Earnings	READ-TO-TABLE	4O-RPT	Segment: LQ4 and READ-TK-TABLE
TP	Master Plan	READ-TP-TABLE	4C-RPT	Segment:LQ1 and READ-TK-TABLE
TRA	Plan Interest Rates	READ-TRA-TABLE	9RABPT	Deferred Plan Usage
TRB	Plan Fund Interest Rates	READ-TRB-TABLE	9RABPT	Deferred Plan Usage
TRC	Plan Allocations	READ-TRC-TABLE	9RCDPT	Deferred Plan Usage
TQ	Annuitant Factors	READ-TQ-TABLE	9Q-RPT	READ-TK-TABLE
TS	Plan Prototype Contribution HED	READ-TS-TABLE	55-HED	Segments: LQ1 or (LQ4) and READ-TK-TABLE
TT	Activity/Option Validation	READ-TT-TABLE	55-OPT	Segments: LQ1 or LQ4 and Activity Segment LZC for TT Activity Tables and READ-TK-TABLE
TU	Breaks in Service Component Plans	READ-TU-TABLE	PBREAK	Segments: LQ1 or LQ4 and READ-TK-TABLE
TV	Nondiscrimination Test Component Plans	READ-TV-TABLE	2C-RPT	READ-TK-TABLE

*Note: Most Benefits Table reads require data be moved to the associated Table work fields before issuing the READ-Tx verbs listed above. The LQ1 Segment stores Welfare Plan enrollments; the LQ4 stores Deferred Plan enrollments.*

In all cases the TZAY (See AY-SCR) Table record is automatically retrieved and supplies the Control Number value.

## Appendix E: Print Position Compile (RETYPE) Error Resolution

---

### Table of Contents

Print Position Compile (RETYPE) Error Resolution .....	342
--	-----

## Print Position Compile (RETYPE) Error Resolution

### 1. INVALID SORT

A RETYPE attempt has been made when no object code for the associated report extract program exists.

**TO RESOLVE:** RELOAD the \_\_\_\_ PT program or correct your RTEDIT name as it may have been keyed in error. Then, RETYPE the RTEDIT again.

### 2. DEFINE-REPORT MISSING IN EXTRACT RECORD

The DEFINE-REPORT instruction was not in the extract source program for the initial RELOAD.

**TO RESOLVE:** Add the DEFINE-REPORT instruction to the \_\_\_\_ PT program and RELOAD the program. Then, RETYPE again.

### 3. INVALID REPORT

The RTEDIT records have been purged from the System Control Repository (FILE01).

**TO RESOLVE:** Reestablish the RTEDIT records. Then, RETYPE again.

### 4. fieldname NOT IN FIELD TABLE

A field originally in the Field Name Table when the RTEDIT was established is no longer there for the RETYPE process.

**TO RESOLVE:** Reestablish the missing field or find a substitute changing the RTEDIT reference. Then, RETYPE again.

## Print Position Compile (RETYPE) Error Resolution, continued

### 5. TOO BIG, REDUCE NUMBER OF FIELDS OR SORT LEVELS

This message occurs when the executable length of the object code for both the \_\_\_\_PT program and RTEDIT component exceed 9900 bytes. The sort logic for your report may have caused the excess executable length, but not necessarily.

**TO RESOLVE:** Review your \_\_\_\_PT logic for efficiency. Consider making common routines within your program a single subroutine.

*Note:* Your report does not operate properly when more than nine sort levels are indicated, but this alone does not cause the error.

### 6. WRITE ERROR

The disk space for your System Control Repository needs to be increased. Records could not be written to FILE01.

**TO RESOLVE:** Contact Operations for assistance.

### 7. REWRITE ERROR

The disk space for your System Control Repository needs to be increased or a duplicate key update has been simultaneously attempted. Records could not be rewritten to FILE01.

**TO RESOLVE:** Contact Operations for assistance.

### 8. RETYPE FAIL

This message appears if any of the above conditions are detected. Resolve all errors and RETYPE.

---

**NOTES**

---

## Glossary

---

## A

### ALLOCATE-NN

This verb allocates nn positions of data into WORK for use by a Batch Report/Process Program.

The ALLOCATE command must appear in the DEFINE-REPORT portion of the source code.

Up to 90 positions can be allocated using verbs ALLOCATE-01 through ALLOCATE-90.

Only one verb may be used per program.

There is also a special ALLOCATE-DATE verb that allocates six positions.

See DEFINE-REPORT.

DEFINE-REPORT ALLOCATE-*nn*.

### ALLOCATE-DATE

This verb allocates six positions of data into WORK for use by a Batch Report/Process program.

When used, this command must appear in the DEFINE-REPORT portion of the source code.

DEFINE-REPORT ALLOCATE-DATE.

## B

### BREAK-DEFAULT

This is an CSL batch report command. It is a sort key option placed after a field name when special control/page breaks or totaling is being performed.

This command is used in the sort paragraph. If special control breaks or totaling are used, a sort option must appear after every field prior to the SORT-LENGTH-*nn* command.

BREAK-DEFAULT is the default sort option used when no special spacing or totaling is needed on the field. It is only valid when used PRIOR to the SORT-LENGTH-*nn* command.

{PRINT OUTPUT} fieldname BREAK-DEFAULT.

## C

### CALCULATE

CALCULATE lets you perform arithmetic operations.

- Fields are processed in order, left to right.
- Only simple operations are allowed. Do not use parentheses or try to perform complex operations with the CALCULATE verb.
- Only numeric fields are valid with CALCULATE.
- Separate calculation symbols with spaces: CALCULATE HOURLY-PAY \* :40 = WEEKLY-PAY.
- Use a colon (: ) to identify a numeric literal value, for example, :40.
- When you use fields in Pointers 6, 7, or 8 (work areas) in a calculation, the system assumes that the fields have no decimal places.

CALCULATE {literal fieldname1} { / \* + - } {literal fieldname2} = fieldname3.

**CALL**

The CALL verb lets you transfer control to another program temporarily.

The program that contains the CALL verb is referred to as the 'calling program', and the program that receives temporary control is referred to as the 'called program'.

- Control returns to the calling program when the called program finds the RETURN verb. Processing continues with the statement after the CALL.
- You can specify multiple CALL statements in a program, but you can nest only five CALLs. The system ignores any remaining nested CALLs when it processes five deep.
- However, remember that every program already contains one implicit CALL: that made by CYB90-, the root program that controls all processing in the system.

CYB90 always gains control when an online or batch program completes processing.

In addition, some macro verbs, for example READ-COMPANY, contain a CALL that you must count if you nest CALL statements.

- The names of calling programs are held in a work area in Pointer 7.
- If you do not specify a called program with the CALL verb, the system uses the value in the KEY01-PROGRAM field (W7-06-012) as the name of the called program.

CALL. { literal fieldname1 }

**D****DEFINE-REPORT**

The DEFINE-REPORT command is used in coding batch reports only. It signals the system that the commands that follow it define report options you have selected.

The DEFINE-REPORT commands control the printing of headings, including standard headings. If no options are chosen, the system defaults are used for heading information.

The system assumes all four heading lines are to be included, unless you specify a DEFINE-REPORT option suppressing them.

Input parameters are allocated in the DEFINE-REPORT section of the program.

DEFINE-REPORT [report option...].

**DOUBLE-SPACE-AFTER**

This command is used in coding batch reports. It is a sort key option placed after a control break field name in the sort paragraph.

DOUBLE-SPACE-AFTER is used when you want the printer to advance 2 lines after printing the control break for the field.

It is only valid when used PRIOR to the SORT-LENGTH-nn command.

{PRINT OUTPUT} fieldname DOUBLE-SPACE-AFTER.

#### DOUBLE-SPACE- BEFORE

This command is used in coding batch reports. It is a sort key option placed after a BEFORE control break field name in the sort paragraph.

DOUBLE-SPACE-BEFORE is used when you want the printer to advance lines before printing the control break for the field.

It is only valid when used PRIOR to the SORT-LENGTH-nn command.

*{PRINT OUTPUT} fieldname DOUBLE-SPACE-BEFORE.*

### **F**

#### FIND

The FIND verb enables you to locate a specific occurrence of a segment.

- Follow a FIND statement with an IF FOUND or IF NOT-FOUND statement to test the result of the FIND operation.
- Unless you use one of the following valid qualifiers, FIND begins at the first segment in the pointer:
  - FROM HERE - Begins the FIND operation where the pointer is currently positioned. Be sure that you know the position if you use this qualifier.
  - STARTING WITH - Begins the FIND operation at a specified key value.

*FIND non-key-field-1 [FROM HERE STARTING WITH {literal key-field-name-1 }].*

#### FIRST-TIME-IND

The FIRST-TIME-IND field is also known as W6-01-035. It is used with the Allocate area in batch report programming. Valid values are:

F = first time Allocate area is written (per Control 1-2)

N = other than the first time that the Allocate is written

Y = update Allocate area and save it

The 'F' and 'Y' values are automatically set. The 'Y' value is programmer controlled. This value can be tested/changed in a report program when creating Period records and for first-time initialization routines such as editing input parameters.

#### FORMS/REPORT-CODE

The FORMS/REPORT-CODE command must be used to tell the batch report program the form of the report and the file name of the source code.

Valid Forms Codes are:

1 = FILE03

2 = FILE17

3 = FILE18

4 = FILE19

The Report Code consists of the first four characters of the source code file name. For example, if the source code file is ZZ-RPT, the Report Code is 'ZZ-R'.

The FORMS/REPORT-CODE command must be preceded by the literals Forms and Report Code values in quotes. The FORMS/REPORT-CODE literal for the ZZ-RPT is: '1ZZ-R' (FILE03).

*{PRINT OUTPUT} FORMS/REPORTCODE literal Forms/ReportCode.*

#### F14-REC-01-90

This field name defines the first 90 characters of the extract record (FILE14) in the RTPRNT process.

An RP program can access this field to address the data (including the sort key) in the extract record. The data is in extract record format as per the report extract program.

#### F14-REC91-150

This field name defines the second 60 characters of the extract record (FILE14) in the RTPRNT process.

An RP program can access this field to address these characters of the extract record. The data is in extract record format as per the report extract program.

## G

### GO TO

The GO TO verb transfers control to another paragraph in your program. This verb provides an alternative to normal sequential processing, where each statement is executed in the order that it appears.

- GO TO is one of the few multi-word verbs that does not use a hyphen. However, TO is required.
- The specified paragraph label can occur before or after the GO TO statement. Control passes to the first matching label that physically follows the GO TO statement. If none exists after the statement, control passes to the first matching paragraph label found by going back through the program from the GO TO statement.
- You can use the statement GO UP TO to avoid any confusion caused by duplicate paragraph numbers.
- Processing continues with the statement that follows the paragraph label specified in the GO TO statement.

GO [UP] TO Pnnn-[label].

## H

### HEADER-n

There are four standard report headings that automatically appear on the top of each page. Headers 1 and 2 have certain default data. The report title, subtitle and detail headings are supplied as literal values in Headers 1-4. Replace the n in the syntax with a value of 1-4.

The beginning print position is indicated in the syntax. Then, the heading literal is enclosed in single quotes.

The heading lines can be suppressed using the NO-HEADINGS define report option. Portions of this heading line can be suppressed.

*HEADER-n. :nnn 'literal string'.*

### HISTORY-RECORD

This verb is combined with an IF statement to test the value of the RECORD-GROUP field.

A value of 'H' is a true condition, indicating that a History record is read. Use this in a batch report program in conjunction with the RUNREP Record Indicator field value of 'A' or 'H' when History record access is required.

*IF [NOT] HISTORY-RECORD*

## I

### INITIAL-PRINT-LINE

This verb moves spaces to the first 135 positions of Pointer 8. Report programs use this area to format a print line.

*INITIAL-PRINT-LINE*

## L

### LABOR-RECORD

This verb is combined with an IF statement to test the value of the RECORD-GROUP field.

A value of 'L' is a true condition, indicating that a Labor record is read. Use this in a batch report program in conjunction with the RUNREP Record Indicator field value of 'A' or 'L' when Labor record access is required.

*IF [NOT] LABOR-RECORD*

### LEFT-JUSTIFY

This instruction is used to compress the space allowed for a numeric or computational field in edited format. This verb is used in conjunction and must precede the PRINT verb and the data's fieldname to be compressed. The SCREEN area is manipulated to accomplish the data compression.

*LEFT-JUSTIFY. PRINT field-name.*

### LINE-ADVANCE

The LINE-ADVANCE command is used in -RP programs to define the number of blank lines you want to print for an output print line (carriage control).

There are two special LINE-ADVANCE values:

99 - suppresses the output print line.

00 - forces a top of page prior to printing the output print line. Headings are NOT repeated.

If this command is used, you must also account for the lines added by calculating LINE-COUNT.

The special 99 value can be used to eliminate duplicate total lines for an employee.

#### LINE-COUNT

This field is used by the batch reporting programs to keep count of output lines. This field must be re-calculated whenever the LINE-ADVANCE command is used. This is done so that print lines per page remains accurate and page breaks happen correctly.

### M

#### MATCH-PLAN-ID

The MATCH-PLAN-ID verb is used in conjunction with the FIND STARTING WITH ... routine. The MATCH-PLAN-ID verb shortens the length of a FIND search to the data field's segment code and Plan ID. Instead of finding an exact match, the system finds the first occurrence for the Plan ID, regardless of the contribution type, fund or other key information.

MATCH-PLAN-ID.

#### MATCH-SEGMENT-

The MATCH-SEGMENT-CODE verb is used with the FIND ... STARTING CODE WITH ... routine.

The verb shortens the length of a FIND search to the data field's segment code. Instead of finding an exact match, the system finds the first occurrence of the segment code. The system matches the segment code key without regard for the date used in the search argument. The search argument requested by the FIND statement includes the segment type, segment code, and a century date (LZF212C17).

The MATCH-SEGMENT-CODE verb shortens this to just the segment and segment code (LZF).

This routine is good for locating data to be used in an 'as of' situation or to make sure you have found an occurrence of the desired segment.

MATCH-SEGMENT-CODE

#### MOVE

The MOVE verb transfers data from one field to another.

■ In these cases, MOVE transfers data based on the length of the receiving field:

- The sending and receiving fields are the same length.
- The sending field is longer than the receiving field.
- The sending and receiving fields are numeric fields. With numeric fields, MOVE also aligns decimal points and converts data from one numeric form to another, when necessary.
- If the sending field or literal value is shorter than the receiving field, MOVE transfers data based on the location of the sending field.

In other words, you must know what pointer defines the sending field.

These rules apply:

- For fields in Pointers 1 - 17 and Pointer 40, the length of the receiving field governs the amount of data moved. The receiving field is completely filled by the operation. This means that MOVE transfers the sending field and as many bytes that follow as needed to fill the receiving field.
- For fields in Pointers 18 - 39 and 41 - 63, MOVE transfers the sending field and leaves any remaining data in the receiving field intact.

- Review the Field Name Table entry if you are unsure what pointer a field belongs to. All company, tax, employee, and other Master Record fields are defined in Pointers 21 - 39. Literal values are defined in Pointer 10.
- In MOVE operations that involve the fields WORK, SCREEN, or PRINT-FIELD, the amount of data transferred is always determined by the length of the other field involved.

MOVE {literal field-name-1figurative-constant} TO fieldname2.

#### MOVE-PLACE-TO- HOLDn

This verb moves the value of P-E-PLACE (the current position within a record or work area of PTR36) into Pointer 06. The length of Pointer 06 hold area is 4, 5 or 6 based on how your machine stores binary data.

This verb allows you to access several L segments and return to the position or segment of origin.

There are 8 hold areas, where n equals 1 through 8.

MOVE-PLACE-TO-HOLD-1.

See also RESET-TO-HOLDn-PLACE.

## N

#### NEW-PAGE-AFTER

Command used in coding batch reports. It is a sort key option placed after a control break field in the sort paragraph.

NEW-PAGE-AFTER is used when you want the printer to advance to a new page after printing the control break for the field.

It is only valid when used PRIOR to the SORT-LENGTH-nn command.

{*PRINT OUTPUT*} field-name NEW-PAGE-AFTER.

**NEW-PAGE-BEFORE**

Command used in coding batch reports. It is a sort key option placed after a control break field in the sort paragraph.

NEW-PAGE-BEFORE is used when you want the printer to advance to a new page before printing the control break for the field.

It is only valid when used PRIOR to the SORT-LENGTH-nn command.

*{PRINT OUTPUT} field-name NEW-PAGE-BEFORE.*

**NO-CONTROL-1-2**

This is a DEFINE-REPORT option that suppresses the printing of Control 1-2 headings on reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-CONTROL-1-2.*

**NO-DATE-OR-TIME**

This is a DEFINE-REPORT option that suppresses the printing of the date and time on the second heading line of reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-DATE-OR-TIME.*

**NO-HEADER-2**

This is a DEFINE-REPORT option that suppresses the printing of the second line of standard headings on reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-HEADER-2.*

**NO-HEADER-3**

This is a DEFINE-REPORT option that suppresses the printing of the third heading line on reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-HEADER-3.*

**NO-HEADER-4**

This is a DEFINE-REPORT option that suppresses the printing of the fourth heading line on reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-HEADER-4.*

NO-HEADINGS

This is a DEFINE-REPORT option that suppresses the printing of standard headings on reports. NO-HEADINGS can be used when a report is to print labels or other reports that do not need headings.

See DEFINE-REPORT.

*DEFINE-REPORT NO-HEADINGS.*

NO-PAGE-NUMBER

This is a DEFINE-REPORT option that suppresses the printing of a page number on the first heading line of reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-PAGE-NUMBER.*

NO-PE-DATES

This is a DEFINE-REPORT option that suppresses the printing of the Period End date literals on the first and second heading lines on a report.

See DEFINE-REPORT.

*DEFINE-REPORT NO-PE-DATES.*

NO-TIME

This is a DEFINE-REPORT option that suppresses the printing of the time on the second heading line of reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-TIME.*

NO-VERSION-NUMBER

This is a DEFINE-REPORT option that suppresses the printing of file version number on the first heading line of reports.

See DEFINE-REPORT.

*DEFINE-REPORT NO-VERSION-NUMBER.*

NO-PRINT-GRAND-TOTAL

This command is used in coding batch reports. It is a sort key option placed after the Forms/Report Code when no report control break is being performed in the sort paragraph.

It must follow the Forms and Report Code and is only valid when used PRIOR to the SORT-LENGTH-nn command.

*{PRINT OUTPUT} field-name NO-PRINT-GRAND-TOTAL*

**NO-PRINT-SUBTOTAL**

The command is used in coding batch reports. It is a sort key option placed after a field in the sort key that should not produce a control break.

It is only valid when used prior to the SORT-LENGTH-nn command.

*{PRINT OUTPUT} field-name NO-PRINT-SUBTOTAL.*

**O****ONE-SPACE**

This verb is used to compress the space between two fields. The verb is used in conjunction with the PRINT verb and follows the PRINT verb and the field to be compressed.

No other fields or literals can be included in the PRINT statement when ONE-SPACE is used.

It is used for alphanumeric data.

The SCREEN area is manipulated by positioning in the output area to the position where the first field ends. A space is moved to the output position pointer to separate the compressed field from the next field in the output position area.

*PRINT field-name ONE-SPACE.*

**OUTPUT**

The OUTPUT verb moves field data or literal values to Pointer 11 (SCREEN).

OUTPUT moves data to Pointer 11 without editing or converting it in any way.

*OUTPUT {literal field-name-1} [literal field-name-2].*

**P****PERFORM**

The PERFORM verb transfers control to a designated paragraph. Control is returned to the statement that follows the PERFORM statement when the program executes the EXIT verb.

- Do not use the RETURN verb to end a PERFORM statement.
- You can nest PERFORM statements up to six levels deep.

*PERFORM Pnnn- [label].*

**PRINT**

The PRINT verb moves field data and literal values to Pointer 11 (SCREEN). Unlike the OUTPUT verb, the PRINT verb edits fields according to the edit length and edit routine specified on the Field Name Table.

- PRINT does not edit fields unless they have a specific edit length and edit pattern.
- You can use PRINT to cause field headings and spaces between fields to display on a form.

*PRINT {literal field-name-1} [literal field-name-2].*

#### PRINT-GRAND-TOTAL

Command used in coding batch reports. It is a sort key option placed after the FORMS/REPORT-CODE literal to produce a report control break.

For a report with totals, it is the highest level of totaling.

It is only valid when used PRIOR to the SORT-LENGTH-nn command.

*{PRINT OUTPUT} field-name PRINT-GRAND-TOTAL.*

#### PRINT-REPORT

PRINT-REPORT is a verb used in an -RP programs. It reads records into a table so the -RP paragraphs can be performed.

This verb must be the first command in the -RP program.

*PRINT-REPORT.*

#### PROCESS

The PROCESS verb enables you to establish a process loop.

A process loop is a series of statements that are executed repetitively. Process loops are an effective way to check multiple-occurrence segments, because the PROCESS logic is executed for each occurrence of a particular segment code.

- If you do not specify an optional qualifier, the PROCESS verb begins at the first occurrence in the stack and continues until it processes all occurrences.
- Do not execute a RETURN verb in a process loop.
- You can use the BYPASS-ENTRY verb in a process loop. This verb causes processing to advance to the next occurrence.

*PROCESS - non-key-field-1[FROM HERE STARTING WITH {literal key-field-1} ENDING WITH{literal key-field-2}]*

*imperative statement*

*END-PROCESS.*

## **R**

#### READ-TA-TABLE

READ-TA-TABLE is a verb used to locate a TA Job Table record for the Job Code/Job Code Extent recorded on the employee's ZD segment. This verb may be used online or in batch.

The VERB first determines the date to be used in the Table Read. It checks to see if there is a more recent ZD on file for the employee; if so, it subtracts one day from the Effective Date of that ZD segment and uses the resulting date as the date used in the following determinations.

If this is the most current ZD on file, the VERB uses today's date. This insures you use the TA Job Table record that is in effect as of the 'last' day the employee is in a Job. Thus, you have the most current Job information; Salary Grade, Title, and so forth, for reporting or calculations.

The VERB then finds the AX TZ Table record using the Control 1-2 present on the ZC segment on file as of the date just identified, or if not present, the Control 1-2 from the Command Line and the date identified.

It takes the Control Number from the AX TZ Table Record, the Job Code and Job Code Extent from the respective fields on the form or the ZD segment, and the date identified to build the Key to read the TA Job Code Table.

The system looks for a match to the Key that has been built. If an exact match cannot be found, the system tries to locate the next highest Table record by the date. If one is not found, the system will return a value in the STAT-KEY field that is greater than '01'.

Check this field for a value greater than '01' and if it is, use PRINT-ERROR 'HRW004' on your screen to indicate there is no TA record on file for this Job Code/Job Code Extent requested or one was not in effect as of the date.

READ-TA-TABLE.

READ-TK-TABLE

This verb reads a TK Table record from the Control File based on an BATCH employee's Q1 or Q4 Benefits Plan enrollment segment data.

The date from the report's RUNREP input parameter is used to find the 'As of' employee level segment for the Table lookup.

READ-TK-TABLE-BATCH

RESET-MY-A8-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-A8-PLACE verb is used with the SAVE-MY-A8-PLACE verb. It moves the value of P-E-HOLD into P-E-PLACE.

See SAVE-MY-A8-PLACE.

RESET-MY-A8-PLACE.

RESET-MY-FREQ-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-FREQ-PLACE verb is used with the SAVE-MY-FREQ-PLACE verb. It moves the value of P-E-HOLD into P-E-PLACE.

See SAVE-MY-FREQ-PLACE.

RESET-MY-FREQ-PLACE.

RESET-MY-HED-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-HED-PLACE verb is used with the SAVE-MY-HED-PLACE verb. It moves the value of P-E-HOLD to P-E-PLACE.

See SAVE-MY-HED-PLACE.

RESET-MY-HED-PLACE.

RESET-MY-LABOR-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-LABOR-PLACE verb is used with the SAVE-MY-LABOR-PLACE verb. It moves the value of P-E-HOLD into P-E-PLACE.

See SAVE-MY-LABOR-PLACE.

RESET-MY-LABOR-PLACE.

RESET-MY-NAME-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-NAME-PLACE verb is used with the SAVE-MY-NAME-PLACE verb. It moves the value of P-E-HOLD into P-E-PLACE.

See SAVE-MY-NAME-PLACE.

RESET-MY-NAME-PLACE.

RESET-MY-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-PLACE verb.

The RESET-MY-PLACE verb is used with either the SAVE-MY-PLACE (pointer 36) or SAVE-MY-WORK-PLACE (pointer 14) verbs. It moves the value stored in P-E-HOLD to P-E-PLACE.

See SAVE-MY-PLACE and SAVE-MY-WORK-PLACE.

RESET-MY-PLACE.

RESET-MY-TAX-PLACE

This verb is used to reset a segment pointer to a value that was stored in a hold field (P-E-HOLD) by a SAVE-MY-xx-PLACE verb.

The RESET-MY-TAX-PLACE verb is used with the SAVE-MY-TAX-PLACE verb. It moves the value of P-E-HOLD into P-E-PLACE.

See SAVE-MY-TAX-PLACE.

RESET-MY-TAX-PLACE.

RESET-TO-HOLD<sub>n</sub>-PLACE

This verb moves the value stored in Pointer 06 by the PLACE MOVE-PLACE-TO-HOLD verb back into P-E-PLACE for Pointer 36.

It is useful when accessing more than two segment codes (for example, ZA, ZD, ZF and ZG) at a time, and returning to the original segment or position processed.

See MOVE-PLACE-TO-HOLD1.

This verb allows you to reset to several L segments, returning to the position or segment of origin. There are eight hold areas, where n equals 1 through 8.

RESET-TO-HOLD<sub>n</sub>-PLACE

---

**RETURN**

The RETURN verb transfers control back to the calling program.

- A RETURN statement is implied at the end of every program, whether it is coded or not.
- When a program executes a RETURN statement, it returns control to the next logical statement after the CALL statement in the calling program.
- The highest-level program is called by CYB90-.
- Do not code the RETURN verb in a process loop.

**RETURN****S****SAVE-MY-A8-PLACE**

This verb holds your position within the stacked segment indicated. The SAVE-MY-A8-PLACE verb holds your current position within the B (Company HED) segment. The position is restored using the RESET-MY-A8-PLACE verb.

**SAVE-MY-A8-PLACE****SAVE-MY-FREQ-PLACE**

This verb holds your position within the stacked segment indicated. The SAVE-MY-FREQ-PLACE verb holds your current position within the AJ (Company Pay Frequencies) segment. The position is restored using the RESET-MY-FREQ-PLACE verb.

**SAVE-MY-FREQ-PLACE****SAVE-MY-HED-PLACE**

This verb holds your position with the stacked segment indicated. The SAVE-MY-HED-PLACE verb holds your current position within the H (EmployeeHED) segment. The position is restored using the RESET-MY-HED-PLACE verb.

**SAVE-MY-HED-PLACE****SAVE-MY-LABOR-PLACE**

This verb holds your position with the stacked segment indicated. The SAVE-MY-LABOR-PLACE verb holds your current position within the G (Employee Labor Location) segment. The position is restored using the RESET-MY-LABOR-PLACE verb.

**SAVE-MY-LABOR-PLACE****SAVE-MY-NAME-PLACE**

The SAVE-MY-NAME-PLACE verb holds your current position within the F (Employee Name and Address) segment. The position is restored using the RESET-MY-NAME-PLACE verb.

**SAVE-MY-NAME-PLACE**

## SAVE-MY-PLACE

This verb holds your position within the stacked segment indicated. The SAVE-MY-PLACE verb holds your current position within the L (Human Resource) segment. The position is restored using the RESET-MY-PLACE verb.

### SAVE-MY-PLACE.

## SET

SET is a verb used to change the value of an index dataname. It is used in conjunction with the words UP, DOWN, FIRST, and TO.

For example: WORK is an index dataname. If the current position of WORK is five, the statement SET WORK UP changes the position of WORK to six; the statement SET WORK DOWN changes the position of WORK to four; the statement SET WORK TO FIRST changes the position of WORK to one, and the statement SET WORK TO :2 changes the position of WORK to 2.

All positioning using SET is counted relative to one (1).

SET can also be used to position a specific pointer to a specified occurrence of a stacked record. The SET verb must be followed by a non-key field. SET-CO-PTRS- TO-1ST SET-EMP-PTRS- TO-1ST SET-PAGE-NBR-TO-1.

SET fieldname {TO {:nnn SAVE FIRST} UP DOWN}

### SET-EMP-PTRS-TO-1ST

All Employee level pointer address values are set to the first occurrence. These pointers are: 29, 30, 31, 32, 33, 34, 35, 36 and 37.

After executing this verb, a program has access to the first occurrence of the given segment, that is, HED 001 (H segment) without issuing a FIND or PROCESS verb.

### SET-EMP-PTRS-TO-1ST.

### SET-CO-PTRS-TO-1ST

All Company level pointer address values are set to the first occurrence. These pointers are: 22, 23, 24, 25, 26 and 27.

After executing this verb, a program has access to the first occurrence of the given segment, that is, HED 001 (B segment) without issuing a FIND or PROCESS verb.

### SET-CO-PTRS-TO-1ST.

### SET-PAGE-NUMBER-TO-1

This command is used in coding batch reports. It is a sort key option placed after control break field name performed in the sort paragraph.

SET-PAGE-NUMBER-TO-1 is used when you want the page to begin re-numbering from 1 after printing total(s) for the control break field.

It is only valid when used prior to SORT-LENGTH-nn command.

### DEFINE-REPORT SET-PAGE-NUMBER-TO-1.

## SORT-LENGTH-nn

SORT-LENGTH-nn is a command used in batch report programming to designate the length of the sort portion of the extract record.

A sort length of 11 is the minimum length allowed and sorts by the Forms and Report Code and Control 1-2. Valid sort lengths are between 11-60. Each sort length has an associated verb, that is, SORT-LENGTH-11, SORT-LENGTH-21.

The SORT-LENGTH-*nn* designation is used by the print process to determine where the sort key data ends in the extract record for a given report.

*{PRINT OUTPUT} fieldname sort key option SORT-LENGTH-*nn**

#### SPACE-EXTRACT-RECORD

This verb is used in batch programs to move spaces to the 150 character extract work area to extract data for the report. This command is used when detail lines vary in length.

*SPACE-EXTRACT-RECORD.*

#### SPACE-OVER

The SPACE-OVER verb moves the specified number of spaces to Pointer 11 (SCREEN). SPACE-OVER must be followed by a colon (:) and a two digit numeric literal in the 01 through 60 range.

*SPACE-OVER :*nn**

### **T**

#### TRIPLE-AFTER-HDNCS

TRIPLE-AFTER-HDNCS is a DEFINE-REPORT option that adds three blank lines after report headings, instead of the default of two blank lines.

See DEFINE-REPORT.

*DEFINE-REPORT TRIPLE-AFTER-HDNCS.*

#### TRIPLE-SPACE-AFTER

This command is used in coding batch reports. It is a sort key option placed after a control break field name performed in the sort paragraph.

TRIPLE-SPACE-AFTER is used when you want the printer to advance three lines after printing the control break field.

It is only valid when used PRIOR to the SORT-LENGTH-*xx* command.

*{PRINT OUTPUT} field-name TRIPLE-SPACE-AFTER.*

#### TRIPLE-SPACE-BEFORE

This is the command used in coding batch reports. It is a sort key option placed after control break field name being performed in the sort paragraph.

TRIPLE-SPACE-BEFORE is used when you want the printer to advance three lines before printing the control break field.

It is only valid when used PRIOR to the SORT-LENGTH-xx command.

*{PRINT OUTPUT} field-name TRIPLE-SPACE-BEFORE.*

### U

#### UPDATE-DATE-TIME

The UPDATE-DATE-TIME verb macro is used to access the exact date and time found in the system maintained fields CURRENT-DATE and CURRENT-TIME.

Without issuing this verb, the contents of the CURRENT-DATE and CURRENT-TIME fields are the initial start date and time when the program began executing.

*UPDATE-DATE-TIME.*

### W

#### WRITE

WRITE is a verb used to output a record from any given file. It is followed by a valid CBSVO or CBSVB file name, that is, WRITE FILE10. The data in Pointer 8 (W8-) is written to the output file named for the appropriate length of the output file record.

*WRITE file-name.*

#### WRITE-EXTRACT

WRITE-EXTRACT is a verb used to write a 150-character extract record for a report.

The WRITE-EXTRACT verb must be the last logical command in the report source code.

The WRITE-EXTRACT verb moves extract record data to displacement 1601 of the SCREEN area and places it in FILE15 so that it can be sorted and printed.

*WRITE-EXTRACT.*

#### WRITE-FILE10

WRITE-FILE10 is a verb used in batch report programming to write an 80-character record to FILE10. The record is written from the SCREEN Work area.

Data is moved to this area using PRINT, OUTPUT, or SPACE-OVER :nn verbs.

*WRITE-FILE-10.*



---

**NOTES**

---

5M-RP program.....	231	FIND- verbs .....	338
Accessing history and labor.....	203	First time indicator.....	169
Accessing table data.....	191	FMLEDT.....	213
Adding a report group.....	17	Forms code.....	85
Adding reports to a report group.....	19	G segment—employee labor splits .....	318
Additional heading lines .....	241	H segment—employee HEDs .....	319
ALLOCATE .....	165	H segment—employee HEDs (labor/history record).....	322
Alternating pointer addresses.....	185	HEADER-n .....	79
Basic verbs.....	171	Heading literals .....	79
Batch job initiator .....	35	Heading overrides .....	209
Batch Job Initiator program (SUBMIT) ...	33, 35	Heading record are ready to print .....	239
Benefits tables.....	340	History and labor records.....	203
Carriage control .....	93	HR base module tables.....	339
Changes to sort logic.....	131	<b>IF HISTORY-RECORD</b> .....	207
Consolidating organization numbers (Control 1- 2s).....	149	<b>IF LABOR-RECORD</b> .....	207
Control break descriptions .....	147	<b>IF PERMANENT-MASTER</b> .....	207
Date calculations.....	179	Initialization .....	73, 81
Date fields .....	173	Initiate a report.....	35
DEFINE-REPORT options .....	77	Initiate reports .....	33
<b>DEFINE-REPORT</b> verb.....	75	J segment—employee taxes.....	323
Defining the report.....	75	J segment—employee taxes (history record)	330
Deleting a report group.....	19	JRPTxxxx script.....	41
Detail data with matching sort key is ready..	243	L segment—employee location .....	335
detail line is ready to print .....	247	L segment—employee salary.....	334
E segment—basic employee.....	308	L segment—employee spouse/dependent.....	331
Editing a report group.....	19	L segment—employee training.....	336
Employee counts.....	133	L segment—employment status.....	332
Employee selection .....	73	Labor and history access.....	207
Enhanced Reporting features .....	47	Labor and history verbs .....	207
Entering report group parameters .....	21	Launch toolbar access.....	53
environments		Limit Companies form (RPTLC).....	51
specifying.....	63	Line advancing.....	133
Error Message .....	59	Line counts and numbering.....	233
Error messages .....	31	Master tables .....	191
Establish a report group .....	17	Multiple extract.....	151
Example code and extract.....	93	Naming the report .....	75
Execute Report Group form (RPTSCR).....	51	New organization number (Control 1-2) is starting .....	237
EXMR0 example .....	115	NO-PRINT-GRAND-TOTAL .....	131
Extract program flow .....	73	Online process monitoring.....	53
Extract record.....	83	Online reporting .....	47
Report portion.....	83	online reporting process.....	49
Sort portion .....	83	Online viewing of completed reports.....	61
Writing the record.....	95	Optional Report Schedules Control form (C12RPT).....	25
Extract record files—special print options technique 1 .....	303	Organization report validation .....	25
Extract record files—special print options technique 2.....	305	OUTPUT.....	89
Extract record files—the report extract step.	301	P100-P. ....	235
Extract sort.....	29	P120-P. ....	237
Extract techniques.....	301	P140-P. ....	239
F segment—employee name and address .....	316	P160-P. ....	241
FIND .....	175	P180-P. ....	243

P200-P.....	245	SAVE/RESET pointer 36 .....	189
P220-P.....	247	Segment/Pointer manipulation.....	183
P240-P.....	249	Segments and pointers .....	183
P260-P.....	251	Select and print routine .....	185
Packaged report features.....	13	Selecting an organization.....	21
Parameter entry form .....	167	Selection.....	81
Parameter Selection form (RPARMS).....	23	SET-CO-PTRS-TO-1ST .....	81
Period record.....	209	SET-EMP-PTRS-TO-1ST.....	81
Pointer 6.....	167	Setting report parameters .....	23
PRINT .....	89	Sort basics .....	131
Print files.....	31	Sort field limit.....	147
Print position and the report output .....	113	Sort key .....	85
Print Position Compile (RETYPE) Error		Sort key definition .....	73
Resolution .....	342	Sort key length .....	141
Print position record preassignment .....	245	Sort key length and placement.....	141
Print position records.....	107	Sort key logic .....	151
Print positions and the extract record .....	111	Sort key options .....	85, 131
PRINT-GRAND-TOTAL.....	131	Sort key placement.....	143
PROCESS .....	177	Sort key separator .....	153
Process Monitor utility.....	55	Sort length.....	87
Process to view a report.....	49	sort level.....	131
Record group.....	203	SPACE-EXTRACT-RECORD.....	81
RELOAD .....	131	SPACE-OVER :nn.....	89
Report code.....	85	Special print option.....	227
Report data.....	73, 95	development.....	229
Report definition .....	73	name.....	229
Report detail, totals, and printing.....	107	paragraphs.....	231
Report development.....	15	program.....	229
Report disposition .....	63	work areas .....	233
Report extract.....	27	Special sort keywords .....	131
Report Extract program (REPORT).....	27	Special sort techniques.....	147
Report Group Activities form (RGMSTR)17, 51		Standard headings .....	75
Report headings .....	73	Submitting, routing, and viewing reports .....	33
report is done .....	251	Summary reporting .....	211
Report navigation.....	45	Table records.....	191
Report print.....	29	Table rules.....	191
Report Print Position Compile program		Table verbs.....	193
(RETYPE).....	121	Title and column headings .....	77, 79
Report print position edit .....	109	Total data is ready.....	245
Report Print program (RTPRNT) .....	29	total line is ready to print .....	249
Report special formatting.....	213	TOTALS-ONLY .....	211
Report steps.....	15	Update date and time .....	213
Report timing .....	169	Use of MOVE .....	173
Report totals.....	133	Using checklists to launch a report.....	53
Reserved word .....	179	Using REPORT.....	27
RETURN verb .....	81	Using the Navigator or menu to launch a report	
RETYPE .....	131	.....	51
RGMSTR parameters.....	163	Variable report headings.....	209
Route reports.....	33	View a report .....	45, 59
RTEDIT naming requirement .....	109	View held report .....	43
RTPRNT phase .....	31	View Held Report program (VIEW).....	43
Run-time input parameters.....	163	View reports.....	33
Save/Reset.....	187	View the process log.....	57
SAVE/RESET pairs .....	189	Viewing report and query output.....	43

When field names do not match..... 119  
Work fields..... 119, 147

Write extract record ..... 73

---

**NOTES**

---

# Hewitt

---

## **Using Solution View Participant's Guide**

**Document Issue: 5.1**

Document Issue Status: Eleventh Release  
Document Issue Level: 5.1  
Document Issue Date: April 2004  
Software Version: 5.1

### **Copyright Notice**

Copyright© 2004 by Hewitt Associates LLC. All rights reserved.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series®  
are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear,  
but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

---

## TABLE OF CONTENTS

---

<b>SECTION 1: COURSE OVERVIEW .....</b>	<b>1</b>
COURSE OVERVIEW .....	3
COURSE LOGISTICS .....	5
COURSE MATERIALS .....	7
<b>SECTION 2: SOLUTION VIEW CAPABILITIES AND FEATURES .....</b>	<b>9</b>
SECTION OVERVIEW .....	11
SOLUTION VIEW CAPABILITIES .....	13
SOLUTION VIEW FEATURES .....	15
SECTION SUMMARY .....	17
SECTION EXERCISE.....	19
<b>SECTION 3: THE ACTION FORM—THE STARTING POINT.....</b>	<b>21</b>
SECTION OVERVIEW .....	23
THE ACTION FORM .....	25
SECTION SUMMARY .....	35
SECTION EXERCISE.....	37
<b>SECTION 4: CREATING A PROGRAM TO GENERATE AN ONLINE REPORT .....</b>	<b>39</b>
SECTION OVERVIEW .....	41
PLANNING AHEAD: W.I.I.F.M. ....	43
THE QUERY WRITER FORM SERIES .....	51
SELECTING THE ACTION .....	53
QUERY OPTIONS FORM.....	55
DEFINING THE FIELDS .....	57
SELECTING SEGMENT KEYS.....	67
CREATING SELECTION CRITERIA LOGIC .....	73
DISPLAYING THE REPORT.....	85
REPORT SAMPLE .....	93
INQUIRY, CHANGING, COPYING, DELETING.....	95
SECTION SUMMARY .....	109

SECTION EXERCISE.....	113
<b>SECTION 5: CREATING A PROGRAM TO GENERATE A BATCH REPORT.....</b>	<b>117</b>
PLANNING AHEAD: W.I.I.F.M.....	121
THE REPORT WRITER FORM SERIES.....	129
REPORT OPTIONS FORM.....	133
DEFINING THE FIELDS .....	135
SELECTING SEGMENT KEYS.....	155
DEFINING CALCULATION ROUTINES.....	161
CREATING SELECTION CRITERIA LOGIC.....	167
REPORT GROUP ACTIVITIES.....	179
BATCH REPORT PROCESS .....	197
SUBMITTING AND VIEWING REPORTS .....	199
INQUIRY, CHANGING, COPYING, DELETING.....	215
SECTION SUMMARY .....	229
SECTION EXERCISE.....	233
<b>SECTION 6: CREATING A PROGRAM TO EXTRACT DATA .....</b>	<b>237</b>
SECTION OVERVIEW .....	239
PLANNING AHEAD: W.I.I.F.M.....	241
THE EXTRACT WRITER FORM SERIES .....	250
SELECTING THE ACTION .....	252
EXTRACT OPTIONS FORM.....	254
DEFINING THE FIELDS .....	257
SELECTING SEGMENT KEYS.....	276
DEFINING CALCULATION ROUTINES.....	282
CREATING SELECTION CRITERIA LOGIC.....	288
OUTPUT OPTIONS.....	300
INQUIRY, CHANGING, COPYING, DELETING.....	306
SECTION SUMMARY .....	320
SECTION EXERCISE.....	324
<b>SECTION 7: CREATING A PROGRAM TO ENTER DATA BY USING A FORM.....</b>	<b>328</b>
SECTION OVERVIEW .....	330
PLANNING AHEAD: W.I.I.F.M.....	332
THE FORM WRITER FORM SERIES.....	338
SELECTING THE ACTION .....	340
DEFINING THE FIELDS .....	342
CREATING HELP FORM TEXT.....	350
ACCESSING THE NEW FORM .....	352
INQUIRY, CHANGING, COPYING, DELETING.....	358
SECTION SUMMARY .....	370
SECTION EXERCISE.....	372

---

<b>SECTION 8: CREATING A PROGRAM TO DEFINE NEW FIELDS AND CREATE AN ENTRY FORM .....</b>	<b>376</b>
SECTION OVERVIEW .....	378
PLANNING AHEAD: W.I.I.F.M. ....	380
THE NEW FIELDS DEFINITION FORM SERIES .....	388
SELECTING THE ACTION .....	390
TITLING AND DESCRIBING THE NEW FORM .....	392
DEFINING THE FIELDS .....	396
ACCESSING THE NEW FORM .....	400
NEW FORM SAMPLE .....	402
DELETING THE NEW FIELDS DEFINITION FORM .....	404
SECTION SUMMARY .....	406
SECTION EXERCISE.....	408
<b>APPENDIX A: EXERCISE ANSWERS.....</b>	<b>413</b>
SECTION 3 EXERCISE ANSWERS.....	414
SECTION 4 EXERCISE ANSWERS.....	415
SECTION 5 EXERCISE ANSWERS.....	419
SECTION 6 EXERCISE ANSWERS.....	428
SECTION 7 EXERCISE ANSWERS.....	431
SECTION 8 EXERCISE ANSWERS.....	433
<b>APPENDIX B: EXTRA FOR EXPERTS .....</b>	<b>437</b>
BATCH REPORTING VALIDATION.....	439
DISPLAYING A REPORT SCHEDULE.....	443
PURGING A REPORT SCHEDULE.....	445
PURGING AN ORGANIZATION SCHEDULE.....	447
CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION.....	449
THE SINGLE FIELD VIEW FORM.....	471
FIELD NAME CROSS REFERENCE FIELD HELP.....	473
MERGING/IMPORTING TEXT FILES.....	475
CONSOLIDATED AND ROLL-UP REPORTING.....	477
ENTERING HEADING INFORMATION .....	479
DESIGNATING THE CONSOLIDATED ORGANIZATION.....	481
USING ROLL-UP AND CONSOLIDATED REPORTING.....	483
CREATING A USER QUICK-REFERENCE FIELD MENU .....	484
ACCESSING A USER QUICK-REFERENCE FIELD MENU .....	489
DELETING ENTRIES FROM THE USER QUICK REFERENCE FIELD MENU.....	491
CHANGING OPERATOR TITLES FOR QUICK REFERENCE FIELD MENU.....	492

<b>APPENDIX C: WORKSHEETS .....</b>	<b>493</b>
QUERY WRITER.....	494
REPORT WRITER.....	495
EXTRACT WRITER.....	496
FORM WRITER.....	497
NEW FIELDS DEFINITION WRITER.....	498
<b>GLOSSARY.....</b>	<b>499</b>

## SECTION 1: COURSE OVERVIEW

---

---

### TABLE OF CONTENTS

COURSE OVERVIEW.....	3
COURSE LOGISTICS .....	5
COURSE MATERIALS.....	7

## **Course Overview**

- **Purpose & Benefits**
- **Audience**
- **Prerequisites**
- **Goals**
- **Expectations**

---

**NOTES**

---

## COURSE OVERVIEW

**Purpose & Benefits** This course will familiarize you with several non-programming end-user tools for creating and maintaining your unique reporting requirements.

**Audience** This course is designed for the following audiences:

- Payroll and Human Resource personnel who use the system on a daily basis
- Anyone using The Solution Series to create custom forms, reports or data extracts
- Data Processing professionals responsible for satisfying end-user data requests

**Prerequisites** Individuals enrolled in the Using Solution View class must have attended one of the introductory courses to The Solution Series. To increase your success potential, it is also recommended that you attend at least one Solution Series application course.

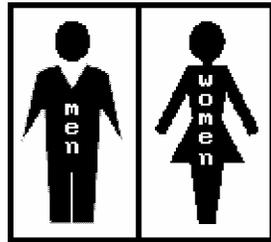
**Goals** At the conclusion of this course, you will be able to do the following:

- Convert a request into specifications for a program
- Recognize the components of Solution View
- Identify the Solution View forms that are used to create the program
- Select and use the Solution View program that will satisfy the original request
- Use the program to produce output
- Analyze the output to verify that it satisfies the request

**Expectations** To get the most out of this course you should do the following:

- Ask questions.
- Share examples of your own Cyborg-related experiences. This sharing of information among participants enhances the learning process.
- Ask where to obtain additional information if you have an interest in a point that is introduced. The Solution Series has a great deal of supplementary documentation.

## Logistics



---

NOTES

---

**COURSE LOGISTICS**

Use the space below and on the facing page to take notes about the course logistics.

**Meals**

---

---

---

**Breaks**

---

---

---

**Telephones**

---

---

---

**Restrooms**

---

---

---

**Security**

---

---

---

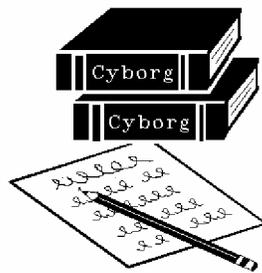
**Questions**

---

---

---

## Course Materials



---

**NOTES**

---

## COURSE MATERIALS

<b>Course Materials</b>	All illustrations and material reflect the system. Instructions for accessing or completing a form are provided.
<b>Table of Contents</b>	As you saw in the agenda, this course has eight sections. Each section is broken down into two or more subsections, which are called topics. Each section has a Table of Contents listing on the Title Page.
<b>Text Layout</b>	<p>This guide is designed in the following format:</p> <ul style="list-style-type: none"><li>■ Left pages contain illustrations. Below each illustration is an area entitled “Notes”. This area is left blank for your notes.</li><li>■ Information about each illustration appears on the right page.</li></ul>
<b>Section Overview</b>	Section content and class activities are introduced.
<b>Section Exercise</b>	Exercises give you an opportunity to practice what you have learned in each section. All sections except the Course Overview section have exercises.
<b>Appendixes</b>	<p>The appendixes are in the back of your Participant Guide. Appendixes contain the following:</p> <ul style="list-style-type: none"><li>■ <b>Exercise Answers</b>—Answers to section exercises.</li><li>■ <b>Extra for Experts</b>—Additional materials and activities for ambitious learners.</li><li>■ <b>Worksheets</b>—Blank worksheets which learners may find useful.</li></ul>
<b>Glossary</b>	An alphabetical listing of defined terms is provided after the last appendix.

---

**NOTES**

---

## SECTION 2: SOLUTION VIEW CAPABILITIES AND FEATURES

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW .....	11
SOLUTION VIEW CAPABILITIES .....	13
SOLUTION VIEW FEATURES .....	15
SECTION SUMMARY .....	17
SECTION EXERCISE.....	19

## Section Objectives

- **Identify the capabilities of Solution View**
- **Identify the features of Solution View**

---

NOTES

---

**SECTION OVERVIEW**

**Purpose**

In this section you will learn what you can expect from the Solution View product.

**Objectives**

Upon completion of this section you will be able to do the following:

- Identify the capabilities of Solution View
- Identify the features of Solution View

## Solution View Capabilities

- QUERY WRITER
  - REPORT WRITER
    - EXTRACT WRITER
      - FORM WRITER
        - NEW FIELDS  
DEFINITION WRITER
          - PC SELECTION  
WRITER

---

### NOTES

---

## SOLUTION VIEW CAPABILITIES

### **Solution View Definition**

Solution View is a user-friendly program writer. It is an online utility that walks you through the creation of new programs without the direct use of Cyborg Scripting Language (Cyborg's programming language). Solution View gives you access to several end-user tools through one main program.

Solution View replaces the Query writer, Report writer and Form writer from previous versions of The Solution Series. Programs created using the Query Maintenance Facility and the Report Maintenance Facility may be updated using Solution View.

### **Solution View Capabilities**

The Solution View utility program is the entry point from which you can access the six facilities that are used to create end-user programs. These are the six Solution View facilities:

- **Query Writer**—used to create programs that are run and display output online through the **Query** dialog box.
- **Report Writer**—used to create programs that are run in batch and produce printed reports.
- **Extract Writer**—used to create programs to generate a data file that may be input to another system.
- **Form Writer**—enables you to create your own employee level form using existing segments which contain delivered or user (previously defined) fields.
- **New Fields Definition Writer**—enables you to create your own company or employee entry forms using user-defined fields.
- **PC Selection Writer**—creates a program that supplies additional employee selection criteria to the PS-SEG information used to gather company/employee data. (This facility is discussed separately in the Extra for Experts appendix.)

## **Solution View Features**

- **The Action Form**
- **Option Field**
- **Field Name Table Menu**
- **Documentation**
- **Visiting Other Programs**

---

NOTES

---

## SOLUTION VIEW FEATURES

- The Action Form** Each of the Solution View facilities is accessed from one central Solution View form, the Action form. Once a facility is accessed through the Action form, all the forms for that facility are chained together automatically in an order determined by your entries. The program is created only when you have completely entered all the required forms (some forms are optional).
- Option Field** An Option field, which resides in the upper right corner of most forms in Solution View, allows you to perform independent tasks from the form you are currently using.
- Field Selection Menu** One of the features that makes Solution View easy to use is the Field Selection Menu. This feature allows you to select field names directly from a multiple level Field-Name Table Menu by typing an entry into the Option field, or by using the question mark (?) feature in the field name entry field.
- You have the option of storing your most commonly used field name options in the User Quick Reference menu option. You can also access the Quick Reference Fields menu option that Cyborg has already established for your use. See also Appendix B: Extra for Experts for more information.
- Using any of these Field-Name Table menus makes selection of fields easier and avoids spelling errors.
- Documentation** After you create a new Solution View program, you are encouraged to write documentation for the program. This documentation can be displayed online. It will inform future users about the program or refresh your own memory. Documentation and Help Facilities are detailed in course 2305—Introduction to Cyborg Scripting Language Programming or in menu SCHL00.
- Visiting Other Programs** When working in Solution View you may find that you need to reference data on another form. You may access other forms or programs and then return to where you were in the Solution View form chain at any time.

## Section Summary

- **Solution View Capabilities**
- **Solution View Features**

---

NOTES

---

**SECTION SUMMARY**

In this section, you learned what you can expect from the Solution View product. Specifically you learned:

■ Solution View Capabilities

---

---

---

■ Solution View Features

---

---

---

## Section 2 Exercise

---

NOTES

---

**SECTION EXERCISE**

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Define Solution View in your own words.

---

---

---

2. Which of the six Solution View facilities will you be using most frequently at your company?

---

---

---

3. Which of the Solution View features will you use the most to make your program writing easier?

---

---

---

---

**NOTES**

---

## **SECTION 3: THE ACTION FORM—THE STARTING POINT**

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW .....	23
THE ACTION FORM .....	25
SECTION SUMMARY .....	35
SECTION EXERCISE.....	37

## Section Objectives

- **Identify the Action form**
- **Define the fields on the Action form**
- **Access the Action form**

---

NOTES

---

## **SECTION OVERVIEW**

### **Purpose**

This section introduces you to the first form in Solution View. Since it is the starting point for all of the Solution View form series, we will discuss it here as a stand-alone topic. We will then incorporate it into the detail procedure for each Solution View facility.

We will follow the format of introducing the form and then give you a procedure for accessing it. Once accessed, we will define all the fields in sequence of appearance, and then give you a procedure for completing the form.

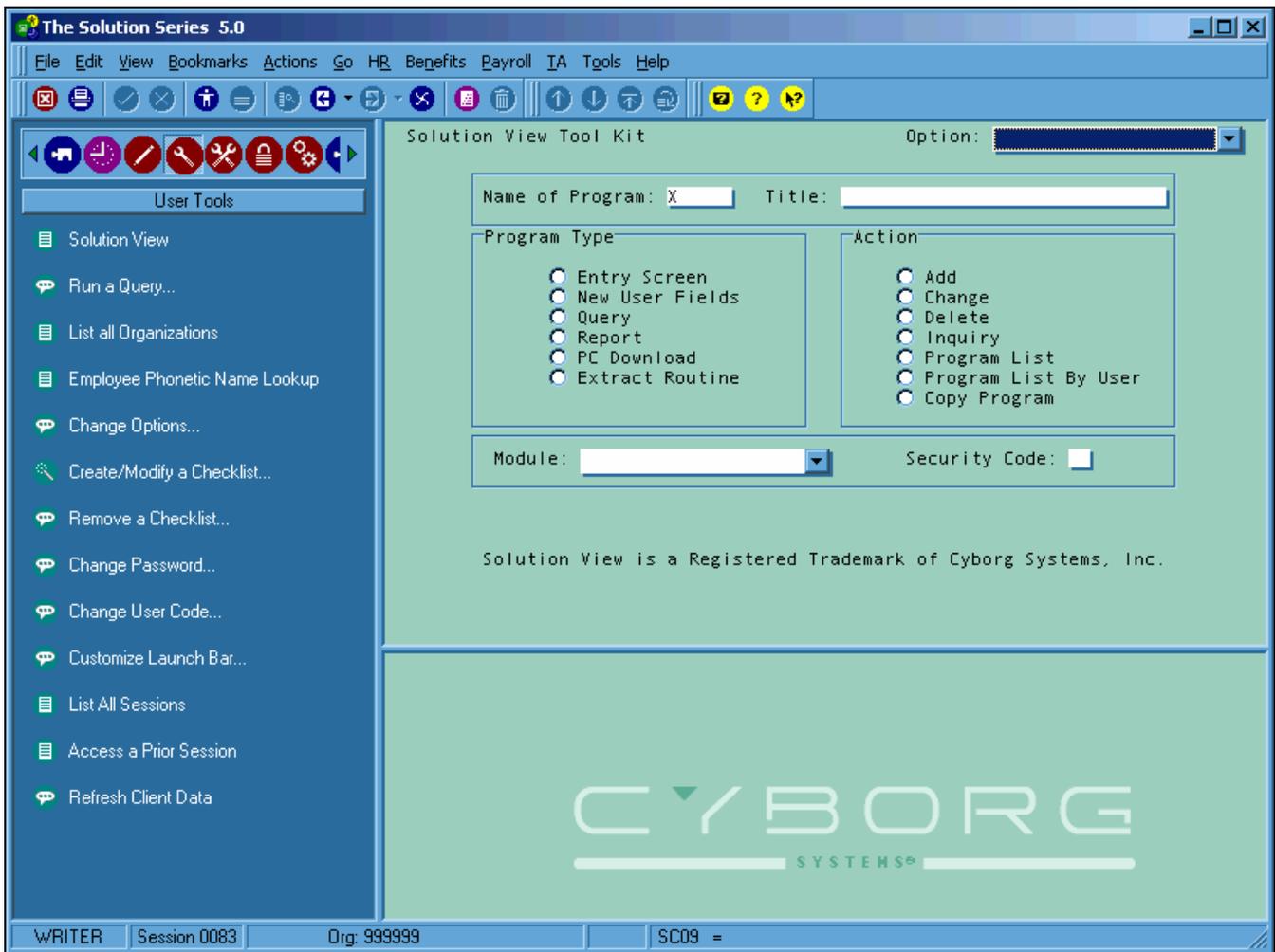
We have provided adequate time for hands-on activities. So, please listen while the instructor demonstrates, and then you will be given practice time as each step is completed.

### **Objectives**

When you have completed this section you will be able to do the following:

- Identify the Action form
- Define the fields on the Action form
- Access the Action form

## The Action Form



### NOTES

## THE ACTION FORM

**The Action Form** The Action form is the first form in Solution View. Regardless of the type of program you are writing, it is the common starting point for all of the Solution View form series. The Action form allows you to perform several different actions with program files.

**Accessing the Action Form** To access the form:

<u>Selection:</u>	<u>Step:</u>
<b>Navigator</b>	 User Tools
	User Tools
	 Solution View

**Result:** The Solution View Action form displays.

**Field Descriptions** **Name of Program**—the name of the program you are adding, changing, copying or deleting. You can use up to 6 alphanumeric characters. This is a required field.

All program names must begin with X to prevent accidental duplication of Cyborg program names. Some programs require added values at the end of this field. We will discuss those requirements later in this material. The report name must be unique if you are adding a program to the system.

**Title**—optional field that contains the title of your program. You can use up to 30 alphanumeric characters. This entry displays at the top of each report page or form.

Single quotes and apostrophes are *not allowed* in the title. If you are creating a Query Report or an Entry Form, brackets ( [ ] ), parentheses ( ), and greater-than and less-than signs (< >) are also not allowed.

## The Action Form, continued

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

---

NOTES

---

THE ACTION FORM, continued

Field Descriptions,  
continue

**Program Type**—required field that indicates the form series that will be used for this report. Valid Program Types are as follows:

- **Entry Form**—used to create your own employee level form using existing segments which contain delivered or user-defined fields.
- **New User Fields**—used to create your own company or employee entry forms using user-defined fields.
- **Query**—used to create programs that run online and display output through the QUERY capability.
- **Report**—used to create programs that run in batch and produce printed reports.
- **PC Download**—used to create a program that supplies additional employee selection criteria to the PS-SEG information used to gather company/employee data.
- **Extract Routine**—used to create programs to generate a data file that may be input to another system.

**Action**—required field that represents the action to be performed. Action codes allow you to Add, Change, Delete, or Copy your reports. Here are some helpful notes about the remaining codes:

- **Inquiry**—allows you to view in an Inquiry mode the forms you have entered to create a program. The Name of Program and Program Type must be entered when using this option.
- **Program List**—displays a menu of all programs created by all users. When using this option, you may place a partial program name in the Name of Program field. For example, placing XP in the field would list all programs beginning with the letters XP. If no programs exist beginning with XP, then a message is returned informing you there are no programs on file.
- **Program List By User**—displays a menu of the names of the programs created by the specific user. When using this option, the same partial program name search is available in the Name of Program field.

**Note:** The New Fields Definition Writer can only be accessed through the Add and Delete options.

## The Action Form, continued

Module: <input type="text"/>	Security Code: <input type="text"/>
------------------------------	-------------------------------------

Module	
Benefits Admin	(BA)
COBRA Administration	(CS)
Distributed Administration	(DS)
EEO	(EO)
Employee Attendance	(TA)
Employee Development	(MP)
Employee Relations	(ER)
Employee Resourcing	(AT)
Health & Safety	(HS)
HRMS	(HR)
Payroll	(PR)
Payroll/HRMS	(PP)
Personnel	(PE)
Position Administration	(PM)
Requisition Administration	(RT)
Salary Administration	(SA)
Time & Attendance Administration	(TN)
Tools	(SC)
Training Administration	(TR)

---

### NOTES

---

THE ACTION FORM, continued

Field Descriptions,  
continued

**Module** (option list SC01)—optional field. Contains the module to be associated with the program, for example, HR (HRMS), PR (Payroll), BA (Benefits Administration). This field defaults to PP (Payroll/HRMS) if no entry is made.

**Security Code**—optional field. Contains the security code which determines who can access and run programs created by the Solution View program. The Security Code must not be higher than your assigned security level.

If no Security Code is entered, the Solution View created program will have no security assigned to it. This means that anyone can call up the program; copy, change or delete the program; or execute (run) the program.

However, even with a blank security code, field security may restrict a person from viewing certain fields in online reports. The fields will be filled with asterisks (\*\*\*\*\*).

Contact your Security Officer for further information regarding your security limitations.

**Note:** You may wish to access another program that enables you to place a certain amount of security on Solution View programs. The **Q-NAME** program allows users to have a certain amount of security affiliated with their authored programs without specific security set-up.

Q-NAME compares the user's log-on ID to the program's Author ID. The only person who can then change, copy, or delete the Solution View program is the person who originally created it or the Security Officer. Anyone else who attempts any of the above actions will receive a reject message indicating that the Operator is not the program author, and therefore, the action is not allowed. However, anyone can execute (run) the program.

In order to activate this feature, remove the at-sign (@) comment designation in the specified lines of the Q-NAME program code. The Q-NAME program is a subroutine that in turn calls the program RQAUTH that validates the OPID. Remove the @ sign from RQAUTH and reload the program and then reload the Q-NAME program.

## The Action Form Option Field



A screenshot of a software interface showing a dropdown menu. The label 'Option:' is on the left. The dropdown box is blue and contains the text '(None)' with a small downward-pointing arrow on the right side.

- | <b>Option</b>        |
|----------------------|
| (None)               |
| Field Selection Menu |
| Leave Field Help     |
| Leave Menu           |
| Previous Form        |
| Reload Program       |
| Restart Process      |
| Set Inquiry to Entry |

---

### NOTES

---

THE ACTION FORM, continued

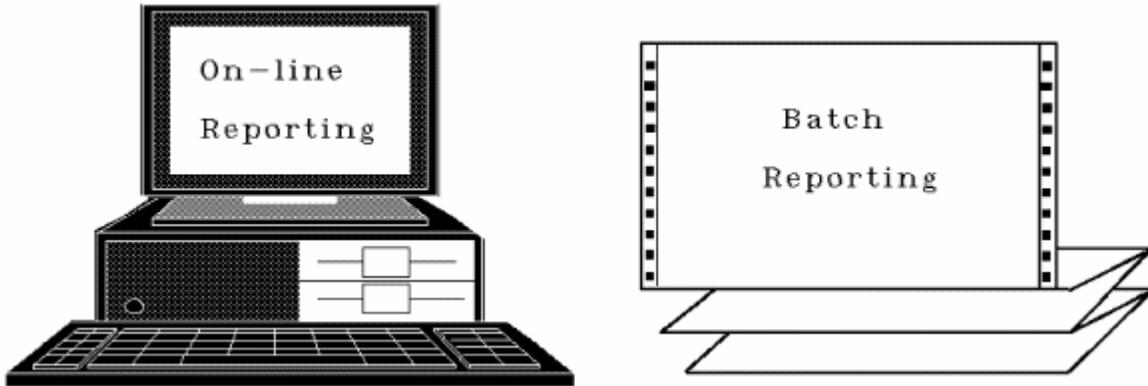
Field Descriptions,  
continued

**Option** (option list SC09)—resides in the top right corner of most of the forms in Solution View. It allows you to perform generic Solution View functions from the form you are currently entering or viewing.

Here are some extra notes on the available options:

- **Field Selection Menu**—access the Field Selection Menu, a multi-level menu structure of field names that contains a Quick Reference Menu with the 45 most commonly used fields, and a User Quick Reference Menu.
- **Leave Field Help**—exits field help and returns you to Solution View.
- **Leave Menu**—exits to the Action form if you are viewing another form or exit from the Field Name Menu when you are done selecting fields.
- **Previous Form**—return to the previous form while in Inquiry or Entry mode.
- **Reload Program**—reload (compile) the program (to cause a RELOAD at a point other than the end of the form chain).
- **Restart Process**—used to go back to the Action form.
- **Set Inquiry to Entry**—change from Inquiry mode to Entry mode (changes from entry back to inquiry are not permitted).

## The Action Form Gateway to the Solution View Form Series



- **Extract Data**
  - **Entry Form**
    - **New Fields Definition**

---

NOTES

---

**THE ACTION FORM, continued**

**Using the Action Form**

The specific procedure for using the Action form is covered in each section that follows. The field descriptions will not be repeated, but you will be instructed on the proper entries that lead to the form series you have chosen. This allows you to use each section as a complete procedure for the type of program you are creating.

After all entries have been made on the Action form, execute the form. If there are any errors, correct them before proceeding. When the Action form is error-free, the first form of the Solution View facility you selected displays.

**What's Next?**

After the summary and exercise for this section, we will go to the next section and use a Query report as our starting point.

We will use the premise that you have just received a memo requesting an online report. We will walk through some planning steps and then actually create the program that satisfies the request.

## Section Summary

### ■ The Action Form

---

NOTES

---

**SECTION SUMMARY**

In this section, you learned how to access the Action form and its purpose.

■ The Action Form

---

---

---

---

## Section 3 Exercise

---

NOTES

---

**SECTION EXERCISE**

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Sign on to The Solution Series and access the Solution View Action form.
2. From the Action form, display a list of all Query reports that have been created by all users.
3. Name two of the actions you can perform by using the Action form Option field.

---

---

---

**NOTES**

---

## SECTION 4: CREATING A PROGRAM TO GENERATE AN ONLINE REPORT

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW .....	41
PLANNING AHEAD: W.I.I.F.M. ....	43
THE QUERY WRITER FORM SERIES .....	51
SELECTING THE ACTION .....	53
QUERY OPTIONS FORM.....	55
DEFINING THE FIELDS .....	57
SELECTING SEGMENT KEYS.....	67
CREATING SELECTION CRITERIA LOGIC .....	73
DISPLAYING THE REPORT.....	85
REPORT SAMPLE .....	93
INQUIRY, CHANGING, COPYING, DELETING.....	95
SECTION SUMMARY .....	109
SECTION EXERCISE.....	113

## Section Objectives

- **Determine the format and content of the report to be produced**
- **Access and complete the form series for creating an online report**
- **Run the online program**
- **Inquire, change, copy, and delete the online program**

---

NOTES

---

**SECTION OVERVIEW**

**Purpose**

In this section you will learn how to use Solution View to create programs that are run online using the Query capability.

**Objectives**

When you have completed this section you will be able to do the following:

- Determine the format and content of the report to be produced
- Access and complete the form series for creating an online report
- Run the online program
- Inquire, change, copy, and delete the online program

## Requirements Memo

**M E M O**

TO: Cyborg End-User  
FROM: The Boss  
DATE: August 1, 199X  
SUBJECT: Salary Survey - Immediate Report Online

---

Due to unexpected requirements from the Corporate Office, I need to see some salary figures right away. Within the next 24 hours, would you please work up a program to give me what I need? I'll need something that can be called up online on the computer in my office whenever I need the information.

The information I need for tomorrow is fairly routine, but the specifics may change down the line. Be sure this online program can be revised whenever the need arises.

The list must contain the employee's legal name, their city and state, their annual salary and the date of their next scheduled salary review. We only need to see employees on the list who have a job code higher than 12550 and earn more than \$35,000.00 annually.

Make sure the listing is organized in employee number order, with the lowest number first. And also, I'll need an employee count at the end of the list, as well as a total and an average of their annual salaries.

Get back to me if you need any additional information.

---

### NOTES

---

**PLANNING AHEAD: W.I.I.F.M.**

**What's In It For Me?**

The Query Writer is a Solution View tool used to write programs that are run online by the Query program. The data you request is displayed on a series of forms that can be viewed consecutively, and totals can be displayed at the end of the report.

This means that you can create your own online report using the Solution View tools. You are not required to know the Cyborg Scripting Language programming language. Still, you will have the online capability with all the flexibility you need to build and display your report.

**Planning the Report**

Prior to completing the Query Writer form series that will create the online program, we suggest that you invest some time in reviewing the original request. This review allows you to define the requirements and the appearance of your report prior to entering the report parameters.

The example on the facing page is a sample memo that requests an online Query report to resolve some information needs. Using this example, it is clear that this manager wants information on employee salaries and next review dates. The information is specific by job code and salary amount. Starting with the specifications that are stated on the memo, let us plan what will appear on the report.

**Requirements Worksheet**

You might want to use a requirements worksheet to plan your online report. A worksheet would allow you to research the report content, sequence and detailed specifications before you begin your entries. Once those requirements have been committed to paper, the process of using the Solution View form series becomes an easy next step.

On the next page, we will walk through a sample worksheet that you could use to specify the report requirements.

# Requirements Worksheet

QUERY WRITER

Date: \_\_\_\_\_

<b>Program Name: X</b>		<b>Module Specific? Y / N</b>		
<b>Program Title:</b>		<b>Security? Y / N</b>		
<b>As of Date:</b>		<b>Total Employee Count? Y / N</b>		
<b>Employee Selection Criteria? Y / N</b>		<b>Display Sequence? <input type="checkbox"/> Employee Number <input type="checkbox"/> Other</b>		
<b>Sequence</b>	<b>Field Name</b>	<b>Segment Occurrence</b>	<b>Print</b>	<b>Total/Average/Both</b>
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>		<b>Action:</b>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>

Action Options: C, P, N, E

---

**NOTES**

---

**PLANNING AHEAD: W.I.I.F.M.,** continued

**Using the Worksheet** Your worksheet might look something like the sample on the facing page. The worksheet contains the following specifications:

- **Program Name**—What is the name of the Query program that will produce the online report? As you recall, all user-defined program names may contain up to six characters and must begin with X.
- **Program Title**—What is the heading that will appear at the top of each page of the online report? There is a 30-character limit for a program title.
- **As Of Date**—Do you want the most current data (default) or data as of a particular date which you specify in this space?
- **Employee Selection Criteria?**—Do you want to specify criteria for selecting employees on this online report?
- **Module Specific?**—Do you want to specify a Module ID to be associated with this online report?
- **Security?**—Do you want to specify the security to be associated with this online report? (The default is no.)
- **Total Employee Count**—Do you want to see a total employee count displayed at the end of the report?
- **Employee Sequence**—In what sequence should the employees be displayed on the online report? (For example, employee number, name, and so forth.) Your answer here depends on the alternate keys you have activated for the Query program that will run this Query Writer program.
- **Field Sequence**—In what sequence should the displayed fields appear on the report from left to right?
- **Field Names**—What are the field names (20 characters) that will be used in the online report? The field names must have been previously defined. The total of all the fields (edited length) or all the field headings may not exceed 78 characters. A reject message informs you if you exceed the limit.

Some of the utilities available to help in your research are Search by Partial Name, Field-Name Table Cross Reference Menu, and Display Field Names for a form. These utilities will be covered later in this section.

## Requirements Worksheet, continued

QUERY WRITER

Date: \_\_\_\_\_

<b>Program Name: X</b>		<b>Module Specific? Y / N</b>		
<b>Program Title:</b>		<b>Security? Y / N</b>		
<b>As of Date:</b>		<b>Total Employee Count? Y / N</b>		
<b>Employee Selection Criteria? Y / N</b>		<b>Display Sequence? <input type="checkbox"/> Employee Number <input type="checkbox"/> Other</b>		
<b>Sequence</b>	<b>Field Name</b>	<b>Segment Occurrence</b>	<b>Print</b>	<b>Total/Average/Both</b>
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>		<b>Action:</b>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>
<b>Field Name:</b>		<b>E G L F</b>		<input type="checkbox"/>

Action Options: C, P, N, E

---

**NOTES**

---

**PLANNING AHEAD: W.I.I.F.M.,** continued

**Using the Worksheet,**  
continued

- **Segment Occurrence**—For those fields which reside in a non-dated multiple-occurrence segment, which segment key(s) should be used for the report? You can specify a single segment key, a range of keys, and/or a list of keys.
- **Print**—Should this numeric field print on the report? (Optionally, you may not print the field, but you can still indicate a total or average of the field at the end of the report.) Non-print fields do not count towards the limit of 78 characters. This entry ties in with the next characteristic below.
- **Total/Average/Both**—Should this numeric field be totaled (4), averaged (6), or both (5) at the end of the report? Leave this space blank to indicate no totaling and averaging of a numeric field.
- **If Employee Selection Criteria**—What employee selection criteria should be used to determine which employees will appear in the report? (For example: salary ranges, date of hire, job codes.) Specify the field names and the values that should be used for comparison. Use the E (equals), G (greater than), L (less than) indicators to make your comparison. Use F (found) to select employees based on the presence of a specific segment on their Master File.

You may also specify NOT prior to your indicators.

- **Action**—You will use this field later when processing the Solution View forms.

## Requirements Worksheet, continued

QUERY WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X Q W S A L		<b>Module Specific?</b> Y / N NO		
<b>Program Title:</b> EMPLOYEE SALARY SURVEY		<b>Security?</b> Y / N NO		
<b>As of Date:</b>		<b>Total Employee Count?</b> Y / N YES		
<b>Employee Selection Criteria?</b> Y / N YES		<b>Display Sequence?</b> <input checked="" type="checkbox"/> Employee Number <input type="checkbox"/> Other		
<b>Sequence</b>	<b>Field Name</b>	<b>Segment Occurrence</b>	<b>Print</b>	<b>Total/Average/Both</b>
1	EMPLOYEE-NAME		YES	
2	CITY/STATE	001	YES	
3	ANNUAL-SALARY		YES	BOTH
4	NEXT-REVIEW-DATE		YES	
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>		<b>Action:</b>
<b>Field Name:</b> JOB-CODE		E G L F G 12550		<input type="checkbox"/> N
<b>Field Name:</b> ANNUAL-SALARY		E G L F G \$35,000.00		<input type="checkbox"/> P
<b>Field Name:</b>		E G L F		<input type="checkbox"/>
<b>Field Name:</b>		E G L F		<input type="checkbox"/>

Action Options: C, P, N, E

---

### NOTES

---

**PLANNING AHEAD: W.I.I.F.M., continued**

**Check Your Requirements**

The completed worksheet on the facing page is an example of specifications that will be used to define the report to be created using the Query Writer form series. When you have completed your worksheet, you will have a draft version of the query and a list of the fields and their associated characteristics.

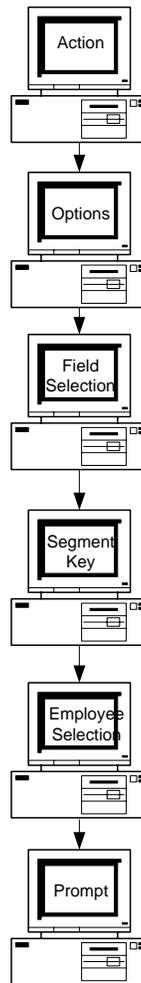
From what we have specified here, we are able to deduce the following:

- four fields are to be displayed
- we will have to total and average one field
- employees on the list are selected by job code and salary amount
- we want a count of employees and a total and average of the annual salaries

**What's Next?**

Now you are ready to use the Solution View tools to enter your program requirements. We will start by previewing the form series for a Query program.

## Query Writer Form Series



---

### NOTES

---

## THE QUERY WRITER FORM SERIES

### Query Writer Forms

The Query Writer is used to write programs that are run online by the Query program. The data you request is displayed on a series of forms that can be viewed consecutively, and totals are displayed at the end of the report.

These are the forms, required and optional, that are available in Solution View to create your online Query report program:

- **Action form**—used to name the Query program, give the report a title, and select the type of action to be performed.
- **Query Options**—used to indicate an As-Of date, count employees, and indicate that you want employee selection criteria.
- **Query Field Selection form**—used to define the fields in the Query report and to indicate whether totals and/or averages are to be displayed.
- **Stacked Segment Key form**—(optional) used to tell the system which key you will use for selecting a specific multiple-occurrence segment where the key is not a date.
- **Employee Selection Criteria form**—(optional) used to determine which employees will be included in the Query report.
- **Prompt form**—informs you that your program has been written and compiled successfully.

**Note:** The **Copy** and **Change Option** forms also exist within the Query Writer facility. These are secondary forms which are not required to create a Query report. They are discussed later in this section.

Now that you have previewed the form series that you will use to create the online Query report, let us look at each task in detail and see what tools are provided on each form and how to use them.

## Action Form

Solution View Tool Kit      Option:

Name of Program:       Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input checked="" type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input checked="" type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

---

### NOTES

---

## SELECTING THE ACTION

**Action Form**                    The first Query Writer form, the Action form, is used to identify the Query report. It takes you directly to the Query form series.

**Accessing the Form**        **Selection:**                    **Step:**

**Navigator**



User Tools

User Tools



Solution View

**Result:** The Solution View Action form displays.

**Field Descriptions**        The Action form field descriptions have already been presented in Section 3. In this section you will learn how to use those fields to access the Query Writer series of forms.

**Using the Action Form**        To add a Query program:

**Selection:**                    **Step:**

**Name of Program**

1. Type the one- to six-character name of the program, beginning with X, (for example, XQWSAL).

**Title**

2. Type a title for the report. This title shows on each page of the Query Report (optional).

**Program Type**

3. Select Query.

**Action**

4. Select Add.

**Module**

5. Select the module ID to be associated with the report (optional).

**Security Code:**

6. Type the Security Code for the report (optional).



7. Click Save or press ENTER.

**Result:** When the Action form is error-free, the Query Options form displays.

**Practice**                        Using the worksheet, please complete this form for your program.

## Query Options Form

Solution View Query Writer: XQWSAL      Option: [dropdown]

Query Options

As Of Date [input field]  
If omitted Query will show current data.

General Options

- Selection Criteria
- Count Employees

---

### NOTES

---

## QUERY OPTIONS FORM

**Query Options Form** The second form in the Query Writer form series, the Query Options form, is used to indicate As Of date options and selection criteria and gives you the ability to count employees.

**Field Descriptions** The Query Options form contains the following fields:

- **As Of Date**—for dated ‘L’ segments, used to specify the date up to which information is to be processed. The segment is included if its date is equal to or prior to this date. This date must be entered in century format, (for example, 12-09-1999 or 19991209).

The date defaults to the current date if no entry is made.

**Note:** You can use an As-of date from any form that has the date as the first key field, such as the Job Assignment/Changes form (05-SCR).

- **Selection Criteria**—used to add an Employee Selection Criteria form to the series of Query Writer forms to indicate your answer to the question: “Do you want to narrow the list of employees based on one or more specific selection criteria?”

When a new Query program is being added, the default for this option field indicates that employee selection criteria is desired.

If you do not want to enter selection criteria, click the box to clear this option.

- **Count Employees**—used to indicate your answer to the question: “Do you want a count of employees who appear on the report?”

When a new Query program is being added, the default for this option field indicates that employees will be counted.

If you do not want an employee count, click the box to clear the option.

**Practice** Using the worksheet, please complete this form for your program.

## Query Field Selection Form

Solution View Query Field Selection: XQWSAL    Option:

Field Name:     Total Option:

<input type="text"/>	<input type="text"/>

---

### NOTES

---

## DEFINING THE FIELDS

**Query Field Selection Form** The third form in the Query Writer form series, the Query Field Selection form, is used to list the fields to be used in the Query program. It also allows you to define whether numeric fields will be totaled and/or averaged at the end of the report.

**Field Descriptions** The Query Field Selection form contains the following fields:

- **Field Name**—used to specify the data field names to be used in the Query Report. The total of all the fields (edited length) or all the field headings may not exceed 78 characters. A reject message informs you if you exceed the limit. (Take into consideration that the Solution View program will add one character to each field as a spacer before the next field.) The data field name entry must already exist on the file.

Your security assignment determines which fields you can access. If your program includes fields to which you are denied access, you will be unable to reload your query.

### **Tools for finding Field Names**

**Field Name Table**—accessing this table in a menu-driven format can be done by placing question marks (?) in the fields and clicking Save or pressing ENTER.

For each question mark (?) entered you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing an X next to each one. You may also place a 1, 2, 3, etc., in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

You can access field help while in the Field Name Table Menu by placing a question mark (?) in the entry box next to the field. To leave the menu select Leave Field Help from the Option field to return to the Field Name Table Menu.

After selecting your data field name entries from the Field Name Table Menu, select Leave Menu from the Option field to return to the Query Field Selection form.

**Note:** When you are accessing the Field Name Table you are accessing the table for The Solution Series, not the Writer tools. Every Solution Series field will be displayed, but not every field is accessible to Solution View.

## Tools for finding Field Names

Search Field Name Table for Field Name SEARCH

The SEARCH program allows you to search for a field name stored on the Field Name Table. SEARCH displays a list of names based on a partial entry in the Key fields.

Complete the text boxes below, then hit enter:

Enter the WORD/VERB/COMMAND:

**SEARCH**

Screen Label To Field Name Cross Ref. FLABEL

This program will list the Field Name and associated label of a screen.

Complete the text boxes below, then hit enter:

Enter Screen Name:

**FLABEL**

---

**NOTES**

---

**DEFINING THE FIELDS**, continued

**Tools for finding  
Field Names,**  
continued

**Search Field Name Table (SEARCH)**—The program SEARCH can also be used without accessing the Field Name Table in the menu driven format. This program can be accessed by selecting:

<u>Selection:</u>	<u>Step:</u>
<b>Navigator</b>	 Development Tools
	Fields and Verbs
	 Search by Partial Name

The system will prompt for the word or field name that you wish to find. Once the form is executed, a list of field names will be returned based on your entry.

**Form Label to Field Name Cross-Reference (FLABEL)**—The program FLABEL is a good tool if you know which form you want to pull information from. The FLABEL program will list the form label as it appears on the form and then the corresponding field name to the right. To access this utility select:

<b>Navigator</b>	 Development Tools
	Fields and Verbs
	 Display field Names for a Form

The system will prompt for a form name (for example, 40-SCR). Once the form is executed, the system will return the cross-reference list.

Other utilities include Field Documentation Display (FIELDS) and Field Name Table Cross-Reference Menu (F-MENU).

## Tools for finding Field Names, continued

---

NOTES

---

**DEFINING THE FIELDS**, continued

**Tools for finding Field Names,** To get the definition of a field the user should select the relevant field on the form and click What's This.  
continued

**Selection:**



**Step:**

1. With relevant field on the form selected, click What's This.

**Result:** The pointer changes to an arrow with a question mark beside it.

2. Click the field again.

**Result:** A Windows Help form appears with the field details.

## Query Field Selection Form

Solution View Query Field Selection: XQWSAL    Option:

Field Name:     Total Option:

<input type="text"/>	<input type="text"/>

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Field Descriptions**,  
continued

**Total Option** (option list SC23)—**for numeric fields only**, used to specify whether the field data is to be printed in detail, totaled, averaged, or some combination of these options. Use the drop down list box to display the options. When left blank, the report displays the field but does not total the field values.

Valid options for this field are:

- Average only (6)
- Print/Average (2)
- Print/Total (1)
- Print/Total/Average (3)
- Total Only (4)
- Total/Average (5)

## Query Field Selection Form, continued

Solution View Query Field Selection: XQWSAL    Option:

Field Name:     Total Option:

<input type="text" value="CITY/STATE"/>	<input type="text"/>
<input type="text" value="ANNUAL-SALARY"/>	<input type="text" value="Print/Total/Average"/>
<input type="text" value="NEXT-REVIEW-DATE"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Using the Query Field Selection Form**

To use the Query Field Selection form:

- | <u>Selection:</u><br>Field Name | <u>Step:</u>  |
|---------------------------------|---|
|                                 | 1. Type the first field name to be used in the report (or type a question-mark in the field, or select Field Selection Menu in the Option field to select field names from a list). |
| Total Option                    | 2. For numeric fields, select the option to cause a display of print lines, totals, and averages at the end of the report.  |
|                                 | 3. Repeat steps 1 and 2 for each field used in the report.  |

Errors on this form must be resolved before proceeding. You may either fix the error or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

## The Stacked Segment Key Form

Solution View Query Key Values: XQWSAL      Option: ▼

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CITY/STATE

Required Keys	Not	EGL	Literal Compare Value	Action
NAME-CODE	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

Selecting the First Occurrence

Selecting a Specific Occurrence

Solution View Query Key Values: XQWSAL      Option: ▼

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CITY/STATE

Required Keys	Not	EGL	Literal Compare Value	Action
NAME-CODE	<input type="checkbox"/>	E	999	P
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

---

### NOTES

---

## SELECTING SEGMENT KEYS

### Stacked Segment Key Form

The Stacked Segment Key form is displayed only when a field name entered on the Query Field Selection form is in a multiple occurrence (stacked) segment whose primary Key field is not a date. This is the only time this form is used. The form is displayed automatically for each segment (not field) you listed on the Query Field Selection form; you do not need to request it.

Solution View assumes you want to display the requested field's data from the first occurrence in the stacked segment. If another occurrence is desired, the criteria must be entered on this form. You may bypass this form by pressing ENTER, or by clicking Bypass this form. (This action assumes the default, the first occurrence.)

The purpose of this form is to include/exclude segment **occurrences** in the program. If the employee does not have the segment of requested key data, this alone does not exclude the **employee** from the report.

### Field Descriptions

The Stacked Segment Key form contains the following fields:

- **Requested Detail Line Field**—an inquiry-only field that displays a field name from the Query Field Selection form if that field occurs in a multiple-occurrence segment whose key is not a date.

Only the first requested field of a multiple occurrence segment is displayed for the selection of the segment key(s). All other fields from the same segment use the same comparison logic.

- **Required Keys**—displays the key field name related to the fields selected on the Query Field Selection form. The key field name(s) used as the basis for selecting occurrences is placed here automatically by Solution View. **Do not change this field.**
- **Not**—used to indicate that the selection criteria checks for a condition that does NOT exist. Check the box, or enter N in this field, to create selection criteria that checks for a condition that does NOT exist. The default entry of blank is interpreted in the positive, in other words, *is*.

## The Stacked Segment Key Form, continued

Solution View Query Key Values: XQWSAL      Option:  

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CITY/STATE

Required Keys	Not	EGL	Literal Compare Value	Action
NAME-CODE	<input type="checkbox"/>	E	001	P
	<input type="checkbox"/>			
	<input type="checkbox"/>			

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

---

### NOTES

---

**SELECTING SEGMENT KEYS**, continued

**Field Descriptions,**  
continued

- **EGL** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are:
  - E or = (equal to)
  - G or > (greater than)
  - L or < (less than)
- **Literal Compare Value**—used to specify a literal value to be compared to the Required Keys field.

Ranges can be entered using slashes (/) to indicate an inclusive range, or semi-colons (;) to indicate an OR single or range condition. **The EGL field must be an E (=) to use ranges or OR conditions.**

In a Query report, when a range (/) or list (;) is specified, only the first literal compare value that is matched to the Required Key is processed and no subsequent checks are made. **A QUERY CAN DISPLAY ONLY ONE LINE PER EMPLOYEE.** The first true condition creates the output line.

Note the following examples:

- To include HEDs 005 through 500

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER		E	005/500

- To exclude HED 500 and HED 510

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER	N	E	500;510

- To include HEDs 001 through 500, or HED 510

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER		E	001/500;510

**Note:** A key field may not contain spaces. Therefore, a blank entry in the Literal Compare Value field is invalid.

- **Action** (option list SC22)—used to specify whether to include/exclude the segment occurrence being compared.

N in the field causes a segment that meets the condition to be retained if it also meets the next statement(s).

P in the field causes a segment that meets the condition to be retained.

## The Stacked Segment Key Form, continued

Solution View Query Key Values: XQWSAL      Option:

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CITY/STATE

Required Keys	Not	EGL	Literal	Compare Value	Action
NAME-CODE	<input type="checkbox"/>	E	001		P
	<input type="checkbox"/>				
	<input type="checkbox"/>				

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

---

### NOTES

---

**SELECTING SEGMENT KEYS**, continued

**Using the Stacked Segment Key Form**

The procedure for using the Stacked Segment Key form to select a segment occurrence is:

To select the **first occurrence** of the stacked segment, click Bypass this form or press ENTER.

To select **any other occurrence**, perform the following steps:

<u>Selection:</u>	<u>Step:</u>
<b>Required Keys</b>	1. Do not change this field, it is automatically updated by the Query Writer.
<b>Not</b>	2. Optionally, indicate N to designate the negative effect of the entry in the EGL field.
<b>EGL</b>	3. Type E (Equal), G (Greater), L (Less), or the symbolic equivalents, =, >, or <, to establish a conditional comparison.
<b>Literal Compare Value</b>	4. Type the literal value to be compared with the corresponding Required Key field.
<b>Action</b>	5. Type P (include in process), or N (check next comparison), to designate the selection of the specific occurrence.
	6. Click Save or press ENTER to move to the next form in the process.
	7. Repeat steps 1 through 6 for each segment key requested.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

## Employee Selection Criteria Form

Solution View Query Writer: XQWSAL      Option:  

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

Display Values

EGLF Values

Actions

---

### NOTES

---

## CREATING SELECTION CRITERIA LOGIC

**Employee Selection Criteria Form** The Employee Selection Criteria form is used to narrow the number of employees that will be selected for the query report. For example, you may want to display only those employees with certain salary ranges, hire dates, and so forth, on the form. This is done by setting a condition on one or more fields. The field(s) does not have to be one of the fields that appear on the Query Field Selection form.

The use of the Employee Selection Criteria form is optional. This form only appears if it is chosen from the Query Options form.

**Criteria used to include/exclude employees on this form may not be inconsistent with segment key selections that were done on the Stacked Segment Key form.** For example, if you selected HED occurrences **not** equal to 600, then 600 should not be included here for employee selection.

**Field Descriptions** The Employee Selection Criteria form contains the following fields:

- **Field Name**—data field name(s) used as the basis for employee selection. The fields on this form do not have to be displayed on the report. Your security assignment determines which fields you can access.

To access the Field Name Table in a menu-driven format, select Field Selection Menu from the Option Field, *or* type a ? in the Field Name field to access the Field Name Table Menu to select the data field name(s). For each ? you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names, and the data fields will appear on the report in the specified order.

- **Not**—to create selection criteria that checks for a negative condition, in other words, a condition that is not true.

For example, if you want only employees who do NOT have an annual salary greater than \$28,000 to appear on the report, you would indicate “Not” in this field (and the appropriate EGL, Literal Compare Value, and Action fields, discussed next).

## Employee Selection Criteria Form Examples of Ranges, OR Conditions, and Found/Not Found Logic

■ **Annual Salary from \$20,000.00 through \$50,000.00**

Field Name	Not	EGLF	Literal Compare Value
ANNUAL-SALARY		E	20000.00/50000.00

■ **Annual Salary of \$22,000.00, \$25,000.00, or \$28,000.00**

Field Name	Not	EGLF	Literal Compare Value
ANNUAL-SALARY		E	22000.00;25000.00;28000.00

■ **Annual Salary of \$20,000.00 through \$26,000.00 or equal to \$30,000.00**

Field Name	Not	EGLF	Literal Compare Value
ANNUAL-SALARY		E	20000.00/26000.00;30000.00

■ **Salary Segment for January 1, 1993, Reclassification Increase**

Field Name	Not	EGLF	Literal Compare Value
ANNUAL-AMOUNT-CHANGE		F	930101;9;I40

■ **A value of spaces is valid. Example:**

Field Name	Not	EGLF	Literal Compare Value
UNION-CODE		E	SPACES

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

Field Descriptions,  
continued

- **EGLF** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Values). Choose the EGLF values button to display the values. Valid entries are:

- E or = (equal to)
- G or > (greater than)
- L or < (less than)
- F (found)

The F value is used to incorporate FOUND or NOT FOUND logic for segment selection. This logic allows you to include or exclude employees based on the presence of a specific segment on their record.

For the F value, the field name can be **ANY** field from the segment. To select a specific occurrence, enter the key field value(s) in the Literal Compare Value field.

- **Literal Compare Value**—used with the corresponding field name entry on this form to specify the selection criteria. The value in each field must be appropriate for the data field in the Field Name field.
- **For EGLF field entries of E, G, and L**, numeric values may be entered with a decimal, commas, or dollar signs, if desired. Dates may be entered in the format YYMMDD or MM-DD-YY. Use century dates when a field displays on the system that way.

Slashes (/) or semi-colons (;) may be entered in the Literal Compare Value field as delimiters. A slash indicates a range and a semi-colon indicates an AND/OR condition. **When using ranges or the OR condition, the EGLF field must be an E (or =).**

A literal value of SPACES is valid here. This can be used to check a field for blanks.

- **For Found/Not Found logic**, one or more key field values must be entered in the proper order, in other words, 1st key field value, 2nd key field value, and so forth. The key field values must be separated by a semi-colon (;).

Only one set of key field values is allowed per Found/Not Found entry. If the Literal Compare Value field is left blank, then the first occurrence of the segment is used.

## Employee Selection Criteria Form and the Codeset Selection Menu

Solution View Query Writer: XQWSAL      Option: [dropdown]

Employee Selection Criteria

Field Name	Not	EGLF	Literal	Compare Value	Action
SEX-CODE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	E		
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			

Display Values

- EGLF Values
- Actions

Solution View Query Writer: XQWSAL      Option: [dropdown]

Codeset Selection Menu For Sex Codes

For Criteria Field: SEX-CODE

Female       Male

Unclassified

----Complete----       Return To Selection Criteria

---

### NOTES

---

**CREATING SELECTION CRITERIA LOGIC**, continued

**Using the Codeset Selection Menu**

The Codeset Selection Menu can be used with the Employee Selection Criteria form when entering data into the Literal Compare Value field for fields that are tied to option lists.

If the field name entered in the Field Name field is tied to an option list, and the Literal Compare value field is left blank, the process will automatically bring the user to the Codeset Selection Menu.

Once there, the selections are made based on the option list description, (for example, Male), and not the code itself, (for example, M).

**Number of Selections**

You can make as many selections that will fit on one line on the Employee Selection Criteria form. If the line following the criteria entry is also blank, in other words, no field name in the Field Name entry field, then up to two lines of selections can be made.

The length of an entry is the length of the code (field length), plus 1. The system will automatically place a semi-colon (;) between each selection made. If two lines of selections are made, the system will also place a "C" (Continue) into the first line's Action Code field.

**Exiting the Codeset Selection Menu**

To leave the Codeset Selection Menu at any time, use the Leave Menu option in the Option field, or click Return to Selection Criteria.

The Codeset Selection Menu will automatically return to the Employee Selection Criteria form when the last entry is processed.

## Employee Selection Criteria Form Action Code Examples

**E** example: Employees who have an ACTIVITY-CODE of 001 **ARE NOT** included on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
ACTIVITY-CODE		E	001	E

**P** example: Employees who have a SEX-CODE of M (males) **ARE** included on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
SEX-CODE		E	M	P

**N and P** example: Employees who have a JOB-CODE of 12550 **AND** a JOB-CODE-EXTENT of 0002 **ARE** included on the report. Satisfying only one of these conditions does not include the employee on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550	N
JOB-CODE-EXTENT		E	0002	P

**P and P** example: Employees who have a JOB-CODE of 12550 **OR** a SALARY-GRADE of S10 **ARE** included on the report. Satisfying either one of these conditions includes the employee on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550	P
SALARY-GRADE		E	S10	P

**C and P** example: Employees who have a CONTROL-3-CODE equal to one of the values on line 1 or line 2 are included in this report.

Field Name	Not	EGLF	Literal Compare Value	Action
CONTROL-3-CODE		E	1255;1926;2183;3065;3067;3402;3555 3505;4721;6800	C P

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

Field Descriptions,  
continued

- **Action** (option list SC22)—used to determine whether to process the record being compared. Choose the Actions button to display the options.

Enter **C** to indicate that the values for comparison continue on the next line; field names and the NOT and EGLF indicators need not be repeated on subsequent lines which contain an action of C.

Enter **N** to indicate that other comparisons need to be checked.

Enter **P** to select the record based on this comparison.

Enter **E** to exclude a record based on this comparison.

The entries have the following effect in the program:

- **C**—the values for this statement continue on the next line.
- **N**—if this statement is true, check the next comparison.
- **P**—if this statement is true, include the employee in the selection process and then check the next line for another comparison.
- **E**—if this statement is true, do not include the employee in the selection process.

## Employee Selection Criteria Form Conditional Statement Examples

**Include employees having a Salary of \$20,000.00 through \$25,000.00**

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	P

**Exclude employees having a Job Code of 12550 OR 16775**

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550;16775	E

**Include only Female employees having a Job Code of 13552 through 13559**

Satisfying only one of these conditions does not include the employee on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	13552/13559	N
SEX-CODE		E	F	P

**Exclude employees with a Salary of \$20,000.00 through \$25,000.00 OR hired before 1992.** Satisfying either one of these conditions excludes the employee from the report.

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	E
HIRE-DATE		L	01-01-92	E

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

**Conditional Statements** Conditional Statements are used to perform an operation dependent on some condition. The conditions that may be used in a conditional statement are:

- |                |                    |
|----------------|--------------------|
| ■ Equal to     | ■ Not Equal to     |
| ■ Greater than | ■ Not Greater than |
| ■ Less than    | ■ Not Less than    |

**Compound Conditional Statements** Compound Conditional Statements are used to test for several conditions. Compound conditional statements include:

- **AND** Tests to see if **all** of several conditions exist. This means that more than one condition must be met by the employee in order to appear on the report.
  - An entry of N in the Action field designates an AND condition.
- **OR** Tests to see if **one** of several conditions exists. This means that at least one of the conditions must be met by the employee in order to appear on the report.
  - An entry of P in the Action field designates an OR condition when you are testing the values between one or more Field Names.
  - A semi-colon (;) inserted between a list of two or more Literal Compare Values designates an OR condition when testing the Field Name entry.
  - A slash (/) inserted between two Literal Compare Values designates an OR condition resulting in a range test for a specific Field Name entry. The range will include the starting value and the ending value.

## Employee Selection Criteria Form

Solution View Query Writer: XQWSAL      Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal	Compare	Value	Action
JOB-CODE	<input type="checkbox"/>	G	12550			N
ANNUAL-SALARY	<input type="checkbox"/>	G	35000.00			P
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					
	<input type="checkbox"/>					

Display Values

- EGLF Values
- Actions

Solution View Query Writer: XQWSAL

Query Program XQWSAL has been RELOADED and may now be run using the QUERY facility.

- Return to the start of the Solution View process
- Execute Query

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

**Using the Employee Selection Criteria Form**

The procedure for using the Employee Selection Criteria form is:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Field Name</b>   | 1. Type the name of the field to be used in the conditional statement.  |
| <b>Not</b>  | 2. Optionally, indicate Not to designate the negative effect of the conditional argument entered in the EGLF field.   |
| <b>EGLF</b>   | 3. Type E (Equal), G (Greater), L (Less), or F (Found), to establish a conditional comparison.  |
| <b>Literal Compare Value</b>  | 4. Type the literal value to be compared with the corresponding Field Name entry.   |
| <b>Action</b>   | 5. Type C (continue values on next line), N (check next comparison), P (include in process) or E (exclude from the process) to designate the action to be taken if the conditional statement is true. |
|   | 6. Repeat steps 1 through 5 for each selection criteria condition that exists.  |
|  | 7. Click Save or press ENTER to move to the next form in the process.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

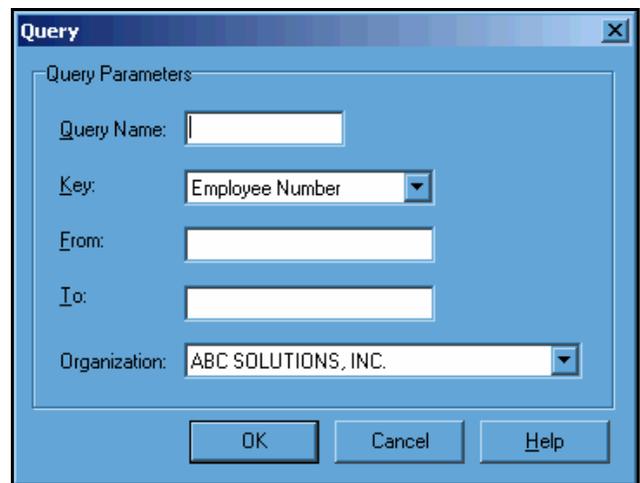
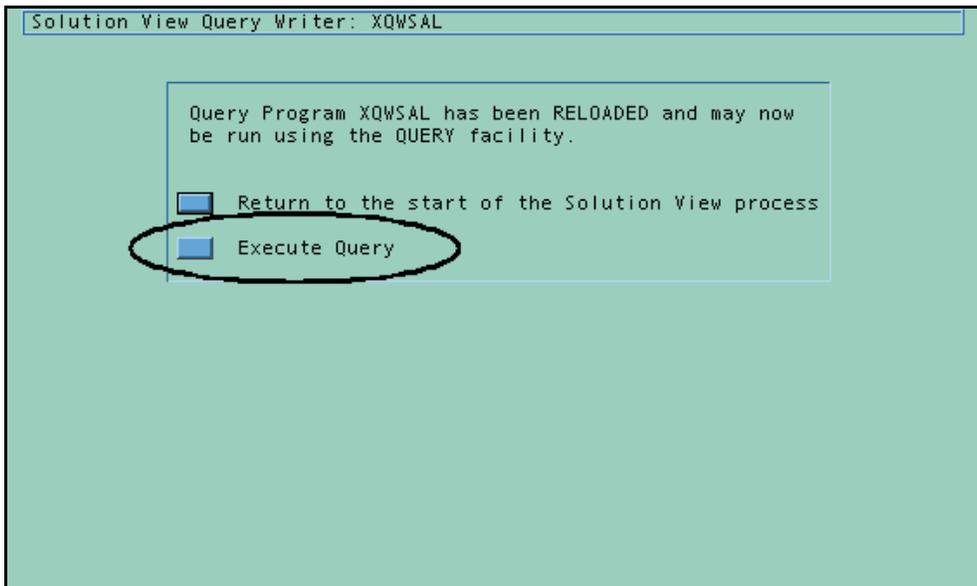
To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the Prompt form displays. It informs you that your Query program has been written and compiled successfully. Cyborg Scripting Language Code now exists for the Query program. The next topic discusses how to execute a Query program.

**Practice**

Using the worksheet, please complete this form for your program.

## Query Program



---

### NOTES

---

## DISPLAYING THE REPORT

**Query Form** Query is a facility which allows you to execute a Query Writer program and view the report online. Query acts like a Command Line which is used to pass parameters to the system for use with your query report.

**Accessing the Form**      Selection:      Step:

**Navigator**       User Tools  
User Tools  
 Run a Query...

**Result:** The Enter Query Parameters dialog box displays.

**Note:** The Query form can be accessed by selecting the “Execute Query” button on the prompt form after the program is RELOADED to execute Query.

**Field Descriptions** The Enter Query Parameters dialog box contains the following fields:

- **Query**—must contain the program name to be executed.
- **Key**—(Alternate Key field) specifies the sequence in which to search the file for entries. For example, the Key could cause Query to search by employee number or employee name.

Use the key HL for queries which have history or labor fields.

To use a key other than the delivered option of Employee Number (00), special set-up procedures are required.

A selection of Employee Number (00) in the key field accesses data on this Control 1-2 only.

Keys other than Employee Number (00) access the data on ALL organizations on file, unless you restrict them by the removal of specified at-signs (@) in the Query source code, and then RELOAD the Query program. Your security assignment determines which organizations you can access.

## From and To Field Reference Chart

Lowest	<b>EBCDIC</b>	<b>ASCII</b>	Lowest
	blank	blank	
	Special Characters	Some Special Characters	
	A	0	
	.	.	
	Z	9	
		Add'l Special Characters	
	0	A	
	.	.	
	9	Z	
∨			∨
Highest			Highest

---

### NOTES

---

**DISPLAYING THE REPORT**, continued

**Field Descriptions,**  
continued

- **From**—specifies the beginning point of the search. The From range is dependent on the entry made in the Key field and usually represents the lowest level range.
- **To**—specifies the ending point of the search. The To range is dependent on the entry made in the Key field and usually represents the highest level range.

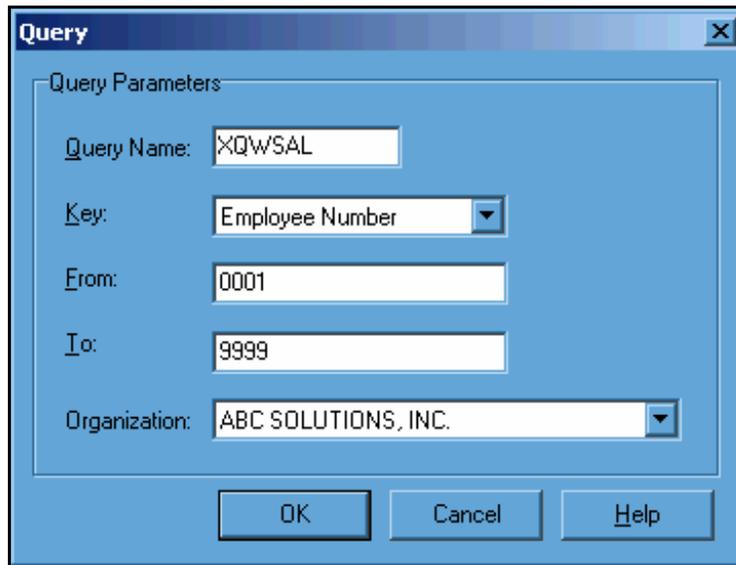
For queries containing history or labor fields, the values in the From and To fields are the Master numbers associated with the history/labor records.

There are two character sets that are used by computers. Depending on the character set that is used at your company, the From and To fields sequence in the following pattern:

EBCDIC—Order is blank, A through Z, then 0 through 9.

ASCII—Order is blank, 0 through 9, then A through Z.

## Query Program



The image shows a Windows-style dialog box titled "Query". It contains a section labeled "Query Parameters" with the following fields:

- Query Name: XQWSAL
- Key: Employee Number (dropdown menu)
- From: 0001
- To: 9999
- Organization: ABC SOLUTIONS, INC. (dropdown menu)

At the bottom of the dialog box are three buttons: "OK", "Cancel", and "Help".

---

### NOTES

---

**DISPLAYING THE REPORT**, continued

**Using Query**            The procedure for using the Enter Query Parameters dialog box to run your Query Writer program is:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---------------------|---|
| <b>Query</b>        | <b>1.</b> Type the name of the Query report to be run, (for example, XQWSAL).   |
| <b>Key</b>          | <b>2.</b> Select the alternate key to designate the order in which the data will be displayed, (for example, by Employee Number). |
| <b>From</b>         | <b>3.</b> Type a From value to establish the beginning search range, (for example, 0001).   |
| <b>To</b>           | <b>4.</b> Type a To value to establish the ending search range, (for example, 9999).  |
| <b>Organization</b> | <b>5.</b> Select the organization (for example, ABC Solutions, Inc.)  |
|                     | <b>6.</b> Click OK or press ENTER.  |
|                     | <b>Result:</b> The Query program executes and displays the requested Query report.  |
|                     | <b>7.</b> Press ENTER to view subsequent forms. Viewing a prior form is not permitted.  |

**Practice**            Using the worksheet, please complete this form for your program.

## QUESCR Program

Launch Enhanced Query Process

Parameters

Query/Extract Name>

Key>

00 Employee Number  
01 Employee Soc Sec Nbr  
02 Employee Name  
See Documentation for others

From>

To>

---

### NOTES

---

DISPLAYING THE REPORT, continued

Using QUESCR

If you are running on Windows or Unix, you may launch a query using the Launch Enhanced Query Process form (QUESCR). Running a query this way allows you to use the Process Monitor to view the progress of the process and the Report Viewer to view the results. The procedure for using the Launch Enhanced Query Process form (QUESCR) to run your Query Writer program is:

Accessing the Form

**Selection:** \_\_\_\_\_ **Step:**

Navigator



User Tools  
User Tools



Launch Enhanced Query Process

**Result:** The Launch Enhanced Query Process form (QUESCR) displays.

**Selection:** \_\_\_\_\_  
**Query/Extract Name**

**Step:**

**Key**

**From**

**To**



1. Type the name of the Query report to be run, (for example, XQWSAL).
2. Enter the key information to search the file for entries. Type 01 for Employee Number, 02 for Employee Social Security Number, or 03 for Employee Name.
3. Type a From value to establish the beginning search range, (for example, 0001).
4. Type a To value to establish the ending search range, (for example, 9999).
5. Click Save or press ENTER.

**Result:** A confirmation form is displayed indicating that the query has been launched. At this point, you can access the Process Monitor and Report Viewer.

Practice

Using the worksheet, please complete this form for your program.

## Query Program Display

Query XQWSAL Key 00 From 1616 99 To 9999

EMPLOYEE SALARY SURVEY

EMPLOYEE-NAME-25	CITY/STATE	ANNUAL SALARY	NEXT-REVIEW DATE
MEYER, JUNE	BURR RIDGE, IL	47,000.00	09-15-1985
MOORE, SAMUEL	JAMAICA ESTATES, NY	63,000.00	10-01-1985
MUIR, LINDA	CHICAGO, IL	42,500.00	09-23-1985
MORRIS, ROBERT	JAMAICA ESTATES, NY	47,000.00	08-01-1985
MERTZ, LYNNE C	DES PLAINES, IL	36,350.00	10-10-1985
MORITZ, KATHERINE C	EL SEGUNDO, CA	58,000.00	09-01-1985
MOREAU, GARDNER	CICERO, IL	42,350.00	09-01-1985
MOHR, MICHAEL T	CHICAGO, IL	48,750.00	08-03-1985
JOHNSEN, RICH	JAMAICA ESTATES, NY	72,267.48	11-01-1985
BARTHOLOW III, JONATHAN	GLENVIEW, IL	52,000.00	01-01-1986
SWEENY, BARBARA	CHICAGO, IL	48,250.00	01-01-1987
AUSTIN, STEVEN F	CHICAGO, IL	36,000.00	12-01-1986
JOHNSON, WALTER D	NAPERVILLE, IL	35,250.00	01-01-1987
CACH, ROBERT	PEARL RIVER, NY	50,780.00	
ANDERSON, DANIEL M	NAPERVILLE, IL	48,100.00	01-01-1986

TOTALS FOR QUERY PROGRAM XQWSAL:

EMPLOYEE COUNT:	56		
ANNUAL-SALARY:	3,209,485.58	AVERAGE:	57,312.24

----Complete----

- 00 Employee Number
- 01 Employee Soc Sec Nbr
- 02 Employee Name

See Documentation for Others

---

### NOTES

---

## REPORT SAMPLE

### **Online Query Display**

The Query report displays and lists all employees who fit the criteria of the Query program and who are within the range in the From and To fields.

Reports may extend to more than one form. Pressing ENTER allows you to view the rest of the report, one form at a time.

The prompt at the bottom of the sample form indicates that the totals are ready to be displayed.

A ---Complete--- message displays when all form and totals have been presented.

A hard-copy of your Query report may be produced by executing Query in batch mode.

## Program Inquiry

Solution View Tool Kit      Option:

Name of Program:       Title:

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Query Writer: XQWSAL      Option:

Query Program General Information

Program Title: EMPLOYEE SALARY SURVEY

Module:      Security Code:

Author: S.O.      Last Modified: 09-16-02      By: S.O.

---

Continue the WRITER inquiry

---

### NOTES

---

## INQUIRY, CHANGING, COPYING, DELETING

**Inquiry Capability** Once you have created a Query program, the capability of viewing the form series through an inquiry function is provided. While in the inquiry mode, you may change a Query program by form by selecting the Set Inquiry to Entry option in the Option field of the form(s) you want to change.

**Accessing the Inquiry Feature** The procedure for accessing the Inquiry feature is:

**Selection:**                      **Step:**

- Navigator**
1.  User Tools  
User Tools
  -  Solution View

**Result:** The Action form displays.

- Name of Program**
2. Type a previously-created Query program name, (for example, XQWSAL).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

- Program Type**
3. Select Query.

- Action**
4. Select Inquiry.



5. Click Save or press ENTER.

**Result:** An inquiry version of the first form in the series displays.

6. Click Continue the WRITER inquiry or press ENTER to move to each successive form in the series.

## Changing a Program

The screenshot shows the 'Solution View Tool Kit' window. At the top right, there is an 'Option:' dropdown menu. Below it, two text boxes are visible: 'Name of Program:' containing 'XQWSAL' and 'Title:' containing 'MIDWEST EMPLOYEE SALARY SURVEY'. The main area is divided into two columns of radio button options. The left column, titled 'Program Type', includes: 'Entry Screen', 'New User Fields', 'Query' (which is selected), 'Report', 'PC Download', and 'Extract Routine'. The right column, titled 'Action', includes: 'Add', 'Change' (which is selected), 'Delete', 'Inquiry', 'Program List', 'Program List By User', and 'Copy Program'. At the bottom, there is a 'Module:' dropdown menu and a 'Security Code:' text box. A footer note at the bottom of the window reads: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Changing a Program** Once you have created a Query program you have the option of changing the Query Program.

It is a good idea to make a copy of the original report using the Program Copier form which may be accessed from the Action form. You can then change the duplicate and have two different programs on file.

**Accessing the form**

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools
-  Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Query program name, (for example, XQWSAL).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Title**

3. Type a new description of the Query, (for example, Midwest Employee Salary Survey) (optional).

**Program Type**

4. Select Query.

**Action**

5. Select Change.

**Module:**

6. Type a new module ID (optional).



7. Click Save or press ENTER.

**Note:** If you did not change the Title, Module, or Security fields and they were previously entered, a warning form is returned. This allows you to change the fields or leave them as they are and execute the form.

**Result:** The Query Change Option form displays.

## Change Option Form

Solution View Query Writer: XQWSAL      Option: [dropdown menu]

Query Change Options

- Field Selection Entries
- Selection Criteria Entries
- Key Field Values Only

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Change  
Option Form**

Select the Field Selection Entries box to change the original field selections.

Select the Selection Criteria Entries to change the Employee Selection Criteria.

Scroll through the forms and recreate the Query program. The forms you will use depend on which change(s) you indicated you wanted on the Query Change Options form.

You may also change a Query program by using the Inquiry Option on the Action form. Select the Set Inquiry to Entry option (E) in the Option field of the forms you want to change.

**Subsequent Change  
Forms/Sequence  
Numbers**

If you have indicated you want to change a Query program, you may also change the field or employee selection criteria sequence.

## Copying a Program

The screenshot shows the 'Solution View Tool Kit' interface. At the top left is the title 'Solution View Tool Kit' and at the top right is a dropdown menu labeled 'Option:'. Below this is a form with two main sections: 'Program Type' and 'Action'. The 'Program Type' section contains radio buttons for 'Entry Screen', 'New User Fields', 'Query' (which is selected), 'Report', 'PC Download', and 'Extract Routine'. The 'Action' section contains radio buttons for 'Add', 'Change', 'Delete', 'Inquiry', 'Program List', 'Program List By User', and 'Copy Program' (which is selected). Below these sections are two more input fields: 'Module:' with a dropdown menu and 'Security Code:' with a small text box. At the bottom of the window, there is a footer text: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Copying a Program**

You may copy and rename any existing Query program using the Program Copier form accessed from the Action form. You can use the copy to make modifications and have two different programs on file.

**Accessing the Form**

**Selection:**      **Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Query program name, (for example XQWSAL).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Query.

**Action**

4. Select Copy Program.



5. Click Save or press ENTER.

**Result:** The Program Copier form displays.

## Program Copier Form

Solution View Query Writer: XQWSAL      Option:

Query Program Copier

New Name:

Original Title:

Module:

Security Code:

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING**, continued

**Field Descriptions**

The Program Copier form contains the following fields:

- **New Name**—used to enter the new Query name.

**Note:** All Query program names must begin with X. The new Query name must be unique, in other words, it cannot be a name that already exists.

- **Original Title**—displays the current title of the Query that is being copied. Optionally, you may change this title for the copied Query. Press TAB to allow the current entry to remain.
- **Module**—displays the current module ID associated with the program. Optionally, you may change this entry for the copied Query. Press TAB to allow the current entry to remain.
- **Security Code**—displays the current security code associated with the program. Optionally, you may change this entry for the copied Query. The Security Code must not be higher than your assigned security level. Press TAB or press ENTER to allow the current entry to remain.

## Program Copier Form, continued

Solution View Query Writer: XQWSAL      Option:

Query Program Copier

New Name:

Original Title:

Module:

Security Code:

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Program Copier Form**

The procedure for using the Program Copier form is:

<u>Selection:</u>	<u>Step:</u>
<b>New Name</b>	<b>1.</b> Type a name for the Query copy, (for example, XMWSAL).
<b>Original Title</b>	<b>2.</b> Optionally, type a new title for the copied Query, (for example, Midwest Employee Salary Survey). Press TAB to bypass the field.
<b>Module</b>	<b>3.</b> Optionally, select a new module to be associated with the Query. Press TAB to bypass the field.
<b>Security Code</b>	<b>4.</b> Optionally, type a new Security Code for the Query.
	<b>5.</b> Click Save or press ENTER.

**Result:** The Query Change Options form displays.

**Altering the Copied Program**

You are now given the ability to alter the copied program.

Select the Field Selection Entries box to change the original field selections.

Select the Selection Criteria Entries to change the Employee Selection Criteria.

Scroll through the forms and recreate the Query program. The forms you will use depend on which change(s) you indicated you wanted on the Query Change Options form.

## Deleting a Program

Solution View Tool Kit      Option:

Name of Program:       Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input checked="" type="radio"/> Query	<input checked="" type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Query Writer: XQWSAL      Option:

Name of Program:       Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

**\*\* QUERY program XQWSAL has been deleted \*\***

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Deleting a Program**

You may delete an existing Query program from the Action form.

**Accessing the Delete Feature**

The procedure for deleting a program is:

**Selection:** \_\_\_\_\_

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Query program name, (forexample, XQWSAL).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Query.

**Action**

4. Select Delete.



5. Click Save or press ENTER.

**Result:** A message displays to inform you that the program has been deleted.

## Section Summary

- **Planning Ahead: W.I.I.F.M.**
- **The Query Writer Form Series**
- **Selecting the Action**
- **Query Options Form**
- **Defining the Fields**
- **Selecting Segment Keys**

---

NOTES

---

**SECTION SUMMARY**

In this section, you learned how to use the Query Writer to create programs that are run online by the Query program.

- Planning Ahead: W.I.I.F.M.

---

---

---

- The Query Writer Form Series

---

---

---

- Selecting the Action

---

---

---

- Query Options Form

---

---

---

- Defining the Fields

---

---

---

- Selecting Segment Keys

---

---

---

## **Section Summary**, continued

- **Creating Selection Criteria Logic**
- **Displaying the Report**
- **Report Sample**
- **Inquiry, Changing, Copying, Deleting**

---

NOTES

---

**SECTION SUMMARY**, continued

You also learned:

- Creating Selection Criteria Logic
- 
- 
- 

- Displaying the Report
- 
- 
- 

- Report Sample
- 
- 
- 

- Inquiry, Changing, Copying, Deleting
- 
- 
-

## Section 4 Exercise

---

NOTES

---

**SECTION EXERCISE**

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Using the requirements worksheet and memo on the following pages, complete the Query report specified by the memo and worksheet.
2. Run your completed report using the Query facility.

## QUERY WRITER

### MEMO

**TO:** Cyborg End User  
**FROM:** The Boss  
**SUBJECT:** Employment Agency Statistics

---

I need to get a breakdown and total on all the new hire placement fees we have incurred through our employment agency. In addition, I need a breakdown and total of any relocation expenses.

Let me know the employee's name, job title, annual salary, the agency fee, and the relocation expense.

I only want the people with a resulting employment status of active salaried regular full-time and hourly regular full-time.

Have the program ready by tomorrow morning. I want to be able to run it on my computer before noon.

**Section 4: Creating a Program to Generate an Online Report**

QUERY WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X A G N C Y		<b>Module Specific?</b> (Y) / N		
<b>Program Title:</b>		<b>Security?</b> Y / (N)		
<b>As of Date:</b>		<b>Total Employee Count?</b> Y / N		
<b>Employee Selection Criteria?</b> Y / N		<b>Display Sequence:</b> <input type="checkbox"/> Employee Number <input type="checkbox"/> Other		
<b>Sequence</b>	<b>Field Name</b>	<b>Segment Occurrence</b>	<b>Print</b>	<b>Total/Average/Both</b>
	EMPLOYEE-NAME-10			
	JOB-TITLE			
	ANNUAL-SALARY			
	AGENCY-FEE			
	RELOCATION-EXPENSE			
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>		<b>Actions: (C, P, N, E)</b>
<b>Field Name:</b> HIRE-SOURCE	<b>(not) E G L F</b>	(see option list HR21)	01	<input type="checkbox"/>
<b>Field Name:</b> RESULTING-EMP-STATUS	<b>(not) E G L F</b>	(see option list HR10)	01, 03	<input type="checkbox"/>
<b>Field Name:</b>	<b>(not) E G L F</b>			<input type="checkbox"/>
<b>Field Name:</b>	<b>(not) E G L F</b>			<input type="checkbox"/>

\* This worksheet has been started for you. You must complete with additional information from the letter.

QUERY WRITER

Date: \_\_\_\_\_

Program Name: X		Module Specific? Y / N		
Program Title:		Security? Y / N		
As of Date:		Total Employee Count? Y / N		
Employee Selection Criteria? Y / N		Display Sequence: <input type="checkbox"/> Employee Number <input type="checkbox"/> Other		
Sequence	Field Name	Segment Occurrence	Print	Total/Average/Both
If Employee Selection Criteria:		Compare Value:		Actions: (C, P, N, E)
Field Name:	(not) E G L F			<input type="checkbox"/>
Field Name:	(not) E G L F			<input type="checkbox"/>
Field Name:	(not) E G L F			<input type="checkbox"/>
Field Name:	(not) E G L F			<input type="checkbox"/>

\* This worksheet has been left blank. You must complete with information from the letter.

# SECTION 5: CREATING A PROGRAM TO GENERATE A BATCH REPORT

---

---

## TABLE OF CONTENTS

PLANNING AHEAD: W.I.I.F.M.....	121
THE REPORT WRITER FORM SERIES.....	129
REPORT OPTIONS FORM.....	133
DEFINING THE FIELDS .....	135
SELECTING SEGMENT KEYS .....	155
DEFINING CALCULATION ROUTINES.....	161
CREATING SELECTION CRITERIA LOGIC.....	167
REPORT GROUP ACTIVITIES.....	179
BATCH REPORT PROCESS .....	197
SUBMITTING AND VIEWING REPORTS .....	199
INQUIRY, CHANGING, COPYING, DELETING.....	215
SECTION SUMMARY .....	229
SECTION EXERCISE.....	233

## Section Objectives

- **Determine the format and content of the batch report to be produced**
- **Access and complete the form series for creating a batch report**
- **Schedule the batch report**
- **Initiate the batch report**
- **View the report output**
- **Inquire, change, copy, and delete the batch program**

---

NOTES

---

## **SECTION OVERVIEW**

### **Purpose**

In this section you will learn how to write programs that run in batch mode.

### **Objectives**

When you have completed this section you will be able to do the following:

- Determine the format and content of the batch report to be produced
- Access and complete the form series for creating a batch report
- Schedule the batch report
- Initiate the batch report
- View the report output
- Inquire, change, copy, and delete the batch program

## Requirements Memo

MEMO

TO: Cyborg End-User  
FROM: The Boss  
DATE: August 2, 199X  
SUBJECT: New Report - Projected Salary Increases

---

Beginning on September 1, 199X, we are required to supply salary projection information to the Main Office. I'll need a printed report each week on selected groups of employees who will be receiving increases.

The report will need to show the following items:

Employee number, employee name in full, Department name, employee's most recent salary change date, and employee's annual salary.

The report will need to display the calculated date which is six months greater than the employee's last salary change date. It will also need to display a projected salary based on a 5% increase in their current annual salary. Then show the difference in dollars between their projected salary and their current salary. I want to be able to see at least up to \$99,999.00 for these figures.

Now for totals and such: I need to see totals for their current salaries, the projected salaries, and the difference. I want the report to give me a total after each department. The report needs to be sequenced first by department, and then by employee number.

For this first report, please show only employees who have a job code higher than 34020, and have salary grades higher than 24.

Thanks for your help. If you have any questions, call me at extension 555.

---

### NOTES

---

## PLANNING AHEAD: W.I.I.F.M.

### What's In It For Me?

Like the Solution View Query Writer, the Solution View Report Writer is a report writing tool that helps you easily create The Solution Series Batch reports without actually writing Cyborg Scripting Language code, the Cyborg programming language. Once you have written the report, using the formatted forms provided, you add the report to a report group (RGMSTR) and run it just like the Cyborg-delivered Cyborg Scripting Language reports.

This means that you can create your own batch reports using the Solution View tools. You are not required to know the Cyborg Scripting Language programming language. Still, you will have all the flexibility you need to build your program and print your report.

### Planning the Report

Prior to completing the Report Writer form series that will create the batch program, we suggest that you invest some time in reviewing the original request. This review allows you to define the requirements and the appearance of your report prior to entering the report parameters.

The graphic on the facing page is a sample memo that requests a hard copy report to resolve some information needs. Using this example, it is clear that this manager wants information on selected employees, showing salary projections. Starting with the specifications that are stated on the memo, let us plan what will appear on the report.

### Requirements Worksheet

You might want to use a requirements worksheet to plan your batch report. A worksheet would allow you to research the report content, sequence and detailed specifications before you begin your entries. Once those requirements have been committed to paper, the process of using the Solution View form series becomes an easy next step.

On the next page, we will walk through a sample worksheet that you could use to specify the report requirements.

# Requirements Worksheet

REPORT WRITER

Date: \_\_\_\_\_

Program Name: X _____ P T				Module Specific? Y / N				
Program Title:				Security? Y / N				
As of Date:				Total Employee Count? Y / N				
Employee Selection Criteria? Y / N								
Seq.	Field Name	Segment Occur.	Control Options	Sort Seq.	Print/ Total	Calculation Type Numeric/Date/TimeSpan	Result Length	Result Decimals
						N D T		
						N D T		
						N T		
						N D T		
						N D T		
						N D T		
						N D T		
						N D T		
Calculation Field Name: _____ Formula: (+ * - /)								
=								
=								
=								
If Employee Selection Criteria:			Compare Value:			Action:		
Field Name:			E G L F			<input type="checkbox"/>		
Field Name:			E G L F			<input type="checkbox"/>		
Field Name:			E G L F			<input type="checkbox"/>		

Action Options: C, P, N, E

CONTROL Options: (V)Break Only, (T)otal Break, (P)age Break Only, Total & Page (B)reak  
 PRINT/TOTAL Options: (P)rint, (T)otal, (B)Print & Total, Total & A(V)erage

## NOTES

**PLANNING AHEAD: W.I.I.F.M.,** continued

**Using the Worksheet** Your worksheet might look something like the sample on the facing page. The worksheet contains the following specifications:

- **Program Name**—What is the name of the Batch Report program that will produce the hard copy report? As you recall, all user-defined program names are six characters and must begin with X. For batch reports, the program name must also end in PT.
- **Program Title**—What is the heading that will appear at the top of each page of the report? There is a 30-character limit for a program title.
- **As Of Date**—Do you want the most current data or data as of a particular date which you specify in this space. The date will be used as a Report Schedule parameter?
- **Employee Selection Criteria?**—Will you want to specify criteria for selecting employees on this online report?
- **Module Specific?**—Do you want to specify a Module ID to be associated with this report?
- **Security?**—Do you wish to specify the security to be associated with this report?
- **Total Employee Count**—Do you want to see a total employee count displayed at total breaks and at the end of the report?
- **Sequence**—In what sequence should the fields appear on the report from left to right?
- **Field Names**—What are the field names (20 characters) that will be used in the report? Field names also provide the headings for the detail. The field names must have been previously defined unless they are fields generated by a calculation routine in this report. The total of the fields or the field headings may not exceed 132 characters.

Some of the utilities available to help in your research are Search By Full Name (WORD), Search Field Name Table for Field Name (SEARCH), Field Documentation Display (FIELDS), Field-Name Table Cross Reference Menu (F-MENU), and Screen Label To Field Name Cross Reference (FLABEL). These utilities will be covered later in this section.

## Requirements Worksheet, continued

REPORT WRITER

Date: \_\_\_\_\_

Program Name: X _____ P T					Module Specific? Y / N			
Program Title:					Security? Y / N			
As of Date:					Total Employee Count? Y / N			
Employee Selection Criteria? Y / N								
Seq.	Field Name	Segment Occur.	Control Options	Sort Seq.	Print/ Total	Calculation Type Numeric/Date/TimeSpan	Result Length	Result Decimals
						N    D    T		
						N    D    T		
						N            T		
						N    D    T		
						N    D    T		
						N    D    T		
						N    D    T		
						N    D    T		
Calculation Field Name: _____ Formula: (+ * - /)								
=								
=								
=								
If Employee Selection Criteria:			Compare Value:			Action:		
Field Name:			E G L F			<input type="checkbox"/>		
Field Name:			E G L F			<input type="checkbox"/>		
Field Name:			E G L F			<input type="checkbox"/>		

Action Options: C, P, N, E

CONTROL Options: (V)Break Only, (T)otal Break, (P)age Break Only, Total & Page (B)reak  
 PRINT/TOTAL Options: (P)rint, (T)otal, (B)Print & Total, Total & A(V)erage

---

### NOTES

---

PLANNING AHEAD: W.I.I.F.M., continued

Using the Worksheet,  
continued

- **Segment Occurrence**—For those fields which reside in a multiple-occurrence segment, which segment key(s) should be used for the report? You can specify a single segment key, a range of keys, and/or a list of keys.
- **Control Options**—Should a change in value for this field cause a page or total break on the report? If so, this field also becomes a part of your sort sequence.
- **Sort Sequence**—Should this field be included in determining the order of your report? If so, give this field a number indicating its sequence (from major to minor) in the sort key. Total of all fields used in the sort may not exceed 47 characters or 6 fields (whichever is greater). Sort fields need not print.
- **Print/Total**—Should this numeric field print on the report? Should this field provide a total or average on the report? You may not want to print the field, but you can still indicate a total or average of the field at the end of the report.
- **Calculation Type**—Is this a field that should be calculated? If so, what method should be used? (Time Span Calculation is to be used when you are defining a formula of your own.) For calculation fields, fill in the next three characteristics.
- **Result Length**—What length is the resulting calculation field?
- **Result Decimals**—How many decimal places will appear in this numeric calculation result?
- **Calculation Field Name/Formula**—Repeat each calculation field and specify the formula for calculating it.
- **If Employee Selection Criteria**—What employee selection criteria should be used to determine which employees will appear in the report? (For example: salary ranges, date of hire, job codes.) Specify the field names and the values that should be used for comparison. Use the E (equals), G (greater than), or L (less than) indicators to make your comparison. Use F (found) to select employees based on the presence of a specific segment on their Employee Database. You may specify NOT prior to your indicators.
- **Action**—You will use this field later when processing the Solution View forms.

## Requirements Worksheet, continued

REPORT WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X P R O P T				<b>Module Specific?</b> Y / N NO						
<b>Program Title:</b> PROJECTED 6 MOS SALARY INCREASE				<b>Security?</b> Y / N NO						
<b>As of Date:</b> NO				<b>Total Employee Count?</b> Y / N YES						
<b>Employee Selection Criteria?</b> Y / N YES										
Seq.	Field Name	Segment Occur.	Control Options	Sort Seq.	Print/ Total	Calculation Type Numeric/Date/ TimeSpan			Result Length	Result Decimals
1	EMPLOYEE-NUMBER			2	P	N	D	T		
2	EMPLOYEE-NAME				P	N	D	T		
3	CONTROL-3	01	TOTAL	1	P	N		T		
4	SALARY-EFFECTIVE				P	N	D	T		
5	ANNUAL-SALARY				B	N	(D)	T		
6	INCREASE-DATE				P	N	D	T		
7	PROJECTED-SALARY				B	(N)	D	T	10	2
8	DIFFERENCE-IN-SALARY				B	(N)	D	T	9	2
<b>Calculation Field Name:</b>									<b>Formula: (+ * - /)</b>	
INCREASE-DATE									= SALARY-EFFECTIVE + 6 MONTHS	
PROJECTED-SALARY									= ANNUAL-SALARY * 105%	
DIFFERENCE-IN-SALARY									= PROJECTED-SALARY - ANNUAL-SALARY	
<b>If Employee Selection Criteria:</b>			<b>Compare Value:</b>			<b>Action:</b>				
Field Name: JOB-CODE			E G L F G 34020						<input type="checkbox"/> N	
Field Name: SALARY-CODE			E G L F G 24						<input type="checkbox"/> P	
Field Name:			E G L F						<input type="checkbox"/>	

Action Options: C, P, N, E

CONTROL Options: (V)Break Only, (T)otal Break, (P)age Break Only, Total & Page (B)reak  
 PRINT/TOTAL Options: (P)rint, (T)otal, (B)Print & Total, Total & A(V)erage

### NOTES

**PLANNING AHEAD: W.I.I.F.M., continued**

**Check Your Requirements**

The completed worksheet on the facing page is an example of specifications that will be used to define the report to be created using the Report Writer form series. When you have completed your worksheet, you will have a draft version of the report and a list of the fields and their associated characteristics.

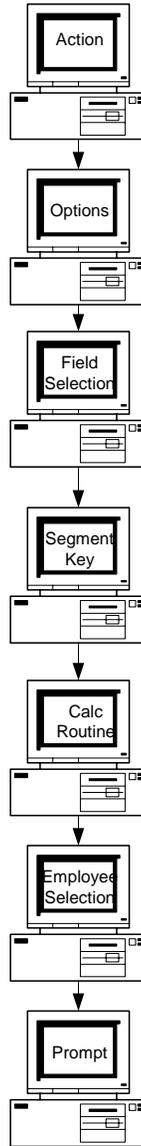
From what we have specified here, we are able to deduce:

- We have to select certain employees for the report
- We have to calculate some fields for the report
- We have to sequence the data and do some totaling

**What's Next?**

Now you are ready to use the Solution View tools to enter your program requirements. We will start by previewing the form series for a Batch Report program.

## Query Writer Form Series



---

### NOTES

---

## THE REPORT WRITER FORM SERIES

### Report Writer Forms

The Report Writer is used to write programs that are scheduled and then run in batch mode. The data you request is printed on hardcopy, and batch reporting allows you to calculate fields within the report. You specify the order in which the data will appear and where page and total breaks will occur.

These are the forms, required and optional, that are available in Solution View to create your report program.

- **Action form**—used to name the report program, give the report a title, and select the type of action to be performed.
- **Report Options**—used to indicate an As Of date, count employees, and indicate that you want employee selection criteria.
- **Report Field Selection form**—used to define the fields to be displayed on the batch report.
- **Stacked Segment Key form**—(optional) used to tell the system which key you will use for selecting a specific multiple-occurrence segment where the key is not a date.
- **Calculation Routine Entry form**—(optional) defines the calculations for the fields you specified on the Report Field Selection form.
- **Employee Selection Criteria form**—(optional) used to determine which employees will be included in the report.
- **Prompt form**—informs you that your program has been written and compiled successfully.

**Note:** The **Copy** and **Change Option** forms also exist within the Solution View Report facility. These are secondary forms and are not required to create a report program. They are discussed later in this section.

Now that you have previewed the form series that you will use to create the batch report, let us look at each task in detail and see what tools are provided on each form and how to use them.

## The Action Form

Solution View Tool Kit      Option:

Name of Program: XPROPT      Title: PROJECT 6 MOS SALARY INCREASES

**Program Type**

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

**Action**

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

---

### NOTES

---

## SELECTING THE ACTION

**Action Form** The first Report Writer form, the Action form, is used to identify the Writer program. It takes you directly to the Report Writer form series.

### Accessing the Form

**Selection:** \_\_\_\_\_

**Step:**

**Navigator**



User Tools

User Tools



Solution View

**Result:** The Solution View Action form displays.

### Field Descriptions

The Action form field descriptions have already been presented in Section 3. In this section you will learn how to use those fields to access the Report Writer series of forms.

### Using the Action Form

To add a Report program:

**Selection:** \_\_\_\_\_

**Step:**

**Name of Program**

1. Type the one- to six-character name of the program, beginning with X, (for example, XPROPT).

**Title**

2. Type a title for the report. This title shows on each page of the Query Report (optional).

**Program Type**

3. Select Report.

**Action**

4. Select Add.

**Module**

5. Select the module ID to be associated with the report (optional).

**Security Code:**

6. Type the Security Code for the report (optional).



7. Click Save or press ENTER.

**Result:** When the Action form is error-free, the Report Options form displays.

### Practice

Using the worksheet, please complete this form for your program.

## Report Options Form

Solution View Report Writer: XPROPT      Option: [dropdown menu]

Report Options

**As Of Date Options**

- Use Current Date
- Use Entered Date
- Use Entered Range

**General Options**

- Selection Criteria
- Count Employees

---

### NOTES

---

## REPORT OPTIONS FORM

- Report Options Form** The second form in the Report Writer form series, the Report Options form, is used to indicated As Of Date options, selection criteria and gives you the ability to count employees.
- Field Descriptions** The Report Options form contains the following fields:
- **As Of Date**—for dated segments, used to specify the date up to which information is to be processed. The segment is included if its date is equal to or prior to this date. The date must be entered in century format, (for example, 12-09-1999 or 19991209). The date defaults to the current date if no entry is made.  
  
The entered date or date range will be established in the Schedule Report Groups when you schedule this report.  
  
**Note:** You can use an As-of date from any form that has the date as the first key field, such as the Job Assignment/Changes form (05-SCR).
  - **Selection Criteria**—used to add an Employee Selection Criteria form to the series of Report Writer forms to indicate your answer to the question: “Do you want to narrow the list of employees based on one or more specific selection criteria?”  
  
When a new Report program is being added, the default for this option field indicates that employee selection criteria is desired.  
  
If you do not want to enter selection criteria, click the box to clear this option.
  - **Count Employees**—used to indicate your answer to the question: “Do you want a count of employees who appear on the report?”  
  
When a new Report program is being added, the default for this option field indicates that employees will be counted.  
  
If you do not want an employee count, click the box to clear the option.
  - **Composite View or Single Field View**—specifies the next form display for Field Selection.  
  
This Single Field View form is discussed in the Extra for Experts appendix.
- Practice** Using the worksheet, please complete this form for your program. Click Composite View for executing the form.

## Report Field Selection Form

Solution View Report Field Selection: XPROPT Option:

	Seq Nbr	Field Name	Control Break	Sort Seq	Print/ Total	Calc Type	Result Length	Result Decimals
<input type="checkbox"/>	00		<input type="checkbox"/>					
<input type="checkbox"/>	05		<input type="checkbox"/>					
<input type="checkbox"/>	10		<input type="checkbox"/>					
<input type="checkbox"/>	15		<input type="checkbox"/>					
<input type="checkbox"/>	20		<input type="checkbox"/>					
<input type="checkbox"/>	25		<input type="checkbox"/>					
<input type="checkbox"/>	30		<input type="checkbox"/>					
<input type="checkbox"/>	35		<input type="checkbox"/>					

Display Values

Control Break     Print/Total     Calculation Types     All

---

### NOTES

---

## DEFINING THE FIELDS

**Report Field Selection Form** The third form in the Report Writer form series, the Report Field Selection form, is used to specify which fields are to appear on the report, set up totals for the report, and specify the fields that are to be used in calculations.

**Field Descriptions** The Report Field Selection form contains the following fields:

- **Command Field Action Code**—used to add, change, or delete the data field names to be printed on the report. This field is at the far left of the form and is one character in length. It is the only field on the form that does not have a field heading above it.

When the Report Field Selection form displays, an A is automatically placed in each Command field so you do not have to key it in for each field name you are adding to your report. Other valid entries for this field are C for Change, D for Delete, and P for Position.

- **Seq Nbr**—requires that you specify a unique sequence number for each field. The Sequence Number determines the order of the fields on the report. Valid sequence numbers are 00–97.

When the Report Field Selection form is displayed, a sequence number incremented by 5 is automatically placed in each Sequence Number field so you do not have to key one in for each field name you are adding to your report. You may wish to increment additional sequence numbers by 5's so you can insert additional fields later.

If a calculation uses the result of a prior calculation, these calculated fields must be listed on this form in logical result order. For example, you would want to sequence the amount of a salary increase prior to using it to calculate the new annual salary amount.

## Report Field Selection Form, continued

Solution View Report Field Selection: XPRDPT Option: **Field Selection Menu**

Seq Nbr	Field Name	Control Break	Sort Seq	Print/Total	Calc Type	Result Length	Result Decimals
A 00							
A 05							
A 10							
A 15							
A 20							
A 25							
A 30							
A 35							

**Go To Next Step**

Display Values

Control Break    Print/Total    Calculation Types    All

Solution View Report Writer: XPRDPT Option:

Field Name Table Main Menu Page: 1

- Clear Selection
- Company Fields
- Table Fields
- Other Record Fields
- User Quick Reference
- Employee Fields
- Time Entry/Adj Flds
- Work Fields
- Quick Reference Flds

### NOTES

DEFINING THE FIELDS, continued

Field Descriptions,  
continued

- **Field Name**—used to specify the names of the data fields to appear on the report. The total length of all fields entered on the report cannot exceed 132 characters. A reject message informs you if you exceed the limit. (Take into consideration that WRITER will add one character to each field as a spacer before the next field.)

The data field name entry must be valid on the Field Name Table, unless you are creating a field name to be used by, or as a result of, a calculation in the report, (for example, **NEW** Annual Salary).

Your security assignment determines which fields you can access. If your program includes fields to which you are denied access, you will be unable to reload your report.

You may select either employee master fields or history fields or labor fields on a report, but not any combination of the three. On an employee master report, history/labor fields may not be selected. On a history/labor report, some employee master fields may be selected.

Tools for finding  
Field Names

**Field Name Table**—accessing this table in a menu-driven format can be done by placing question marks (?) in the fields and clicking Save or pressing ENTER.

For each question mark (?) entered you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing an X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

You can access field help while in the Field Name Table Menu by placing a question mark (?) in the entry box next to the field. To leave the menu select Leave Field Help from the Option field to return to the Field Name Table Menu.

After selecting your data field name entries from the Field Name Table Menu, select Leave Menu from the Option field to return to the Query Field Selection form.

**Note:** When you are accessing the Field Name Table you are accessing the table for The Solution Series, not the Writer tools. Every Solution Series field will be displayed, but not every field is accessible to Solution View.

## Tools for finding Field Names

Search Field Name Table for Field Name SEARCH

The SEARCH program allows you to search for a field name stored on the Field Name Table. SEARCH displays a list of names based on a partial entry in the Key fields.

Complete the text boxes below, then hit enter:

Enter the WORD/VERB/COMMAND:

**SEARCH**

**FLABEL**

Screen Label To Field Name Cross Ref. FLABEL

This program will list the Field Name and associated label of a screen.

Complete the text boxes below, then hit enter:

Enter Screen Name:

---

**NOTES**

---

## DEFINING THE FIELDS, continued

Tools for finding  
Field Names,  
continued

**Search Field Name Table (SEARCH)**—The program SEARCH can also be used without accessing the Field Name Table in the menu driven format. This program can be accessed by selecting:

**Selection:****Step:****Navigator**

Development Tools  
Fields and Verbs



Search by Partial Name

The system will prompt for the word or field name that you wish to find. Once the form is executed, a list of field names will be returned based on your entry.

**Form Label to Field Name Cross-Reference (FLABEL)**—The program FLABEL is a good tool if you know which form you want to pull information from. The FLABEL program will list the form label as it appears on the form and then the corresponding field name to the right. To access this utility select:

**Navigator**

Development Tools  
Fields and Verbs



Display field Names for a form

The system will prompt for a form name (for example, 40-SCR). Once the form is executed, the system will return the cross-reference list.

Other utilities include Field Documentation Display (FIELDS) and Field Name Table Cross-Reference Menu (F-MENU).

## Tools for finding Field Names, continued

---

NOTES

---

**DEFINING THE FIELDS**, continued

**Tools for finding** To get the definition of a field the user should select the relevant field on the form

**Field Names**, and click What's This.

continued

**Selection:**



**Step:**

1. With relevant field on the form selected, click What's This.

**Result:** The pointer changes to an arrow with a question mark beside it.

2. Click the field again.

**Result:** A Windows Help form appears with the field details.

## Report Field Selection Form

Control Breaks	
Y - Break Only	P - Page Break Only
T - Total Break	B - Total and Page Break

---

### NOTES

---

## DEFINING THE FIELDS, continued

### Field Descriptions, continued

Only eight data fields can be entered and viewed on the Report Field Selection form at one time. To add more fields after the first eight have been keyed and accepted, you have these options:

- Type P in the Command field with a sequence number greater than any on the form to return a clear form to allow entry of additional data fields.
- Type P in the Command field of the last data field to position that data field at the top of the Report Field Selection form.
- Type A in the Command field and a unique sequence number over an existing sequence number.

After you have entered one or more forms of field names, you may page backward or forward through them using these options:

-  ■ To return to the first form of entries click the Top of selection list icon.
-  ■ Use the Move up or Move down icons to view the previous or next eight fields on the report.

**Control Break** (option list SC16)—allows you to establish control breaks on a report, for example, print a total or start a new page **when a specified value of a sort key changes**, such as Division or Department.

Control Break options can be displayed by clicking the Control Break button or type a ? in the field.

A field that you set up as a Control Break field **MUST** be included as part of your Sort Key, which is discussed next. For example, when doing a control break on Department, the report data must be in department sequence.



DEFINING THE FIELDS, continued

Field Descriptions,  
continued

- **Sort Seq**—defines the fields to build the Sort Key. The Sort Key determines the order in which the employees will appear on the report (the Report code and Control 1-2 are predefined by the system as the highest level sort sequences). **The sort key field(s) need not appear on the report.**

For example, you may want the report to be in Employee Number order. Or maybe you want the report to be in order by Pay Frequency and then by Employee Number. You designate the order by numbering the fields here. Valid entries are 0–9.

At least one data field must have an entry in the Sort Seq field. You can have *up to six sort fields for a report*. A data field cannot have the same sort sequence as another data field.

Once you have specified the fields to include in the Sort Key, the facility calculates the sort length. The sort length is calculated by totaling the storage length of all the data fields included in the Sort Key. Valid sort length values are between 12 and 58, inclusive.

Twelve is the minimum length because of the default sort, which is always present. The default sort includes the following 11 characters:

- Forms/Report Code (length of 5)
- Control 1 (length of 2)
- Control 2 (length of 4)

Therefore, do not define a Sort Key with a sort length that exceeds 47 characters because of the eleven character default sort ( $11 + 47 = 58$ ). However, your Sort Key must be at least one character, because you must include AT LEAST ONE of the fields from the Field Name selection in the Sort Key.

It is strongly recommended that you include the Employee-Number field as the last Sort Key to make the sort key unique. This is a unique field for each of your employees and is usually included when reporting anyway. You may then choose any other sort fields you wish (to a limit of 47 characters in total).

When accessing History or Labor data, use the Employee-Number field as the first sort field, and the Master-Number field as the last sort field to assure a unique sort key. The Master-Number need not print.

## Report Field Selection Form, continued

Print/Total Options	
P - Print Only	V - Total and Average
T - Total Only	A - Print/Total/Average
B - Print and Total	

Calculation Types		
N - Numeric	D - Date	T - Time Span

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Field Descriptions,**  
continued

- **Print/Total** (option list SC17)—specifies whether a **numeric** field is to be printed, totaled, or both. The valid entries can be displayed at the bottom of the form, or type a ? in the field.

If you leave the Print/Total field blank, no printing or totaling of the data field occurs. The field can be left blank when designating a field as sort key data only, or when defining an interim field for a calculation, which is discussed next.

- **Calc Type** (option list SC18)—allows you to enter your own unique data field name in the Field Name and specify the type of calculation in this field. The valid entries can be displayed at the bottom of the form, or type a ? in the field.

A Calculation Routine Entry form is returned for each calculated data field defined on this form. Use this form to define how to calculate the field. You must also enter the Result Length and Result Decimals fields on this form, discussed next.

## Report Field Selection Form, continued

### Fields in YYMMDD Format

AGE =	Current Date - Employee Birth Date
LENGTH-OF-SERVICE = (or if Terminated) =	Current Date - Employment Date Date of Termination - Employment Date
ADJUSTED-SERVICE = (or if Terminated) =	Current Date - Adjusted Seniority Date Activity Date - Adjusted Seniority Date
ORIGINAL-SERVICE = (or if Terminated) =	Current Date - Original Hire Date Activity Date - Original Hire Date

### Fields in YY Format

YEARS-OF-AGE =	Current Date - Employee Birth Date
SERVICE-YEARS = (or if Terminated) =	Current Date - Employment Date Date of Termination - Employment Date
ADJUSTED-SERVICE-YRS = (or if Terminated) =	Current Date - Adjusted Seniority Date Activity Date - Adjusted Seniority Date
ORIGINAL-SERVICE-YRS = (or if Terminated) =	Current Date - Original Hire Date Activity Date - Original Hire Date

---

## NOTES

---

## DEFINING THE FIELDS, continued

### Field Descriptions, continued

There are eight fields you can enter on the Report Field Selection form that are already programmed to automatically calculate time-spans. A time-span is the difference in time between two dates. **DO NOT define these fields as calculated fields by typing T in the Calc Type field.** Solution View knows how to calculate these fields.

Four of these predefined time-span fields are calculated in YYYYMMDD format. The fields, along with the method by which they are calculated, are displayed in the graphic on the facing page.

The fields are:

- AGE
- LENGTH-OF-SERVICE
- ADJUSTED-SERVICE
- ORIGINAL-SERVICE

The other four predefined time-span fields are calculated using the same methods but in a two-digit year format, (for example, YY). These fields are:

- YEARS-OF-AGE
- SERVICE-YEARS
- ADJUSTED-SERVICE-YRS
- ORIGINAL-SERVICE-YRS

The above fields can also be used on the Employee Selection Criteria form to create report selection logic.

**Note:** All of the above fields can be used as they are, with no T in the Calc Type field.

A Calculation Routine Entry form is returned for each calculated data field defined on this form. The Calculation Routine Entry form is where you define how to calculate the field.

## Report Field Selection Form, continued

	Result Length Rules
Numeric Fields -	01 - 14 Default value is 09.
Time Spans -	02 (yrs) 04 (yrs/mos) 06 (yrs/mos/days) Default value is 06.
Date Fields -	10 only

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Field Descriptions**,  
continued

- **Result Length**—on fields to be calculated, used to enter the length of the calculated data field.

Do not include decimals or commas in the resulting length.

- **Result Decimals**—used to define the number of decimal places to the right of the decimal point for numeric calculations.

Valid entries for this field are 0–6.

The default value is 0.

## Report Field Selection Form, continued

Solution View Report Field Selection: XPROPT Option: ▼

Seq Nbr	Field Name	Control Break	Sort Seq	Print/Total	Calc Type	Result Length	Result Decimals
<input type="checkbox"/> A 00	EMPLOYEE-NUMBER	<input type="checkbox"/>	2	<input type="checkbox"/> P	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 05	EMPLOYEE-NAME	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> P	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 10	CONTROL-3	<input checked="" type="checkbox"/> T	1	<input type="checkbox"/> P	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 15	SALARY-EFFECTIVE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> P	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 20	ANNUAL-SALARY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> B	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 25	INCREASE-DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> P	<input type="checkbox"/> D	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> A 30	PROJECTED-SALARY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> B	<input type="checkbox"/> N	10	2
<input type="checkbox"/> A 35	DIFFERENCE-IN-SALARY	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/> B	<input type="checkbox"/> N	09	2

Display Values

Control Break   
  Print/Total   
  Calculation Types   
  All

---

### NOTES

---

## DEFINING THE FIELDS, continued

Using the Report  
Field Selection Form

<u>Selection:</u>	<u>Step:</u>
<b>Action</b>	1. Press TAB to accept the A (Add) or type C (Change) or D (Delete).
<b>Seq Nbr</b>	2. Type a two-digit sequence number to specify the order in which the field will appear from left to right on the report <b>OR</b> press TAB to accept the sequence number already in place.
<b>Field Name</b>	3. Type the name of the first field to be used in the report.
<b>Control Break</b>	4. Type a Control Break Option to control the paging or totaling of the report (optional).
<b>Sort Sequence</b>	5. Type a number representing the sort sequence of the displayed data (optional).
<b>Print/Total</b>	6. Type the Print/Total Option, if any.
<b>Calc Type</b>	7. Type N (Numeric), D (Date) or T (TimeSpan) if this field is to be created as the result of a calculation.
<b>Result Length</b>	8. Enter the length of the calculated data field without commas or decimals.
<b>Result Decimals</b>	9. Enter the number of decimal places to the right of the decimal point for numeric calculations.
	10. Repeat steps 1 through 9 until all fields have been listed.
	11. Click Go To Next Step to continue to the next form.

Errors on this form must be resolved before proceeding. You may either fix the error, or click Cancel to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and press ENTER to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

### The Stacked Segment Key Form

Solution View Report Key Values: XPROPT      Option:  

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	<input type="checkbox"/>	<input style="width: 100%;" type="text"/>	<input type="text"/>
<input style="width: 100%;" type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input style="width: 100%;" type="text"/>	<input type="text"/>
<input style="width: 100%;" type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input style="width: 100%;" type="text"/>	<input type="text"/>

**EGL Values**

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

**Actions**

N = Next: Check next key field  
P = Proceed: No further checks performed

---

### NOTES

---

## SELECTING SEGMENT KEYS

### Stacked Segment Key Form

The Stacked Segment Key form is only displayed when a field name entered on the Report Field Selection form is in a multiple occurrence (stacked) segment whose primary Key field is not a date. This is the only time this form is used. The form is displayed automatically for each segment (not field) you listed on the Report Field Selection form; you do not need to request it.

Solution View assumes you want to display the requested field's data from the first occurrence in the stacked segment. If another occurrence is desired, the criteria must be entered on this form. You may bypass this form by pressing ENTER, or clicking Bypass this form. (This action assumes the default, the first occurrence.)

The purpose of this form is to include/exclude segment **occurrences** in the program. If the employee does not have the segment of requested key data, this alone does not exclude the **employee** from the report.

### Field Descriptions

The Stacked Segment Key form contains the following fields:

- **Requested Detail Line Field**—an inquiry-only field that displays a field name from the Query Field Selection form if that field occurs in a multiple-occurrence segment whose key is not a date.

Only the first requested field of a multiple occurrence segment is displayed for the selection of the segment key(s). All other fields from the same segment use the same comparison logic.

- **Required Keys**—displays the key field name related to the fields selected on the Query Field Selection form. The key field name(s) used as the basis for selecting occurrences is placed here automatically by Solution View. **Do not change this field.**
- **Not**—used to indicate that the selection criteria checks for a condition that does NOT exist. Check the box, or enter N in this field, to create selection criteria that checks for a condition that does NOT exist. The default entry of blank is interpreted in the positive, in other words, *is*.

## The Stacked Segment Key Form, continued

Solution View Report Key Values: XPROPT    Option:   

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

Selecting the First Occurrence

Selecting a Specific Occurrence

Solution View Report Key Values: XPROPT    Option:   

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	02	<input checked="" type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

---

### NOTES

---

**SELECTING SEGMENT KEYS**, continued

**Field Descriptions,**  
continued

- **EGL** (option list 21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are:
  - E or = (equal to)
  - G or > (greater than)
  - L or < (less than)
  
- **Literal Compare Value**—used to specify a literal value to be compared to the Required Keys field.

Ranges can be entered using slashes (/) to indicate an inclusive range, or semicolons (;) to indicate an OR single or range condition. The EGL field must be an E (=) to use ranges or OR conditions.

- To specify HEDs from 001 through 500:

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER		E	001/500

- To specify HED 500 or HED 510:

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER		E	500;510

- To specify HEDs from 001 through 500, or HED 510:

<b>Required Keys</b>	<b>Not</b>	<b>EGL</b>	<b>Literal Compare Value</b>
HED-NUMBER		E	001/500;510

**BATCH PROGRAMS DISPLAY MULTIPLE LINES PER EMPLOYEE IF MORE THAN ONE MATCH IS FOUND.**

**Note:** A Key field may not contain spaces. Therefore, a literal entry of SPACES in the Literal Compare Value field is invalid.

- **Action** (option list SC22)—allows you to specify whether to include/exclude the segment occurrence being compared.

N in the field causes a segment that meets the condition to be retained if it also meets the next statement(s).

P in the field causes a segment that meets the condition to be retained.

## The Stacked Segment Key Form, continued

Solution View Report Key Values: XPROPT      Option:

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	E	01	P
	<input type="checkbox"/>			
	<input type="checkbox"/>			

EGL Values

- Equal To: E or =
- Greater Than: G or >
- Less Than: L or <

Actions

- N = Next: Check next key field
- P = Proceed: No further checks performed

### NOTES

## SELECTING SEGMENT KEYS, continued

**Using the Stacked Segment Key Form**

The procedure for using the Stacked Segment Key form to select a segment occurrence is:

To select the **first occurrence** of the stacked segment, click Bypass this form or press ENTER.

To select **any other occurrence**, perform the following steps:

<u>Selection:</u>	<u>Step:</u>
<b>Required Keys</b>	1. Do not change this field; it is automatically updated by the Report Writer.
<b>Not</b>	2. Optionally, indicate N to designate the negative effect of the entry in the EGL field.
<b>EGL</b>	3. Type E (Equal), G (Greater), L (Less), or the symbolic equivalents, =, >, or <, to establish a conditional comparison.
<b>Literal Compare Value</b>	4. Type the literal value to be compared with the corresponding Required Keys field.
<b>Action</b>	5. Type P (include in process) or N (check next comparison) to designate the selection of the specific occurrence.
	6. Click Save or press ENTER to move to the next form in the process.
	7. Repeat steps 1 through 6 for each segment key requested.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field and press ENTER to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

## Calculation Routine Entry Form

Solution View Report Writer: XPROPT      Option:

Calculation Routine Entry

Result Field		Factor/Field Name	
INCREASE-DATE	=	<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>

Result: Date      Resulting Date: YYYYMMDD

---

### NOTES

---

## DEFINING CALCULATION ROUTINES

**Calculation Routine Entry Form** This form defines calculations within the report. A Calculation Routine Entry form is returned for each data field name you specified a Calc Type for on the Report Field Selection form.

**Note:** A data field defined on a previous Calculation Routine Entry form may be used in subsequent calculations.

**Field Descriptions** The Calculation Routine Entry form contains the following fields:

- **Result, Result Field, Resulting format**—are all returned in Inquiry mode and are based on the fields you entered on the Report Field Selection form. The type of Result for the calculated field displays in the upper right corner. The Result Field is in the middle of the form. The format of the Resulting field is located above the line at the bottom of the form.
- **Factor/Field Name**—allows you to define a formula to calculate by typing in factors and field names. Use this field to enter the field names or literal needed to perform the calculation.

You may select the field names to use in the formula by typing in the field name(s), or accessing the Field Name Table Menu.

- **Operation** (option list SC20)—defines the mathematical operations used in the calculation. This field is located to the right of the Factor/Field Name field and is not labeled. The valid entries, listed at the bottom of the form, are:
  - + (Add)
  - - (Subtract)
  - \* (Multiply)
  - / (Divide)

**Note:** The = operation located to the right of the Result Field is always present on this form; it is returned in Inquiry mode and may not be changed or removed. Also, the = operation may not be used in any other Operation field on this form.

## Calculation Routine Entry Form, continued

Calculation Operations	
Result Field -----	Factor/Field Name -----
NEW-SALARY	= ANNUAL-SALARY * 1.065 + 35.00
NEW-SALARY	= (Annual-Salary times 106.5%) plus 35.00

---

### NOTES

---

DEFINING CALCULATION ROUTINES, continued

Field Descriptions,  
continued

**SPECIAL NOTES ON CALCULATIONS:** Calculations are performed in the order they appear on the Calculation Routine Entry form regardless of the operation(s) being used. In the example on the opposite page, you can see that the first operation is completed prior to starting the second stage of the calculation.

When using percentages as part of your calculation, remember that the % symbol may only be used when the factor is a whole number. For example, you cannot specify a factor of 106.5% to express one hundred and six and a half percent. The correct entry is 1.065.

Let us look at a few more examples:

4½% is expressed as .045 (not 4.5%)  
7% can be expressed as .07 or 7%  
105% can be expressed as 1.05 or 105%

As a rule, you may:

- use a percent sign with whole numbers only
- not use BOTH a decimal point and a percent sign for a whole number

For customer-defined timespan calculations:

*Timespans* can be expressed within the timespan calculation as follows:

- 6 digits 010305                      meaning 1 yr, 3 mos, 5 days  
    or 0103                            meaning 1 yr, 3 mos  
    or 03                                meaning 3 yrs
- 6 digits w/dashes 01-03-05      meaning 1 yr, 3 mos, 5 days  
    or 1-3-5                            meaning 1 yr, 3 mos, 5 days  
    or 1-3                                meaning 1 yr, 3 mos

*Dates* can be expressed within a timespan calculation as follows:

- 01-01-1999 = the calculation should use January 1, 1999
- 19990101 = the calculation should use January 1, 1999

## Calculation Routine Entry Form, continued

Solution View Report Writer: XPROPT      Option:

Calculation Routine Entry

Result Field	Factor/Field Name
INCREASE-DATE	= SALARY-EFFECTIVE      Add      +
	00-06-00

Result: Date      Resulting Date: YYMMDD

Solution View Report Writer: XPROPT      Option:

Calculation Routine Entry

Result Field	Factor/Field Name
PROJECTED-SALARY	= ANNUAL-SALARY      Multiply      *
	105%

Result: Numeric      Resulting Field Length: 10      Number of Decimals: 2

Solution View Report Writer: XPROPT      Option:

Calculation Routine Entry

Result Field	Factor/Field Name
DIFFERENCE-IN-SALARY	= PROJECTED-SALARY      Subtract      -
	ANNUAL-SALARY

Result: Numeric      Resulting Field Length: 09      Number of Decimals: 2

### NOTES

## DEFINING CALCULATION ROUTINES, continued

**Using the Calculation Routine Entry Form**

To use the Calculation Routine Entry form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Result Field</b>   | 1. Do not enter. Display-only field.  |
| <b>Factor/Field Name</b>  | 2. Type the first factor or field name in the calculation.  |
| <b>Operation</b>  | 3. Select the Calculation Option to be performed between the previous Factor/Field Name and the next Factor/Field Name. |
| <b>Factor/Field Name</b>  | 4. Type the next factor or field name calculation.  |
|   | <b>Note:</b> Calculation fields previously defined in this report may be used in the current calculation.               |
|   | 5. Repeat steps 3 and 4 for each arithmetic operation to be performed.  |
|  | 6. Click Save or press ENTER to move to the next form in the process.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or click Cancel to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and execute the form to return to the Action Form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete these forms for your program.



## CREATING SELECTION CRITERIA LOGIC

### Employee Selection Criteria Form

The Employee Selection Criteria form allows you to narrow the number of employees that will be selected for the report. For example, you may want to display only those employees with certain salary ranges, hire dates, and so forth, on the form. This is done by setting a condition on one or more fields. The field(s) does not have to be one of the fields that appears on the Report Field Selection form.

Fields that you created for use in calculations may **NOT** be used on this form, or an error will result.

The use of the Employee Selection Criteria form is optional. This form only appears if it is chosen from the Report Options form.

**Note: Criteria used to include/exclude employees on this form may not be inconsistent with segment key selections that were entered on the Stacked Segment Key form.** For example, if you selected HED occurrences **not** equal to 600, then 600 should not be included here for employee selection.

### Field Descriptions

The Employee Selection Criteria form contains the following fields:

- **Field Name**—data field name(s) used as the basis for employee selection. The fields on this form do not have to be displayed on the report. Your security assignment determines which fields you can access.

To access the Field Name Table in a menu-driven format, select Field Selection Menu from the Option Field or a ? in the Field Name field to access the Field Name Table Menu to select the data field name(s). For each ? you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

- **Not**—to create selection criteria that checks for a negative condition, in other words, a condition that is not true.

For example, if you want only employees who do NOT have an annual salary greater than \$28,000 to appear on the report, you would indicate “Not” in this field (and the appropriate EGL, Literal Compare Values, and Action fields, discussed next).

## The Employee Selection Criteria Form Examples of Ranges, OR Conditions, and Found/Not Found Logic

### ■ Salary Grades SG25 or SG28 or SG35

Field Name	Not	EGLF	Literal Compare Value
SALARY-GRADE		E	SG25;SG28;SG35

### ■ Termination Dates ranging from 01-01-91 through 06-01-91

Field Name	Not	EGLF	Literal Compare Value
DATE-OF-TERMINATION		E	01-01-91/06-01-91

### ■ Job Codes ranging from 10500 through 30650 or equal to 34020

Field Name	Not	EGLF	Literal Compare Value
JOB-CODE		E	10500/30650;34020

### ■ Salary Segment for June 1, 1993, Increased Hours

Field Name	Not	EGLF	Literal Compare Value
ANNUAL-SALARY		F	930601;9;S06

### ■ A value of spaces is valid. Example:

Field Name	Not	EGLF	Literal Compare Value
UNION-CODE		E	SPACES

---

## NOTES

---

## CREATING SELECTION CRITERIA LOGIC, continued

Field Descriptions,  
continued

- **EGLF** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are:
  - E or = (equal)
  - G or > (greater than)
  - L or < (less than)
  - F (found)

The F value is used to incorporate FOUND or NOT FOUND logic for segment selection. This logic allows you to include or exclude employees based on the presence of a specific segment on their record.

For the F value, the field name can be **ANY** field from the segment. To select a specific occurrence, enter the key field value(s) in the Literal Compare Value field.

- **Literal Compare Value**—used with the corresponding field name entry on this form to specify the selection criteria. The value in each field must be appropriate for the data field in the Field Name field.

**For EGLF field entries of E, G, and L**, numeric values may be entered with a decimal, commas, or dollar signs, if desired. Enter dates in the format MM-DD-YY or YYMMDD. Use century dates when a field displays on the system that way.

Slashes (/) or semi-colons (;) may be entered in the Literal Compare Value field as delimiters. A slash indicates a range and a semi-colon indicates an AND/OR condition. **When using ranges or the OR condition, the EGLF field must be an E (or =).**

A literal value of SPACES is valid here. This can be used to check a field for blanks.

**For Found/Not Found logic**, one or more key field values must be entered in the proper order, in other words, 1st key field value, 2nd key field value, and so forth. The key field values must be separated by a semi-colon (;).

Only one set of key field values is allowed per Found/Not Found entry. If the Literal Compare Value field is left blank, then the first occurrence of the segment is used.

## Employee Selection Criteria Form and the Codeset Selection Menu

Solution View Report Writer: XPROPT      Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal	Compare	Value	Action
SEX-CODE	<input type="checkbox"/>	<input type="checkbox"/>	E			
	<input type="checkbox"/>	<input type="checkbox"/>				
	<input type="checkbox"/>	<input type="checkbox"/>				
	<input type="checkbox"/>	<input type="checkbox"/>				
	<input type="checkbox"/>	<input type="checkbox"/>				
	<input type="checkbox"/>	<input type="checkbox"/>				

Display Values

EGLF Values  
 Actions

Solution View Report Writer: XPROPT      Option:

Codeset Selection Menu For Sex Codes  
For Criteria Field: SEX-CODE

Female       Male  
 Unclassified

----Complete----       Return To Selection Criteria

### NOTES

## CREATING SELECTION CRITERIA LOGIC, continued

### Using the Codeset Selection Menu

The Codeset Selection Menu can be used with the Employee Selection Criteria form when entering data into the Literal Compare Value field for fields that are tied to option lists.

If the field name entered in the Field Name field is tied to an option list and the Literal Compare Value field is left blank, the process will automatically bring the user to the Codeset Selection Menu.

Once there, the selections are made based on the option list description, for example, Male, and not the code itself, (for example, M).

### Number of Selections

You can make as many selections that will fit on one line on the Employee Selection Criteria form. If the line following the criteria entry is also blank, in other words, no field name in the Field Name entry field, then up to two lines of selections can be made.

The length of an entry is the length of the code (field length), plus one. The system will automatically place a semi-colon (;) between each selection made. If two lines of selections are made, the system will also place a C (Continue) into the first line's Action Code field.

### Exiting the Codeset Selection Menu

To leave the Codeset Selection Menu at any time, use the Leave Menu option in the Option field, or click Return to Selection Criteria.

The Codeset Selection Menu will automatically return to the Employee Selection Criteria form when the last entry is processed.

## Employee Selection Criteria Form Action Examples

**E** example: Employees who have an ACTIVITY-CODE of 001 **ARE NOT** included on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
ACTIVITY-CODE		E	001	E

**P** example: Employees who have a SEX-CODE of M (males) **ARE** included on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
SEX-CODE		E	M	P

**N and P** example: Employees who have a JOB-CODE of 12550 **AND** a JOB-CODE-EXTENT of 0002 **ARE** included on the report. Satisfying only one of these conditions does not include the employee on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550	N
JOB-CODE-EXTENT		E	0002	P

**P and P** example: Employees who have a JOB-CODE of 12550 **OR** a SALARY-GRADE of S10 **ARE** included on the report. Satisfying either one of these conditions includes the employee on the report.

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550	P
SALARY-GRADE		E	S10	P

**C and P** example: Employees who have a CONTROL-3-CODE equal to one of the values on line 1 or line 2 are included in this report.

Field Name	Not	EGLF	Literal Compare Value	Action
CONTROL-3-CODE		E	1255;1926;2183;3065;3067;3402;3555 3505;4721;6800	C P

---

### NOTES

---

**CREATING SELECTION CRITERIA LOGIC**, continued

**Field Descriptions,**  
continued

- **Action** (option list SC22)—used to determine whether to process the record being compared.

Enter **C** to indicate that the values for comparison continue on the next line; field names and the NOT and EGL indicators need not be repeated on subsequent lines which contain an action of C.

Enter **N** to indicate that other comparisons need to be checked.

Enter **P** to select the record based on this comparison.

Enter **E** to exclude a record based on this comparison.

The entries have the following effect in the program:

- **C**—the values for this statement continue on the next line
- **N**—if this statement is true, check the next comparison
- **P**—if this statement is true, include the employee in the selection process and then check the next line for another comparison
- **E**—if this statement is true, do not include the employee in the selection process

## The Employee Selection Criteria Form Conditional Statement Examples

**Include employees having a SALARY of \$20,000.00 through \$25,000.00**

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	P

**Exclude employees having a Job Code of 12550 OR 16775**

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550;16775	E

**Include only Female employees having a Job Code of 13552 through 13559**

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	13552/13559	N
SEX-CODE		E	F	P

**Exclude employees with a Salary of \$20,000.00 through \$25,000.00 OR hired before 1992**

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	E
HIRE-DATE		L	01-01-92	E

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

**Conditional Statements**

Conditional Statements are used to perform an operation dependent on some condition. The conditions that may be used in a conditional statement are:

- |                |                    |
|----------------|--------------------|
| ■ Equal to     | ■ Not Equal to     |
| ■ Greater than | ■ Not Greater than |
| ■ Less than    | ■ Not Less than    |

**Compound Conditional Statements**

Compound Conditional Statements are used to test for several conditions. Compound conditional statements include:

- **AND** Tests to see if all of several conditions exist. This means that more than one condition must be met by the employee in order to appear on the report.
  - An entry of N in the Action field designates an AND condition.
- **OR** Tests to see if one of several conditions exists. This means that at least one of the conditions must be met by the employee in order to appear on the report.
  - An entry of P in the Action field designates an OR condition when you are testing the values between one or more Field Names.
  - A semi-colon (;) inserted between a list of two or more Literal Compare Values designates an OR condition when testing the Field Names entry.
  - A slash (/) inserted between two Literal Compare Values designates an OR condition resulting in a range test for a specific Field Names entry. The range will include the starting value and the ending value.

## The Employee Selection Criteria Form

Solution View Report Writer: XPROPT      Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE	<input type="checkbox"/>	G	34020	N
SALARY-GRADE	<input type="checkbox"/>	G	24	P
	<input type="checkbox"/>			

Display Values

EGLF Values  
 Actions

Solution View Report Writer: XPROPT

Report program XPROPT has been RELOADED and may now be run using the REPORT facility.

Return to the start of the Solution View process  
 Go to Report Group Activities to schedule report

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

**Using the Employee Selection Criteria Form**

To use the Employee Selection Criteria form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Field Name</b>   | 1. Type the name of the field to be used in the conditional statement.  |
| <b>Not</b>  | 2. Indicate Not to designate the negative effect of the conditional argument entered in the EGL field (optional).   |
| <b>EGLF</b>   | 3. Type E (Equal), G (Greater), L (Less), or F (Found) to establish a conditional comparison.   |
| <b>Literal Compare Value</b>  | 4. Type the literal value to be compared with the corresponding Field Name entry.   |
| <b>Action</b>   | 5. Type C (continue values on next line), N (check next comparison), P (include in process) or E (exclude from the process) to designate the action to be taken if the conditional statement is true. |
|   | 6. Repeat steps 1 through 5 for each selection criteria condition that exists.  |
|  | 7. Click Save or press ENTER to move to the next form in the process.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the Prompt form displays. It informs you that your Report program has been written and compiled successfully. Cyborg Scripting Language Code now exists for the Report program. You may now request the report to be run using the Report Scheduling programs which are detailed in the next topic.

**Practice**

Using the worksheet, please complete this form for your program.

## **Report Group Activities**

- **Creating a Report Group**
- **Selecting Companies**
- **Entering Report Parameters**

---

NOTES

---

## REPORT GROUP ACTIVITIES

### Creating a Report Group

The Solution Series is delivered with many different report groups set up. Customized groups can be added using a combinations of delivered reports and reports created by the organization.

### Using the Report Group Activities Form

To use the Report Group Activities form:

**Selection:** \_\_\_\_\_

**Step:**

**Navigator**

1.  Reporting  
Report Scheduling
-  Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Click Add.

**Result:** The Add Report Group form is displayed.

**Report Group>**

3. Enter the name of the report group. This can be up to 6 alphanumeric characters (for example MYRPTS).

**Note:** Do not use 'PAY' as the first 3 characters of the report group.

**Title:**

4. Enter the title of the report group. This can be up to 40 alphanumeric characters.

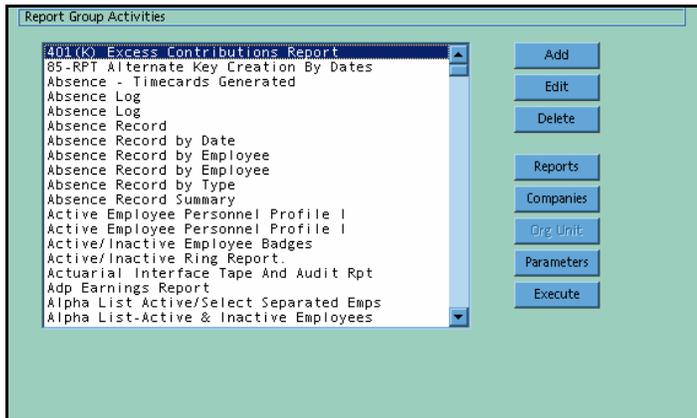
**Source Records**

5. Select the type of information to be processed from the Source Records drop-down list.

**Note:** It is required that reports that process Employee Master Records and reports that process History and Labor Records are separated into different report groups.

Position Administration reports are run separately from other types of reports because they process data based on a defined organization unit, rather than Organization. If you want to set up a report group that contains Position Administration reports, you must identify the Source Records as 'Position Administration'; otherwise, the Position Administration reports will not be an option when you move to the next form.

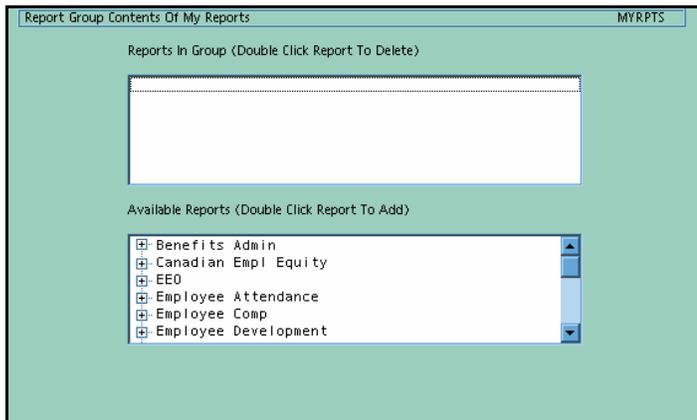
## Creating a Report Group



RGMSTR



RGADD



RPTGRP

## NOTES

REPORT GROUP ACTIVITIES, continued

Creating a Report Group, continued

Report Group View

6. Select the Report Group View. The default setting of 'Public' allows the report to be shared with others using the system. 'Private' should be selected if you do not want to share the new report group with anyone using a user code different than your own.



7. Click Save or press ENTER.

**Result:** The report group is added and the Report Group Contents form is displayed.

8. Select reports to include in the new report group by double-clicking a report title in the Available Reports box.

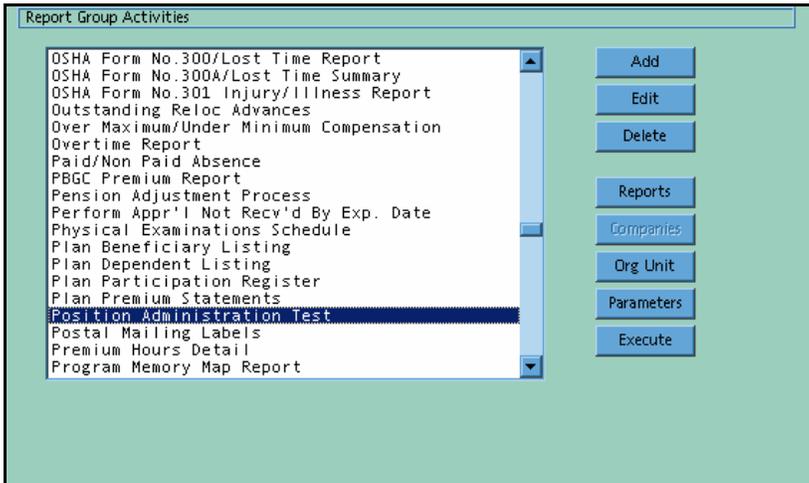
**Note:** The Available Reports option list first displays categories of reports. Double-click on a category to list the reports related to the category. Then double-click on a report to add it.



9. Click Save or press ENTER.

**Result:** The report is added and the Report Group Activities form is displayed with the new report group listed.

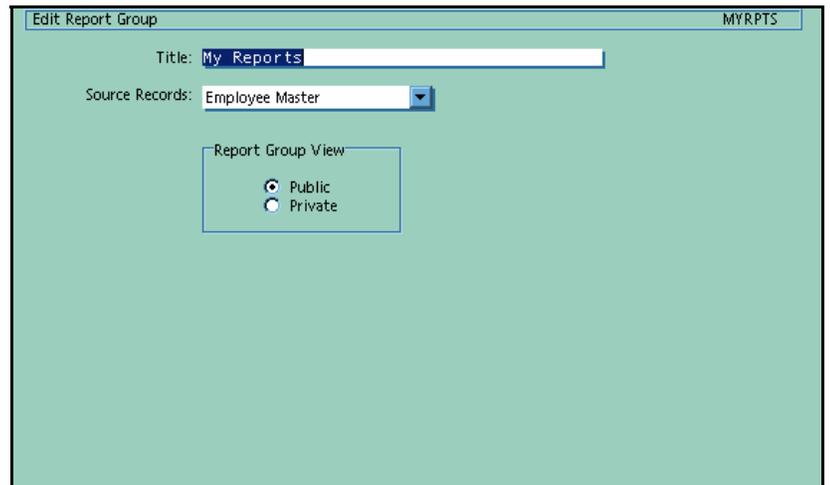
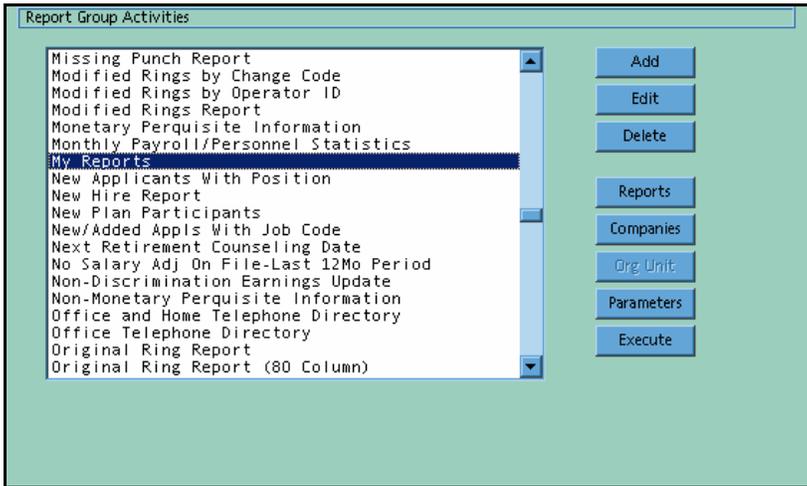
## Configuring a Position Administration Report Group



### NOTES



## Editing a Report Group



---

### NOTES

---

REPORT GROUP ACTIVITIES, continued

**Editing a Report Group**

**Note:** It is recommended that you do not edit the sample report groups delivered by Cyborg.

To edit a report group:

**Selection:**      **Step:**

**Navigator**  
Report Scheduling

1.  Reporting
-  Schedule Report Groups

**Result:** The Report Group Activities form displays.

3.      Select the report group you want to edit from the scrollable list box.
4.      Click **Edit**.

**Result:** The Edit Report Group form is displayed.

5.      Revise the settings you want to change for this report group.

**Note:** If editing a Position Administration report group, the Source Record field will display in Inquiry mode and cannot be changed.

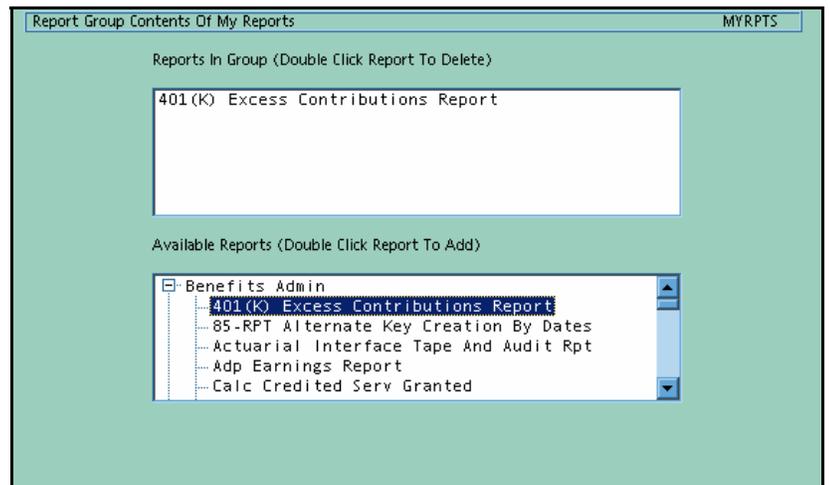
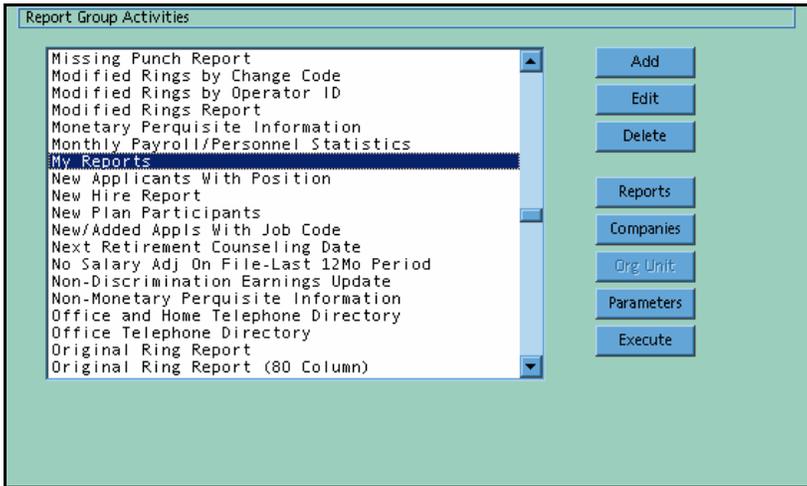
If editing a non-Position Administration report group, you may revise the Source Records setting, but 'Position Administration' is not an option.



5.      Click Save or press ENTER.

**Result:** The report group is revised and the Report Group Activities form is displayed.

## Adding a report to a Report Group



### NOTES

REPORT GROUP ACTIVITIES, continued

**Adding a report  
to a Report Group**

To add a report to a report group, perform the following steps:

**Selection:** \_\_\_\_\_ **Step:**

**Navigator**

1.  Reporting  
Report Scheduling  
 Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Select the report group to which you are adding a report from the scrollable list box.
3. Click Reports.

**Result:** The Report Group Contents form is displayed.

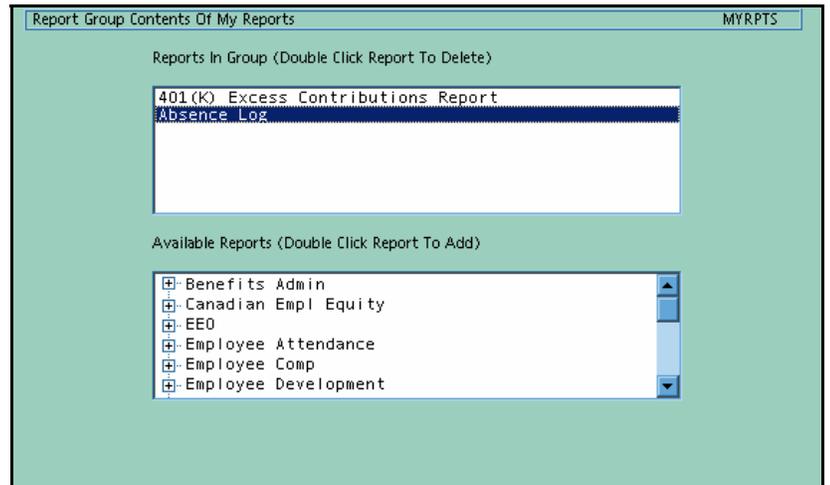
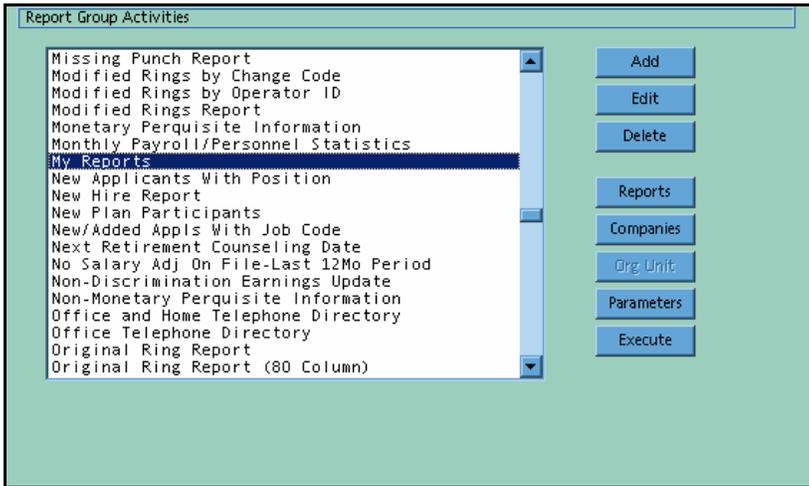
4. Select a report category and report by double-clicking on the report category displayed in the Available Reports list box. The category expands to list related topics. Double-click on a report to add it to the report group.



5. Click Save or press ENTER.

**Result:** The Report Group Activities form is displayed.

## Deleting a report from a Report Group



### NOTES

REPORT GROUP ACTIVITIES, continued

**Deleting a report  
from a Report Group**

To delete a report from a report group, perform the following steps:

**Selection:** \_\_\_\_\_ **Step:**

**Navigator**

1.  Reporting  
Report Scheduling  
 Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Select the report group from which you are deleting a report from the scrollable list box.
3. Click Reports.

**Result:** The Report Group Contents form is displayed.

4. Select a report by double-clicking on the report displayed in the Reports In Group list box.

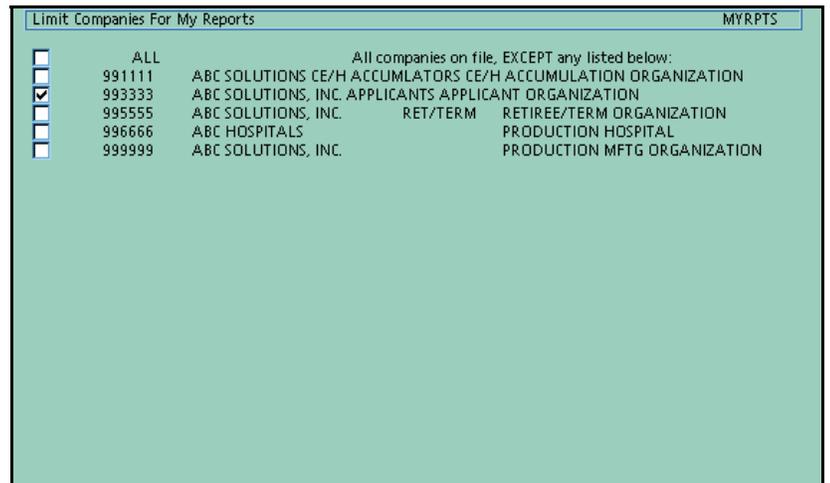
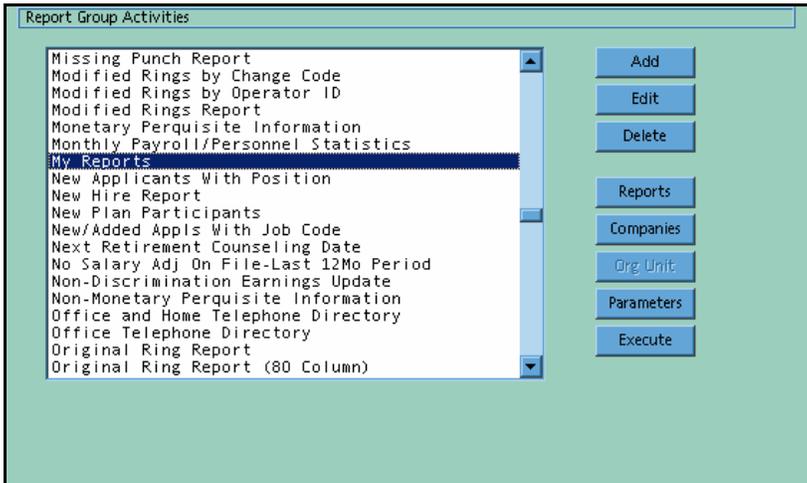


5. Click Save or press ENTER.

**Result:** The Report Group Activities form is displayed.

**Note:** Only the report is deleted from the report group, not the report source code or the report group itself.

## Limiting companies included in report processing



### NOTES

REPORT GROUP ACTIVITIES, continued

**Limiting companies included in report processing**

**Note:** Because Position Administration reports process data based on a defined organization unit, rather than Control 1-2, this configuration option is not available to report groups containing Position Administration reports.

**Selection:      Step:**

**Navigator**

1.  Reporting  
Report Scheduling  
 Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Select a report group for which you are limiting companies from the scrollable list box.
3. Click Companies.

**Result:** The Limit Companies form is displayed.

4. Select those companies you want to include in the report run by selecting the check box next to each company containing data you want processed by this report group.

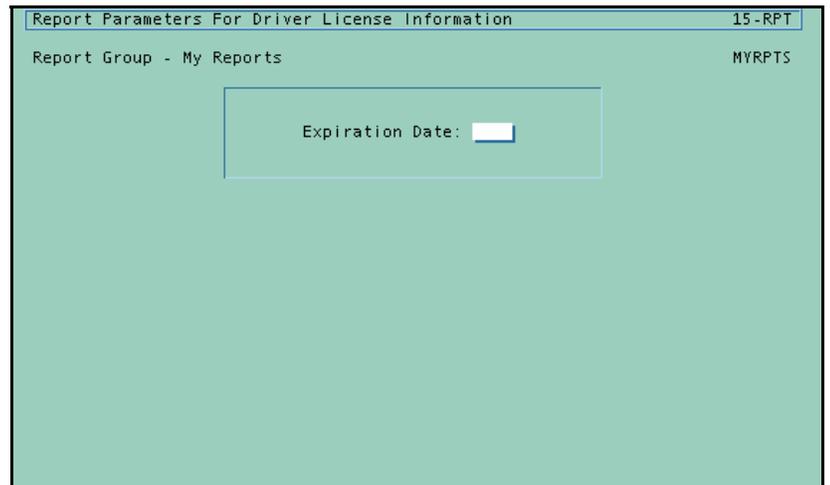
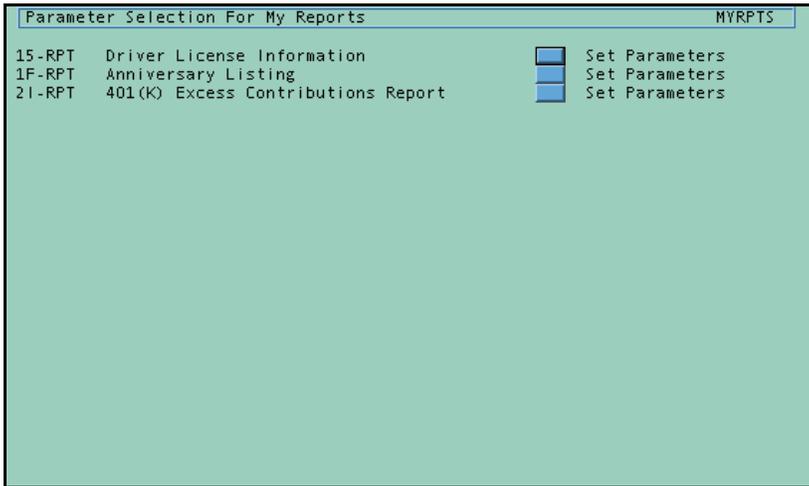
**Note:** If none are selected, all companies are reported.



5. Click Save or press ENTER.

**Result:** The reports in the report group will run only for those companies selected. The Report Group Activities form is displayed.

## Entering report parameters



---

### NOTES

---

## REPORT GROUP ACTIVITIES, continued

## Entering report parameters

This section describes how to enter report parameters.

**Selection:**

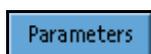
## Navigator

**Step:**

1.  Reporting  
Report Scheduling  
 Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Select a report group from the scrollable list box.



3. Click Parameters.

**Result:** The Parameter Selection form is displayed, listing all the reports included in the report group. A 'Set Parameters' button is displayed next to those reports for which you may set parameters.



4. Click the Set Parameters button next to the title of the report you want to configure.

**Result:** A Report Parameters form is displayed for the report for which you are entering parameters.

**Note:** The Report Parameters form that displays is dependent on which report's parameters you are entering.



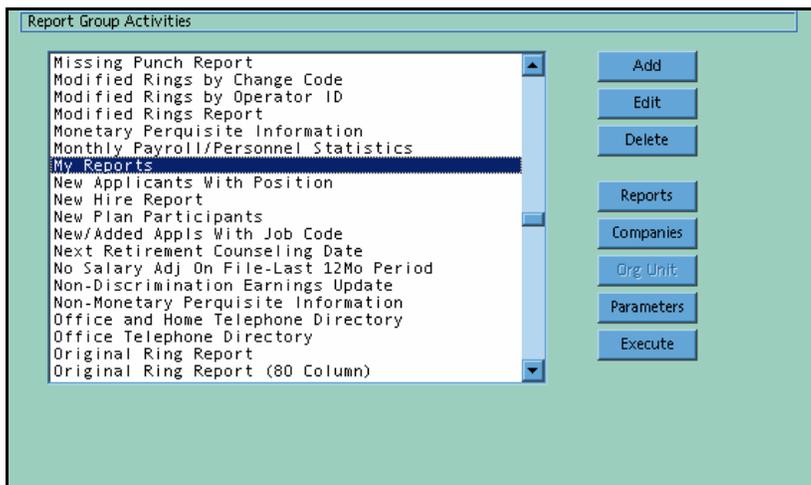
5. Complete the fields on the form, using the form-level help icon to get field-by-field entry instructions for the form.



6. Click Save or press ENTER.

**Result:** The Report Group Activities form is displayed. When a batch report job set up to run this report group is initiated online or through off-line batch operations, the reports will run as configured.

## Deleting a report group



**Warning Messages**

- +  Report group and its contents will be deleted

**Information Messages**

A Warning message means that Cyborg has found a problem with your information. Although Cyborg can proceed if this information is correct, the warning gives you a chance to review the information in case a mistake has been made.

An example of a Warning message would be entering an employee start date and birth date that make an employee fourteen years old. Although this may be true, it is most likely to be a typing error.

---

## NOTES

---

## REPORT GROUP ACTIVITIES, continued

**Deleting a report group** This section describes how to delete a report group.

**Selection:**

Navigator

**Step:**

1.  Reporting  
Report Scheduling  
 Schedule Report Groups

**Result:** The Report Group Activities form displays.

2. Select a report group from the scrollable list box.

**Note:** only the report group is deleted, not the report source code.



3. Click Delete.

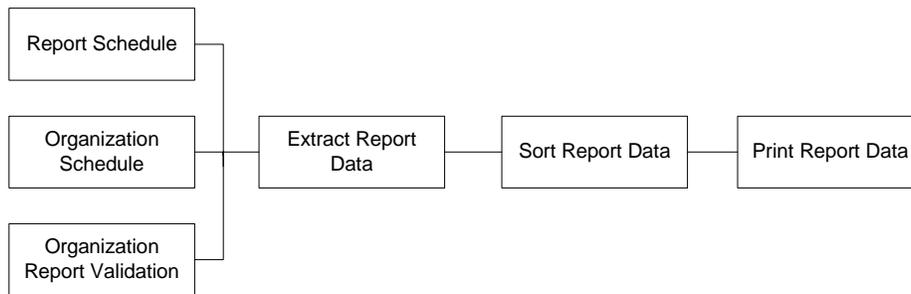
**Result:** A warning message displays in the message area.



4. Select the check box in the message area and click Save or press ENTER.

**Result:** The report group is deleted.

## Solution Series Batch Reporting



CORPORATION 99 ACME MANUFACTURING		PROJECTED 6 MOS SAL INCREASES REPT				PAGE 1	
DIVISION 9999 PRODUCTION CTL 1-2		AS OF: 07-30-92		XPRO TIME 13:39:29		DATE 07-30-92	
EMPLOYEE NUMBER	EMPLOYEE-NAME	CONTROL-3	SALARY EFFECTIVE	ANNUAL SALARY	INCREASE DATE	PROJECTED SALARY	DIFFERENCE IN-SALARY
2002	BARNES, JOHNSON	@@NONE@@	02-01-87	87,180.84	08-01-87	91,539.88	4,359.04
2007	LANNON, PATRICE	@@NONE@@	09-01-86	44,200.00	03-01-87	46,410.00	2,210.00
3001	BLOOM, ALEXANDER	@@NONE@@	04-01-84	37,999.92	10-01-84	39,899.92	1,900.00
3003	ALSON, GEOFFREY	@@NONE@@	02-19-84	31,999.92	08-19-84	33,599.92	1,600.00
3008	GRIFFITH, BERNARD	@@NONE@@	06-01-84	50,000.08	12-01-84	52,500.08	2,500.00
3009	TEACHEN, JUDITH	@@NONE@@	10-01-84	36,720.00	04-01-85	38,556.00	1,836.00
3011	LAUGHLIN, SANDRA T.	@@NONE@@	09-01-85	45,000.00	03-01-86	47,250.00	2,250.00
6001	BYERS, TODD	@@NONE@@	03-12-78	32,000.00	09-12-78	33,600.00	1,600.00
6003	BETTS, J. T.	@@NONE@@	02-19-84	32,000.00	08-19-84	33,600.00	1,600.00
6008	GRIFFITHS, ROBERT	@@NONE@@	06-01-84	50,000.00	12-01-84	52,500.00	2,500.00
6009	BEACHEM, JUDITH	@@NONE@@	09-26-83	34,000.00	03-26-84	35,700.00	1,700.00
*CONTROL-3 @@NONE@@				481,100.76		505,155.80	24,055.04
EMPLOYEE COUNT: 11							
1002	MOORE, SAMUEL	EASTERN	10-01-85	60,000.00	04-01-86	63,000.00	3,000.00
1975	GRIMES, THEODORE J.	EASTERN	06-01-85	62,999.82	12-01-85	66,149.81	3,149.99
*CONTROL-3 EASTERN				122,999.82		129,149.81	6,149.99
EMPLOYEE COUNT: 2							
1112	JOHNSON, RICH DANIEL	MIDWEST	05-01-84	68,499.96	11-01-84	71,924.96	3,425.00
1113	BARTHOLOW III, JONATHAN	MIDWEST	01-01-85	39,900.00	07-01-85	41,895.00	1,995.00
1117	ADAMS, RICHARD	MIDWEST	06-01-86	38,221.44	12-01-86	40,132.51	1,911.07
1848	COLLINS, ANNA MARIE	MIDWEST	10-01-86	39,657.60	04-01-87	41,640.48	1,982.88
*CONTROL-3 MIDWEST				186,279.00		195,592.95	9,313.95
EMPLOYEE COUNT: 4							
1007	MORITZ, KATHERINE C.	WESTERN	09-01-85	45,000.00	03-01-86	47,250.00	2,250.00
*CONTROL-3 WESTERN				45,000.00		47,250.00	2,250.00
EMPLOYEE COUNT: 1							
*CONTROL 1-2 999999				835,378.58		877,148.56	41,768.98
EMPLOYEE COUNT: 18							
*REPORT CODE XPRO				835,378.58		877,148.56	41,768.98
EMPLOYEE COUNT: 18							

### NOTES

## BATCH REPORT PROCESS

### Overview of the Batch Report Process

The last step in The Solution Series Batch Report process is the running of the report(s) in batch mode. This process requires intervention by your Data Processing department and is discussed in detail in the Solution Centre documentation for the Report Extract and Report Print programs.

For our purpose let us look at an overview of the steps required in scheduling and running the batch report(s).

Online steps include:

- **Report Group Activities**, which includes the scheduling of the report using the Report Group form, organizations, and parameters also selected here.

Batch steps include:

- **Report Extract**, which uses the Report Extract program to read the Report Scheduler form parameters and creates the report records for each report.
- **Extract Sort**, which sorts the report data into the order that the data will appear on the report.
- **Report Print**, which uses the Report Print program to format and print the report detail to hardcopy.

### Report Output Sample

The report output displayed on the facing page is the result of a report program that was created using Solution View, scheduled online and using a batch process.

## Submit and View Reports Online

- **Batch Job Initiator**
- **View Held Report**

---

NOTES

---

## SUBMITTING AND VIEWING REPORTS

### Initiating and Viewing Reports

The Solution Series allows you to initiate and preview reports or queries from an online session. The programs used in this process are:

- Initiate Scheduled Reports
- View Held Report

### Initiate Scheduled Reports

The Initiate Scheduled Reports program will:

- Provide the Query or Report fields for creating the Control Record which is required for the batch execution of reports.
- Either execute the batch job stream, or display a message telling you which Job Control Language (JCL) job stream to submit after logging off The Solution Series. The procedure you use is dependent upon your operating system. The appropriate forms are presented to you automatically.
- Optionally, route the report output for online viewing, or print the report.

### View Held Report

The View Held Report program is used to:

- Preview the output from a batch execution of a report or query started through the Initiate Scheduled Report process.
- Print the report.
- Delete the report records from the Employee Database.

## Batch Job Initiation

```
SELECT OPTION ==>   
BATCH JOB INITIATION MENU  
1 Report  
2 Query
```

```
SELECT OPTION ==> 1  
BATCH JOB INITIATION MENU  
1 Report  
2 Query
```

---

### NOTES

---

**SUBMITTING AND VIEWING REPORTS**, continued

**Accessing the Report Batch Job Initiation Menu**

To access the Report Batch Job Initiation Menu form:

<u>Selection:</u>	<u>Step:</u>
Navigator	 Reporting Report Scheduling  Initiate Scheduled Reports

**Result:** The Batch Job Initiation form displays.

**Field Descriptions**

The Report Batch Job Initiation Menu form contains the following field:

**Select Option**—allows you to select the next form for entry, either the Report Batch Job Initiator form or the Query Batch Job Initiator form. The two valid options are:

- 1—(Reports) leads you through subsequent SUBMIT forms to submit a report to the batch system.
- 2—(Query) leads you through subsequent SUBMIT forms to submit a query to the batch system.

**Using the Batch Job Initiation Menu Form**

To use the Batch Job Initiation form to initiate a report:

<u>Selection:</u>	<u>Step:</u>
Select Option	1. Type 1 (Reports).
	2. Click Save or press ENTER.

**Result:** The Report Batch Job Initiator form displays.

**Practice**

Using the worksheet, please complete these forms for your program.

## Report Batch Job Initiator Form

REPORT BATCH JOB INITIATOR

Enter Report Group name: WEEKLY

Hold output for online review? Y (Enter Y or N)

Normal, roll-up, or consolidate? (Enter space, R, or C)

---

### NOTES

---

**SUBMITTING AND VIEWING REPORTS**, continued

**Report Batch Job Initiator Form**                      The Report Batch Job Initiator form is the second form in the initiating process. The fields on this form are used to create the Batch Control Record and determine the routing of the report output.

**Field Descriptions**                                      The Report Batch Job Initiator form contains the following fields:

- **Enter Report Group name**—used to specify the name of the Report Group you wish to execute.
- **Hold Output for online review?**—used to specify whether you wish to view your report output online or whether you wish to print the report directly without online viewing. The valid options are:
  - **Y**—to allow viewing of the report output online
  - **N**—to print the report directly without online viewing
- **Normal, roll-up, or consolidate?**—Choose space for normal reporting, R for Roll-up reporting, or C for Consolidated reporting.

**Using the Report Batch Job Initiator Form**

<u>Selection:</u>	<u>Step:</u>
<b>Enter Report Group</b>	1. Type the name of the report group to be processed, (for example, WEEKLY).
<b>Hold output for online review?</b>	2. Type Y to route the report output to the Employee Database for online viewing  OR  Type N to route the report output to the Audit/Message/Report File without viewing online.
<b>Normal, roll-up, or consolidate?</b>	3. Choose space for normal batch reporting  OR  Choose R to perform roll-up reporting  OR  Choose C to perform consolidated reporting.

**Note:** Setup of Consolidated Reporting is explained in the Extra for Experts appendix.

## Report Batch Job Initiator Form

```
REPORT BATCH JOB INITIATOR
JOB JRPTXXXX  SUBMITTED
```

---

**NOTES**

---

SUBMITTING AND VIEWING REPORTS, continued

Using the Report Batch  
Job Initiator Form,  
continued

Selection:  


Step:  
4. Click Save or press ENTER.

**Result:** The Report Batch Job Initiator process will either:

- Display the message “JOB JRPTxxxx SUBMITTED”, and submit the batch job stream to execute the Report process

OR

- Display the message “LOG OFF AND SUBMIT JOB JRPTxxxx, which requires you to log off The Solution Series and submit the Job Control Language (JCL) job stream JRPTxxxx.

JRPTxxxx is a predefined job stream that will execute the REPORT process, where xxxx is the OPERATOR-ID of the user who initiated the batch job.

**Other  
Considerations**

You can use the Submit process to run a query during report processing, or start a report run during query processing. However, you can not run multiple queries or multiple Report Groups under the same operator ID simultaneously.

Submit does not interrupt your session to inform you that processing is complete. If you requested output for online viewing, you can periodically use the View Held Report program to check for held output.

To avoid problems that can result from placing too many records in the Employee Database, we recommend that authorized users limit the submit process to low volume reports and queries.

For technical considerations for the Submit program, refer to the Solution Series Technical class (2301) or the online documentation.

## Online Report Viewing

```
ACTION ==>  D-Delete, P-Print, V-View ID ==> 004  
  
ID# PGS DATE TIME  
004 0004 09-18-2002 09:28:16
```

```
ACTION ==>  V D-Delete, P-Print, V-View ID ==> 004  
  
ID# PGS DATE TIME  
004 0004 09-18-2002 09:28:16
```

---

### NOTES

---

SUBMITTING AND VIEWING REPORTS, continued

**Viewing Report Output**

Recall that the View Held Report form allows you to view, print, and/or delete the batch reports for online viewing from the Initiate Scheduled Reports process.

**Accessing the View Form**

<b>Selection:</b>	<b>Step:</b>
<b>Navigator</b>	 Reporting  Report Scheduling  View Held Report

**Result:** The View Held Report form displays.

**Note:** If the Submit process is not complete you will see the message, “YOU DO NOT HAVE ANY HELD PRINT FILES”.

**Field Descriptions**

The View form contains the following fields:

- **Action**—allows you to specify the action to be taken with each report ID. You may enter one of these three values:
  - **D**—(delete) To delete output that has been held for viewing.
  - **P**—(print) To submit held output for printing.
  - **V**—(view) To view held output.
- **ID**—used to display the three-digit ID for the most recent report (or query) that you submitted. You can also enter one of the ID’s listed in the ID field below.

There are four display fields for each print file held under your operator ID:

- **ID#**—a 3-character identifier for each report that has been routed for online viewing based on your OPERATOR-ID.
- **PGS**—the number of pages for each report.
- **Date**—the date the request was processed (in MM-DD-CCYY format).
- **Time**—the time the request was processed (in military format).

## Online Report Viewing

SIDE ==> L L-Left, R-Right PAGE ==> 0002 LINE ==> 01 Pages=0004

CORPORATION 99 ABC SOLUTIONS, INC. RET/TERM PROJECT 6 MOS SALARY INCREASES  
 DIVISION 5555 RETIREE/TERM ORGANIZATION AS OF: 09-18-2002

EMPLOYEE NUMBER	EMPLOYEE-NAME	CONTROL-3	SALARY EFFECTIVE	ANN SAL
2007	LANNON, PATRICE		09-01-1986	44,
6001	BYERS, TODD		03-12-1978	32,
6002	LITTLE, DORA		05-15-1983	24,
6006	WINTER, MICHAEL		09-25-1985	25,
6008	GRIFFITHS, ROBERT		06-01-1984	50,
6009	BEACHEM, JUDITH		09-26-1983	34,
*CONTROL-3				210,
EMPLOYEE COUNT: 6				
1114	WELKER, GEORGE W	**INVALID CODE*	07-01-1986	31,
1117	ADAMS, RICHARD	**INVALID CODE*	06-01-1986	38,
1165	MANNING, WILLIAM Z	**INVALID CODE*	10-01-1986	25,
1184	BARKER, MARTINN A	**INVALID CODE*	10-03-1983	33,

SIDE ==> R L-Left, R-Right PAGE ==> 0002 LINE ==> 01 Pages=0004

OBJECT 6 MOS SALARY INCREASES REPT XPRO PAGE 1  
 AS OF: 09-18-2002 TIME 09:38 DATE 09-18-2002

SALARY EFFECTIVE	ANNUAL SALARY	INCREASE DATE	PROJECTED SALARY	DIFFERENCE IN-SALARY
09-01-1986	44,200.00	03-01-1987	46,410.00	2,210.00
03-12-1978	32,000.00	09-12-1978	33,600.00	1,600.00
05-15-1983	24,999.96	11-15-1983	26,249.96	1,250.00
09-25-1985	25,650.00	03-25-1986	26,932.50	1,282.50
06-01-1984	50,000.08	12-01-1984	52,500.08	2,500.00
09-26-1983	34,000.08	03-26-1984	35,700.08	1,700.00
	210,850.12		221,392.62	10,542.50
CODE* 07-01-1986	31,831.80	01-01-1987	33,423.39	1,591.59
CODE* 06-01-1986	38,221.44	12-01-1986	40,132.51	1,911.07
CODE* 10-01-1986	25,099.92	04-01-1987	26,354.92	1,255.00
CODE* 10-03-1983	33,000.00	04-03-1984	34,650.00	1,650.00

### NOTES

## SUBMITTING AND VIEWING REPORTS, continued

## Viewing Held Reports

To view a report:

**Selection:**  
**Action****Step:**

1. Type V.
2. Type the three digit ID# of the report to be viewed, (for example, 001).
3. Click Save or press ENTER.

**Result:** The control record showing the router message displays. Continue to press ENTER to view the report.

## Navigating the Report Display

Three fields are available for navigating within the report:

- **SIDE**—used to view the 80 characters on the left (L) or the 80 characters on the right (R) side of the report.
- **PAGE**—used to enter a four-digit page number to scroll up or down in the report. (The total number of printed pages in the report displays at the end of the entry line.)
- **LINE**—used to enter a two-digit line number to move a specified line to the top of the form.

You may enter values in these optional entry fields, or simply press ENTER to scroll through the report. However, when you reach the end of the output file, you can not return to a previous printed page. Instead, the system returns you to the View form, where you can view the output again, delete it, or send it to a print file.

## Report Navigation Example

To view the right side of the report:

**Selection:**  
**SIDE****Step:**

1. Type R.
2. Click Save or press ENTER.

**Result:** The right side of the page displays on the form.

## Online Report Printing

```
ACTION ==> P D-Delete, P-Print, V-View ID ==> 006
ID# PGS DATE TIME
006 0004 09-18-2002 09:38:19
```

```
JOB JPRTXXX SUBMITTED
```

---

### NOTES

---

**SUBMITTING AND VIEWING REPORTS**, continued

**Printing a Report**

If you did not use the submit process to print the report, but rather to view it online, then you may print the report using the View Held Report program.

**Using View Held Report to Print a Report**

To print a query or batch report:

- | <u>Selection:</u><br>Action   | <u>Step:</u>  |
|---|---|
| <b>ID</b>   | 1. Type P.  |
|  | 2. Type the three digit ID# of the report to be printed, (for example 010). |
|   | 3. Click Save or press ENTER.   |

**Result:** The message “JOB JPRTxxxx SUBMITTED” displays and the batch job stream to execute the print process executes.

OR

The message “LOG OFF AND SUBMIT JOB JPRTxxxx” displays which requires you to log off from The Solution Series and submit the JCL job stream JPRTxxxx.

**Note:** JPRTxxxx is a predefined job stream that executes the PRINT process, where xxxx is the OPERATOR-ID for the user who initiated the batch job.

(Intervention may be required from Data Processing to establish job streams for each unique sign-on.)

## Online Report Delete

```
ACTION ==> [D] D-Delete, P-Print, V-View ID ==> 007  
ID# PGS DATE TIME  
007 0004 09-18-2002 09:43:32
```

```
---Delete completed---
```

---

### NOTES

---

SUBMITTING AND VIEWING REPORTS, continued

**Deleting a Report**

If you choose to view the report or query online, Submit writes ZR records to the Employee Database. The records are temporary, and are deleted in one of the following ways:

- when the Employee Database is rebuilt
- when you print the report
- when you delete the records by using the following delete procedure (if you do not print the report and are concerned with the size of the Employee Database)

**Using View Held Report to Delete a Report**

To delete a report:

**Selection:** \_\_\_\_\_  
**Action**

**Step:**

**ID**

1. Type D.
2. Type the three digit ID# of the report to be deleted, (for example 018).
3. Click Save or press ENTER.



**Result:** The report records are deleted from the Employee Database. The report program remains on the System Control Repository.



## INQUIRY, CHANGING, COPYING, DELETING

### Inquiry Capability

Once you have created a Report program, the capability of viewing the form series through an inquiry function is provided. While in the inquiry mode, you may change a Report program form by selecting the Set Inquiry To Entry option in the Option field of the form(s) you want to change.

### Accessing the Inquiry Feature

To access the Inquiry feature:

#### Selection:

#### Step:

#### Navigator

1.  User Tools  
User Tools
-  Solution View

**Result:** The Action form displays.

#### Name of Program

2. Type a previously-created Report program name, (for example, XPROPT).

**Note:** If you are not sure which programs are on file, select either the Program List by User, or the Program List option, to view them.

#### Program Type

3. Select Report.

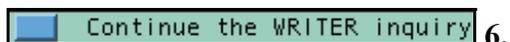
#### Action

4. Select Inquiry.



5. Click Save or press ENTER.

**Result:** An inquiry version of the first form in the series displays.



6. Click Continue the WRITER inquiry or press ENTER to move to each successive form in the series.

## Changing a Program

The screenshot shows the 'Solution View Tool Kit' window. At the top right, there is an 'Option:' dropdown menu. Below it, a horizontal box contains 'Name of Program: XPROPT' and 'Title:'. Underneath, there are two columns of radio button options. The left column is titled 'Program Type' and includes: Entry Screen, New User Fields, Query, Report (selected), PC Download, and Extract Routine. The right column is titled 'Action' and includes: Add, Change (selected), Delete, Inquiry, Program List, Program List By User, and Copy Program. At the bottom of the form, there is a 'Module:' dropdown menu and a 'Security Code:' text input field. At the very bottom of the window, a line of text reads: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Changing a Program**

Once you have created a Report program you have the option of changing it.

It is a good idea to make a copy of the original report using the Program Copier form which may be accessed from the Action form. You can then change the duplicate and have two different reports on file.

**Accessing the Change Option Form**

To access the Change Option form:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools
-  Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Report program name, (for example, XPROPT).

**Note:** If you are not sure which programs are on file, select either the Program List by User, or the Program List option, to view them.

**Title**

3. Type a new description of the Report (optional).

**Program Type**

4. Select Report.

**Action**

5. Select Change.

**Module**

6. Type a new module ID (optional).

**Security Code**

7. Type a new security code (optional).



8. Click Save or press ENTER.

**Note:** If you did not change the Title, Module, or Security fields and they were previously entered, a warning form is returned. This allows you to change the fields or leave them as they are and execute the form.

**Result:** The Report Field Selection or Change Option form displays.

## Report Change Options Form

Solution View Report Writer: XPROPT      Option: ▼

Report Change Options

- Field Selection Entries
- Selection Criteria Entries
- Calculation Factor Entries
- Key Field Values Only

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Report Change Options Form**

If Data Line entries are already on file (entered on the Report Field Selection form) and no Selection Criteria or Key field records are on file, the Report Field Selection form returns. You may now change this form and recreate the Report program the same way you created it.

However, if you have entered the Selection Criteria, Calculation Routine Entry, or Stacked Segment Key forms, the Report Change Options form displays.

Select the Field Selection Entries box to change the original field selections.

Select the Selection Criteria Entries box to change the Employee Selection Criteria.

Select the Calculation Factor Entries box to change the Calculation Factor Entries.

Select the Key Field Values Only box to change the stacked segment Key Entries.

Scroll through the forms and recreate the report. The forms you will use depend on which change(s) you indicated on the Report Change Options form.

You may also change a Report program by using the Inquiry option on the Action form. Select the Set Inquiry to Entry option in the Option field of the forms you want to change.

**Subsequent Change Forms/Sequence Numbers**

If you have indicated you want to change a Report program, you may also change the field or employee selection criteria sequence.

## Copying a Program

The screenshot shows the 'Solution View Tool Kit' interface. At the top right, there is an 'Option:' dropdown menu. Below it, a form contains the following fields and options:

- Name of Program:** XPROPT
- Title:** [Empty text box]
- Program Type:**
  - Entry Screen
  - New User Fields
  - Query
  - Report
  - PC Download
  - Extract Routine
- Action:**
  - Add
  - Change
  - Delete
  - Inquiry
  - Program List
  - Program List By User
  - Copy Program
- Module:** [Empty dropdown menu]
- Security Code:** [Empty text box]

At the bottom of the window, the text reads: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

---

### NOTES

---

INQUIRY, CHANGING, COPYING, DELETING, continued

**Copying a Program** You may copy and rename any existing Report program using the Program Copier form accessed from the Action form. You can use the copy to make modifications and have two different programs on file.

**Accessing the Program Copier Form** To access the Program Copier form:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools
-  Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Report program name, (for example, XPROPT).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Report.

**Action**

4. Select Copy.



5. Click Save or press ENTER.

**Result:** The Program Copier form displays.

## Program Copier Form

Solution View Report Writer: XPROPT      Option:

Report Program Copier

New Name:

Original Title:

Module:

Security Code:

---

### NOTES

---

INQUIRY, CHANGING, COPYING, DELETING, continued

**Field Descriptions**

The Program Copier form contains the following fields:

- **New Name**—used to enter the new report name.

All batch report program names must begin with X. The new report name must be unique, in other words, it cannot be a name that already exists.

- **Original Title**—displays the current title of the report that is being copied. Optionally, you may change this title for the copied report. Press TAB to allow the current entry to remain.

- **Module**—displays the current module ID associated with the program. Optionally, you may change this entry for the copied report. Press TAB to allow the current entry to remain.

- **Security Code**—displays the current security code associated with the program. Optionally, you may change this entry for the copied Query. The Security Code must not be higher than your assigned security level. Press TAB or press ENTER to allow the current entry to remain.

## Program Copier Form

Solution View Report Writer: XPROPT      Option: [dropdown]

Report Program Copier

New Name: XSALPT

Original Title: PROJECT 6 MOS SALARY INCREASES

Module: [dropdown]

Security Code: [input]

Solution View Report Writer: XSALPT      Option: [dropdown]

Report Change Options

- Field Selection Entries
- Selection Criteria Entries
- Calculation Factor Entries
- Key Field Values Only

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Program Copier Form**

To use the Program Copier form:

<u>Selection:</u>	<u>Step:</u>
<b>New Name</b>	<b>1.</b> Type a name for the Report copy, (for example, XSALPT).
<b>Original Title</b>	<b>2.</b> Type a new title for the copied Report. Press TAB to bypass the field (optional).
<b>Module</b>	<b>3.</b> Select a new module ID to be associated with the Report. Press TAB to bypass the field (optional).
<b>Security Code</b>	<b>4.</b> Type a new Security Code for the Report (optional).
	<b>5.</b> Click Save or press ENTER.

**Altering the Copied Program**

You are now given the ability to alter the copied program. If Data Line entries are already on file (entered on the Report Field Selection form) and no Selection Criteria or Key field records are on file, the Report Field Selection form returns. You may now change this form and recreate the Report program the same way you created it.

However, if you have entered the Employee Selection Criteria or the Stacked Segment Key form, the Change Option form displays.

Indicate the forms you want to change on the Change Options form.

Scroll through the forms and recreate the Report program. The forms you will use depend on which change(s) you indicated you wanted on the Report Change Options form.

## Deleting a Program

Solution View Tool Kit      Option:

Name of Program: XSALPT    Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input checked="" type="radio"/> Delete
<input checked="" type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Report Writer: XSALPT      Option:

Name of Program: X    Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

**\*\* REPORT program XSALPT has been deleted \*\***

---

### NOTES

---

INQUIRY, CHANGING, COPYING, DELETING, continued

**Deleting a Program** You may delete an existing Report program using the Action form.

**Accessing the Delete Feature** To delete a report program:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools
-  Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Report program name, (for example, XSALPT).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Report.

**Action**

4. Select Delete.



5. Click Save or press ENTER.

**Result:** A message displays to inform you that the program has been deleted.

## Section Summary

- **Planning Ahead: W.I.I.F.M.**
- **The Report Writer Form Series**
- **Selecting the Action**
- **Report Options Form**
- **Defining the Fields**
- **Selecting Segment Keys**

---

NOTES

---

**SECTION SUMMARY**

In this section, you learned how to write programs that run in batch mode. Specifically you learned:

- Planning Ahead: W.I.I.F.M.

---

---

---

- The Report Writer Form Series

---

---

---

- Selecting the Action

---

---

---

- Report Options Form

---

---

---

- Defining the Fields

---

---

---

- Selecting Segment Keys

---

---

---

## **Section Summary**, continued

- **Defining Calculation Routines**
- **Creating Selection Criteria Logic**
- **Report Group Activities**
- **Batch Report Process**
- **Submitting and Viewing Reports**
- **Inquiry, Changing, Copying, Deleting**

---

### NOTES

---

**SECTION SUMMARY**, continued

You also learned:

- Defining Calculation Routines

---

---

---

- Creating Selection Criteria Logic

---

---

---

- Report Group Activities

---

---

---

- Batch Report Process

---

---

---

- Submitting and Viewing Reports

---

---

---

- Inquiry, Changing, Copying, Deleting

---

---

---

## Section 5 Exercise

---

NOTES

---

**SECTION EXERCISE**

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Using the requirements worksheet and memo on the following pages, complete the Report program specified by the memo and worksheet.
2. Schedule, run, and view your completed report.

## REPORT WRITER

### MEMO

**TO:** Cyborg End User  
**FROM:** The Boss  
**SUBJECT:** Salary Projection Report

---

Administration has approved a 5% merit increase for all employees who have an above average performance rating. As you recall, employees who had an excellent rating received an 8% merit increase last month.

This increase will only impact employees whose position in range is not greater than 80%.

The date that we will give the increase to each employee will vary based on how long it has been since their last salary increase. Therefore I would like to see a time span that will reflect the current date minus the salary increase date.

Sort the report in order of smallest time span to greatest time span, and then by employee number as a second sort.

I would like to see the following listed on the report: employee name, employee number, rating value, salary effective, salary date span, position in range, annual salary, and projected annual salary.

Total the annual salaries and projected annual salaries and give an employee count.

Please see me if you have any questions.

**Section 5: Creating a Program to Generate a Batch Report**

REPORT WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X <u>S</u> <u>A</u> <u>L</u> <u>P</u> <u>T</u>		<b>Module Specific?</b> Y / N YES								
<b>Program Title:</b>		<b>Security?</b> Y / N NO								
<b>As of Date:</b>		<b>Total Employee Count?</b> Y / N								
<b>Employee Selection Criteria?</b> Y / N										
Sequence	Field Name	Segment Occurrence	Control Options	Sort Sequence	Print/Total	Calculation Type Numeric/Date/Time-Span			Result Length	Result Decimals
	EMPLOYEE-NAME					N	D	T		
	EMPLOYEE-NUMBER					N	D	T		
	RATING-VALUE					N	D	T		
	SALARY-EFFECTIVE					N	D	T		
	SALARY-DATE-SPAN					N	D	(T)	06	
	POSITION-IN-RANGE					N	D	T		
	ANNUAL-SALARY					N	D	T		
	PROJ-MERIT-INC-SAL					(N)	D	T	08	2
<b>Calculation Field Name:</b> Formula: (+ * - /)										
=										
=										
=										
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>				<b>Actions: (C, P, N, E)</b>				
<b>Field Name:</b> RATING-VALUE		<b>(not) E G L F</b>		<i>(See option list HR16)</i>		<input type="checkbox"/>				
<b>Field Name:</b> POSITION-IN-RANGE		<b>(not) E G L F</b>		80.00		<input type="checkbox"/>				
<b>Field Name:</b>		<b>(not) E G L F</b>		<input type="checkbox"/>						

\*This worksheet has been started for you. You must complete it with additional information from the letter.



## SECTION 6: CREATING A PROGRAM TO EXTRACT DATA

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW.....	239
PLANNING AHEAD: W.I.I.F.M.....	241
THE EXTRACT WRITER FORM SERIES.....	250
SELECTING THE ACTION.....	252
EXTRACT OPTIONS FORM.....	254
DEFINING THE FIELDS.....	257
SELECTING SEGMENT KEYS.....	276
DEFINING CALCULATION ROUTINES.....	282
CREATING SELECTION CRITERIA LOGIC.....	288
OUTPUT OPTIONS.....	300
INQUIRY, CHANGING, COPYING, DELETING.....	306
SECTION SUMMARY.....	320
SECTION EXERCISE.....	324

## Section Objectives

- **Determine the format and content of the extract to be produced**
- **Access and complete the form series for creating an extract routine**
- **Inquire, change, copy and delete the extract routine**

---

NOTES

---

## SECTION OVERVIEW

### **Purpose**

In this section you will learn how to create programs to generate an 80- or 150-character data file that may be used as input to another system.

### **Objectives**

When you have completed this section you will be able to:

- Determine the format and content of the extract to be produced
- Access and complete the form series for creating an extract routine
- Inquire, change, copy and delete the extract routine

## Requirements Memo

### MEMO

TO: Cyborg End-User  
FROM: The Project Leader  
DATE: August 3, 199X  
SUBJECT: Project Assistance - Extract Data

---

Based on our new project directives, I'll need some help from you in the area of extracting some data from our current Payroll/HR files for inclusion in a corporate report that will be run with their customized programs.

We'll need the following information, in this sequence, from our current files:

Employee number, social security number, employee name in full, department number, the employee's next salary review date (six months out from their last review date), and annual salary.

For this first report, please show only employees who have a salary grade of S10 or T15, and only those employees with an annual salary greater than \$9,999.99 and less than \$60,000.00.

Can you have this program ready by next Monday afternoon?

I really appreciate your help on this last minute request. If you have any questions, call me at extension 8687.

---

### NOTES

---

**PLANNING AHEAD: W.I.I.F.M.**

**What's In It For Me?**

The Extract Writer is used to generate a data file that may be used by another system. You can generate an 80- or 150-character data file. The Extract Writer works essentially the same as Report Writer. *However, there are no sort or totaling options.* Once you have written the extract program using the formatted forms provided, you can use that file in your selected in-house or external system.

This means that you can create your own extract file using the Solution View tools. You are not required to know the Cyborg Scripting Language programming language. Still, you will have all the flexibility you need to build your file and give it the characteristics your outside program requires.

**Planning the Extract**

Prior to completing the Extract Writer form series that will create the extract program, we suggest that you invest some time in reviewing the original request. This review allows you to define the requirements and the contents of your extract prior to entering the extract parameters.

The sample memo on the facing page requests an extract file to resolve some information needs. Using this example, it is clear that this manager wants information on employees who have certain salary grades and are within certain salary ranges. One field will have to be calculated to provide a salary review date six months into the future. Starting with the specifications that are stated on the memo, let us plan what will appear in the file.

**Requirements Worksheet**

You might want to use a requirements worksheet to plan your extract program. A worksheet would allow you to research the extract content and detailed specifications before you begin your entries. Once those requirements have been committed to paper, the process of using the Solution View form series becomes an easy next step.

On the next page, we will walk through a sample worksheet that you could use to specify the extract requirements.

# Requirements Worksheet

EXTRACT WRITER

Date: \_\_\_\_\_

Program Name: X _____ X T			Module Specific? Y / N					
Program Title:			Security? Y / N					
As of Date:			Extract Length? 80 / 150					
Employee Selection Criteria? Y / N								
Seq.	Field Name	Segment Occurrence	Output Option No Edit/Edit/No Output			Calculation Type Numeric/Date/TimeS pan	Result Length	Result Decimals
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
Calculation Field Name:			Formula: (+ * - /)					
			=					
			=					
			=					
If Employee Selection Criteria:		Compare Value:			Action:			
Field Name:		E G L F			<input type="checkbox"/>			
Field Name:		E G L F			<input type="checkbox"/>			
Field Name:		E G L F			<input type="checkbox"/>			

Action Options: C, P, N, E

---

## NOTES

---

**PLANNING AHEAD: W.I.I.F.M., continued**

**Using the Worksheet** Your worksheet might look something like the sample on the facing page. The worksheet contains the following specifications:

- **Program Name**—What is the name of the Extract program that will produce the extract file? As you recall, all user-defined program names are six characters and must begin with X. For extract programs, the program name must also end in XT.
- **Program Title**—What is the title of the extract? There is a 30-character limit for an extract title.
- **As Of Date**—Do you want the most current data or data as of a particular date to be used as a Report Group parameter?
- **Employee Selection Criteria?**—Will you want to specify criteria for selecting employees on this extract?
- **Module Specific?**—Do you want to specify a Module ID to be associated with this extract?
- **Security?**—Do you wish to specify the security to be associated with this extract?
- **Extract Length**—What is the extract length (limit of 80 characters for PC Solution, 150 characters otherwise).
- **Sequence**—In what sequence (00 to 99) should this field appear in the extract file?
- **Field Names**—What are the field names (20 characters) that will be used in the extract? The field names must have been previously defined, unless they are fields generated by a calculation routine in this report.

Some of the utilities available to help in your research are, Search Field Name Table For Field Name (SEARCH), Field Documentation Display (FIELDS), Field-Name Table Cross Reference Menu (F-MENU), and Form Label To Field Name Cross Reference (FLABEL).

- **Segment Occurrence**—For those fields which reside in a multiple-occurrence segment, which segment key(s) should be used for the extract? You can specify a single segment key, a range of keys,

and/or a list of keys.

# Requirements Worksheet, continued

EXTRACT WRITER

Date: \_\_\_\_\_

Program Name: X _____ X T			Module Specific? Y / N					
Program Title:			Security? Y / N					
As of Date:			Extract Length? 80 / 150					
Employee Selection Criteria? Y / N								
Seq.	Field Name	Segment Occurrence	Output Option No Edit/Edit/No Output			Calculation Type Numeric/Date/TimeS pan	Result Length	Result Decimals
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
			O	P	blank	N	D	T
Calculation Field Name:			Formula: (+ * - /)					
			=					
			=					
			=					
If Employee Selection Criteria:		Compare Value:			Action:			
Field Name:		E G L F			<input type="checkbox"/>			
Field Name:		E G L F			<input type="checkbox"/>			
Field Name:		E G L F			<input type="checkbox"/>			

Action Options: C, P, N, E

NOTES

**PLANNING AHEAD: W.I.I.F.M., continued**

**Using the Worksheet, continued** ■ **Output Option**—Should this field be:

- O Output the field with no Edit Pattern, (for example, Date 911231, Salary 001250000)
- P Output the field with Edit Pattern, (for example, Date 12-31-91, Salary 12,500.00)
- blank No output, because it is a field used only for a calculation

■ **Calculation Type**—Is this a field that should be calculated? If so, what method should be used? (Time Span Calculation is to be used when you are defining a formula of your own.) For calculation fields, fill in the next three characteristics.

■ **Result Length**—What length will the resulting calculation field be? Do not include decimals or commas in the resulting length.

■ **Result Decimals**—How many decimal places will appear in this numeric calculation result?

■ **Calculation Field Name/Formula**—Repeat each calculation field and specify the formula for calculating it.

■ **If Employee Selection Criteria**—What employee selection criteria should be used to determine which employees will appear in the report? (For example: salary ranges, date of hire, job codes.) Specify the field names and the values that should be used for comparison. Use the E (equals), G (greater than) or L (less than) indicators to make your comparison. Use F (found) to select employees based on the presence of a specific segment on their Employee Database.

You may also specify NOT prior to your indicators.

■ **Action**—You will use this field later when processing the Solution View forms.

# Requirements Worksheet, continued

EXTRACT WRITER

Date: \_\_\_\_\_

Program Name: <u>X S A L X T</u>				Module Specific?            Y / N    NO						
Program Title: <u>ANNUAL BUDGET EXTRACT</u>				Security?                        Y / N    NO						
As of Date:				Extract Length?                80 / 150    80						
Employee Selection Criteria?    Y / N    YES										
Seq.	Field Name	Segment Occurrence	Output Option No Edit/Edit/No Output			Calculation Type Numeric/Date/TimeS pan			Result Length	Result Decimals
1	EMPLOYEE-NUMBER		(O)	P	blank	N	D	T		
2	SOCIAL-SECURITY-NBR		(O)	P	blank	N	D	T		
3	EMPLOYEE-NAME		(O)	P	blank	N		T		
4	CONTROL-3-CODE	01	(O)	P	blank	N	D	T		
5	NXT-SALARY-EFFECTIVE		(O)	P	blank	N	(D)	T		
6	ANNUAL-SALARY		O	(P)	blank	N	D	T		
			O	P	blank	N	D	T		
			O	P	blank	N	D	T		
Calculation Field Name:								Formula: (+ * - /)		
NXT-SALARY-EFFECTIVE								= SALARY-EFFECTIVE + 6 MONTHS		
								=		
								=		
If Employee Selection Criteria:				Compare Value:			Action:			
Field Name: SALARY-GRADE				(E) G L F			<input type="checkbox"/> N			
Field Name: ANNUAL-SALARY				(E) G L F			<input type="checkbox"/> P			
Field Name:				E G L F			<input type="checkbox"/>			

Action Options: C, P, N, E

## NOTES

**PLANNING AHEAD: W.I.I.F.M., continued**

**Check Your Requirements**

The completed worksheet on the facing page is an example of specifications that will be used to define the extract program to be created using the Extract Writer form series. When you have completed your worksheet, you will have a list of the fields and their associated characteristics.

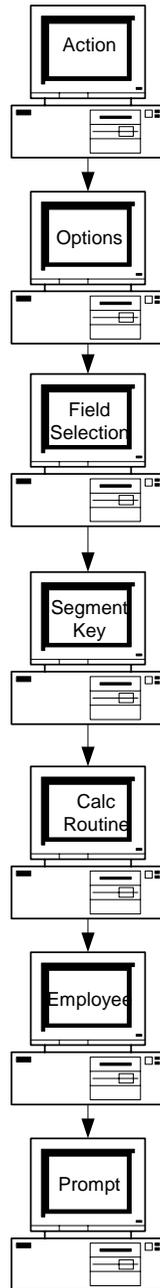
From what we have specified here, we are able to deduce the following:

- the extract will include six fields
- the extract will need a calculated next salary review date
- the listing will include only part of the employee population

**What's Next?**

Now you are ready to use the Solution View tools to enter your program requirements. We will start by previewing the form series for an Extract program.

## Extract Writer Form Series



---

### NOTES

---

## THE EXTRACT WRITER FORM SERIES

**Extract Writer** The Extract Writer is used to generate a data file that may be used by another system.

**Forms** You can generate an 80- or 150-character data file. The Extract Writer works essentially the same as Report Writer. However, there are no sort, break, or totaling options. Once you have written the extract program using the formatted forms provided, you can use that file in your chosen in-house or external system.

These are the forms, required and optional, that are available in Solution View to create your Extract program:

- **Action form**—used to name the extract program, give the extract a title, and select the type of action to be performed.
- **Options Form**—used to specify as of date options, indicate if the extract is subject to employee selections criteria, and the size of the character file.
- **Extract Field Selection form**—used to define the fields to be included in the extract.
- **Stacked Segment Key form**—(optional) used to tell the system which key you will use for selecting a specific multiple-occurrence segment where the key is not a date.
- **Calculation Routine Entry form**—(optional) defines the calculations for the fields you specified on the Extract Field Selection form.
- **Employee Selection Criteria form**—(optional) used to determine which employees will be included in the extract.
- **Prompt form**—informs you that your program has been written and compiled successfully.

**Note:** The **Copy** and **Change Option** forms also exist within the Solution View Extract facility. These are secondary forms and are not required to create an extract program. They are discussed later in this section. Now that you have previewed the form series that you will use to create the extract, let us look at each task in detail and see what tools are provided on each form and how to use them.

## Action Form

Solution View Tool Kit Option:

Name of Program:  Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input checked="" type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input checked="" type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

### NOTES

## SELECTING THE ACTION

**Action Form** The first Extract Writer form, the Action form, is used to identify the Writer Program and takes you directly to the Extract Writer form series.

### Accessing the Form

**Selection:** \_\_\_\_\_ **Step:**

#### Navigator



User Tools

User Tools



Solution View

**Result:** The Action form displays.

### Field Descriptions

The Action form field descriptions have already been presented in Section 3. In this section you will learn how to use those fields to access the Extract Writer series of forms.

### Using the Action Form

To add an Extract program:

**Selection:** \_\_\_\_\_

**Step:**

**Name of**

1. Type the name of the program, beginning with X

**Program**

and ending with XT.

**Title**

2. Type a title for the extract (optional).

**Program Type**

3. Select Extract.

**Action**

4. Select Add.

**Module**

5. Select the module ID to be associated with the extract (optional).

**Security Code**

6. Type the Security Code for the extract (optional).



7. Click Save or press ENTER.

**Result:** When the Action form is error-free, the Extract Options form displays.

### Practice

Using the worksheet, please complete this form for your program.

## Extract Options Form

Solution View Extract Writer: XSALXT      Option: [dropdown]

Extract Options

As Of Date Options

- Use Current Date
- Use Entered Date
- Use Entered Range

General Options

- Selection Criteria
- 80 Character Record
- 150 Character Record

Composite View      Single Field View

### NOTES

## EXTRACT OPTIONS FORM

### Extract Options Form

The second form in the Extract Writer form series, the Extract Options form, is used to specify as of date options, the size of the character file, and to indicate if the extract is subject to employee selection criteria.

### Field Descriptions

The Extract Options form contains the following fields:

- **As Of Date Options**—for dated segments, used to select dated segments As Of a specified date. Segment occurrences with dates equal to or less than this date are included.

The date defaults to the current date if no entry is made.

The entered date or date range will be established in the Report Group, where you will add the extract. To tie the Report Group As Of date or date range to the extract, add the extract program to your Report Group, then click the Parameters prompt button and enter the dates.

**Note:** You can use an As-of date from any form that has the date as the first key field, such as the Job Assignment/Changes form (05-SCR).

- **Selection Criteria**—used to add an Employee Selection Criteria form to the series of Extract Writer forms to indicate your answer to the question: “Do you want to narrow the list of employees based on one or more specific selection criteria?”

When a new Extract program is being added, the default for this option field indicates that employee selection criteria is desired.

If you do not want to enter selection criteria, choose the box to de-select this option.

- **Extract Length**—specifies the maximum number of characters allowed in your extract data file. Valid choices are 80 and 150.

**Note:** When using the PC Solution, the only valid choice is 80. On non-PC platforms, the default is 150.

- **Composite View or Single Field View**—specifies the next form display for Field Selection.

The Single Field View form is discussed in the Extra for Experts appendix.

**Practice**

Using the worksheet, please complete this form for your program. Click Composite View before executing the form.

## Extract Field Selection Form

Solution View Extract Field Selection: XSALXT Option:

	Seq Nbr	Field Name	Output Field	Calc Type	Result Length	Result Decimals
<input type="checkbox"/>	00	<input type="text"/>				
<input type="checkbox"/>	02	<input type="text"/>				
<input type="checkbox"/>	04	<input type="text"/>				
<input type="checkbox"/>	06	<input type="text"/>				
<input type="checkbox"/>	08	<input type="text"/>				
<input type="checkbox"/>	10	<input type="text"/>				
<input type="checkbox"/>	12	<input type="text"/>				
<input type="checkbox"/>	14	<input type="text"/>				

Display Values

Output Field     Calculation Types     Both

---

### NOTES

---

## DEFINING THE FIELDS

### Extract Field Selection Form

The third form in the Extract Writer form series, the Extract Field Selection form, is used to specify which fields are to appear on the extract, define the output options, and specify the fields that are to be used in calculations.

### Field Descriptions

The Extract Field Selection form contains the following fields:

- **Command Field Action Code**—used to add, change, or delete the Data Field Names to be printed on the extract. This field is at the far left of the form and is one character in length. It is the only field on the form that does not have a field heading above it.

When the Extract Field Selection form displays, an A (Add) is automatically placed in each Command Field so you do not have to type it in for each Field Name you are adding to your extract. Other valid entries for this field are C for Change, D for Delete, and P for Position (explained further later in this section).

- **Seq Nbr**—requires that you specify a unique sequence number for each field. The Sequence Number determines the order of the fields in the extract. Valid sequence numbers are 00–99.

When the Extract Field Selection form is displayed, a sequence number incremented by 2 is automatically placed in each Sequence Number Field so you do not have to type one in for each field name you are adding to your extract. You may wish to increment additional sequence numbers by 2's so you can insert additional fields later.

If a calculation uses the result of a prior calculation, these calculated fields must be listed on this form in logical result order. For example, you would want to sequence the amount of a salary increase prior to using it to calculate the new annual salary amount.

## Extract Field Selection Form, continued

Solution View Extract Field Selection: XSALXT Option:

	Seq Nbr	Field Name	Output Field	Calc Type	Result Length	Result Decimals
<input type="checkbox"/>	00		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	02		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	04		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	06		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	08		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	10		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	12		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	14		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Display Values

Output Field     Calculation Types     Both

---

### NOTES

---

DEFINING THE FIELDS, continued

Field Descriptions,

continued

- **Field Name**—used to specify the names of the data fields to include in the extract. The total length of all fields entered on the extract cannot exceed the extract length that is selected. A reject message informs you if you exceed the limit. The data field name entry must be valid on the Field Name Table, unless you are creating a field name to be used by, or as a product of, a calculation in the extract.

You may select either employee master fields or history and labor fields on an extract, but not both. On an employee master extract, history/labor fields may not be selected. On a history/labor extract, some employee master fields may be selected.

- **In the Extract Writer only**, a literal can also be used as field names. You can use a pair of single quotes to indicate a literal. If the literal includes a single quote or apostrophe, then you may use the word “quote” to replace the apostrophe. Example, to use the word *COMPANY’S*, your entry would be three fields:

‘COMPANY’  
QUOTE  
‘S’

**Note:** In the same vein, you may also use the following reserved fields to represent a special character to another system:

LOW-VALUE, HIGH-VALUE, and SPACE.

Your security assignment determines which fields you can access. If your program includes fields to which you are denied access, you will be unable to reload your extract.

Tools for finding

Field Names

**Field Name Table**—accessing this table in a menu-driven format can be done

by placing question marks (?) in the fields and clicking on OK or pressing ENTER.

For each question mark (?) entered you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing an X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

You can access field help while in the Field Name Table Menu by placing a question mark (?) in the entry box next to the field. To leave the menu select Leave Field Help from the Option field to return to the Field Name Table Menu.

After selecting your data field name entries from the Field Name Table Menu, select Leave Menu from the Option field to return to the Query Field Selection form.

**Note:** When you are accessing the Field Name Table you are accessing the table for The Solution Series, not the Writer tools. Every Solution Series field will be displayed, but not every field is accessible to Solution View.

## Tools for finding Field Names

Search Field Name Table for Field Name SEARCH

The SEARCH program allows you to search for a field name stored on the Field Name Table. SEARCH displays a list of names based on a partial entry in the Key fields.

Complete the text boxes below, then hit enter:

Enter the WORD/VERB/COMMAND:

**SEARCH**

**FLABEL**

Screen Label To Field Name Cross Ref. FLABEL

This program will list the Field Name and associated label of a screen.

Complete the text boxes below, then hit enter:

Enter Screen Name:

---

**NOTES**

---

**DEFINING THE FIELDS**, continued

**Tools for finding**

**Field Names,**

continued

**Search Field Name Table (SEARCH)**—The program SEARCH can also

be used without accessing the Field Name Table in the menu driven format.

This program can be accessed by selecting:

**Selection:**      **Step:**

**Navigator**



Development Tools  
Fields and Verbs



Search by Partial Name

The system will prompt for the word or field name that you wish to find. Once the form is executed, a list of field names will be returned based on your entry.

**Form Label to Field Name Cross-Reference (FLABEL)**—The program FLABEL is a good tool if you know which form you want to pull information from. The FLABEL program will list the form label as it appears on the form and then the corresponding field name to the right. To access this utility select:

**Navigator**



Development Tools  
Fields and Verbs



Display Field Names for a form

The system will prompt for a form name (for example, 40-SCR). Once the form is executed, the system will return the cross-reference list.

Other utilities include Field Documentation Display (FIELDS) and Field Name Table Cross-Reference Menu (F-MENU).

## Tools for finding Field Names, continued

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Tools for finding**  
on the  
**Field Names**,  
continued

To get the definition of a field the user should select the relevant field on the form and click What's This.

**Selection:**



**Step:**

1. With relevant field on the form selected, click What's This.

**Result:** The pointer will change to an arrow with a question mark beside it.

2. Click the field again.

**Result:** A Windows Help form appears with the field details.

## Extract Field Selection Form

Output Field Options

**O** - Output with no Edit Pattern, for example, Date 981231, Salary 001250000

**P** - Output with Edit Pattern, for example, Date 12-31-1998, Salary 12,500.00

**Blank** - Creates no output; only for fields used in a calculation

---

### NOTES

---

## DEFINING THE FIELDS, continued

### Field Descriptions, continued

Only eight data fields can be entered and viewed on the Report Field Selection form at one time. To add more fields after the first eight have been keyed and accepted, you have these options:

- Type P in the Command field with a sequence number greater than any on the form to return a clear form to allow entry of additional data fields.
- Type P in the Command field of the last data field to position that data field at the top of the Report Field Selection form.
- Type A in the Command field and a unique sequence number over an existing sequence number.

After you have entered one or more forms of field names, you may page backward or forward through them using these options:



- To return to the first form of entries click the Top of selection list icon.



- Use the Move up, or Move down icons to view the previous, or next eight fields on the report.

**Output Field** (option list SC24)—defines output options. You can display the options by choosing the Output Field button or type a ? in the field. You have three options:

- **O** to Output the field unedited
- **P** to Output the field with Edit Pattern
- **Blank** creates no output; only valid for fields used in a calculation (fields with a Calc Type entry of N, D, or T)

## Extract Field Selection Form, continued

Calculation Types
N - Numeric      D - Date      T - Time Span

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Field Descriptions,**

continued

- **Calc Type** (option list SC18)—allows you to enter your own unique data field name in the Field Name and specify the type of calculation in this field.

You can display the options by choosing the Calculation Types button, *or* by typing ? in the field.

**Note:** A Calculation Routine Entry form is returned for each calculated data field defined on this form. Use this form to define how to calculate the field. You must also enter the Result Length and Result Decimals fields on this form, discussed next.

## Extract Field Selection Form, continued

### Fields in YYMMDD Format

AGE =	Current Date – Employee Birth Date
LENGTH-OF-SERVICE = (or if Terminated) =	Current Date – Employment Date Date of Termination – Employment Date
ADJUSTED SERVICE = (or if Terminated) =	Current Date – Adjusted Seniority Date Activity Date – Adjusted Seniority Date
ORIGINAL-SERVICE = (or if Terminated) =	Current Date – Original Hire Date Activity Date – Original Hire Date

### Fields in YY Format

YEARS-OF-AGE =	Current Date – Employee Birth Date
SERVICE-YEARS = (or if Terminated) =	Current Date – Employment Date Date of Termination – Employment Date
ADJUSTED-SERVICE-YRS = (or if Terminated) =	Current Date – Adjusted Seniority Date Activity Date – Adjusted Seniority Date
ORIGINAL-SERVICE-YRS = (or if Terminated) =	Current Date – Original Hire Date Activity Date – Original Hire Date

---

### NOTES

---

## DEFINING THE FIELDS, continued

### Field Descriptions,

continued

There are eight fields you can enter on the Extract Field Selection form that are already programmed to automatically calculate time-spans. A time-span is the difference in time between two dates. **DO NOT define these fields as calculated fields by typing T in the Calc Type field.** Solution View knows how to calculate these fields.

Four of these predefined time-span fields are calculated as time-span fields in YYMMDD format. The fields, along with their method by which they are calculated, are displayed in the graphic on the opposite page.

The fields are:

- AGE
- LENGTH-OF-SERVICE
- ADJUSTED-SERVICE
- ORIGINAL-SERVICE

The other four predefined time-span fields are calculated using the same methods but in a two-digit year format, (for example, YY). These fields are:

- YEARS-OF-AGE
- SERVICE-YEARS
- ADJUSTED-SERVICE-YRS
- ORIGINAL-SERVICE-YRS

The above fields can also be used on the Employee Selection Criteria form to create extract selection logic.

**Note:** All of the above fields can be used as they are, with no T in the Calc Type field.

A Calculation Routine Entry form is returned for each calculated data field defined on this form. The Calculation Routine Entry form is where you define how to calculate the field.

## Extract Field Selection Form, continued

<b>Result Length Rules</b>	
Numeric Fields -	01 – 14 Default value is 09
Time Spans -	02 (yrs) 04 (yrs/mos) 06 (yrs/mos/days) Default value is 06
Date Fields -	10 (with edit pattern) 6 (without edit pattern)

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Field Descriptions**,  
continued

- **Result Length**—on fields to be calculated, used to enter the length of the calculated data field.

Do not include decimals or commas in the resulting field length.

- **Result Decimals**—allows you to define the number of decimal places to the right of the decimal point for numeric calculations.

Valid entries for this field are 0–6.  
The default value is 0.

## Extract Field Selection Form, continued

Solution View Extract Field Selection: XSALXT Option:

	Seq Nbr	Field Name	Output Field	Calc Type	Result Length	Result Decimals
A	00	EMPLOYEE-NUMBER	0			
A	02	SOCIAL-SECURITY-NBR	0			
A	04	EMPLOYEE-NAME	0			
A	06	CONTROL-3-CODE	0			
A	08	NXT-SALARY-EFFECTIVE	0	D		
A	10	ANNUAL-SALARY	P			
A	12					
A	14					

Display Values

Output Field     Calculation Types     Both

### NOTES

**DEFINING THE FIELDS, continued**

<b>Using the Extract Field Selection Form</b>	<b><u>Selection:</u> Action</b>	<b><u>Step:</u></b>
		<b>1.</b> Type A (add) to represent the action to be performed.
	<b>Seq Nbr</b>	<b>2.</b> Type a two-digit sequence number to specify the order in which the field will appear in the extract <b>OR</b> press TAB to accept the sequence number already in place.
	<b>Field Name</b>	<b>3.</b> Type the name of the first field to be processed in the extract.
	<b>Output Field</b>	<b>4.</b> Type O (output with no edit) or P (output with edit).
	<b>Calc Type</b>	<b>5.</b> Type N (numeric calc), D (date calc), or T (time span calc) if this field is to be created as the result of a calculation.
	<b>Result Length</b>	<b>6.</b> Enter the length of the calculated data field, without commas or decimals.
	<b>Result Decimals</b>	<b>7.</b> Enter the number of decimal places to the right of the decimal point for numeric calculations.
		<b>8.</b> Repeat steps 1 through 7 until all fields have been listed.
		<b>9.</b> Click Go To Next Step to continue to the next form.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action Form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.



## SELECTING SEGMENT KEYS

### Stacked Segment Key Form

The Stacked Segment Key form is only displayed when a field name entered on the Extract Field Selection form is in a multiple occurrence (stacked) segment whose primary Key field is not a date. This is the only time this form is used. The form is displayed automatically for each segment (not field) you listed on the Extract Field Selection form; you do not need to request it.

As with Query and Report programs, Solution View assumes you want to display the requested field's data from the first occurrence in the stacked segment. If another occurrence is desired, the criteria must be entered on this form. You may bypass this form by pressing ENTER, or choosing the Bypass this form button. (This action assumes the default, the first occurrence.)

The purpose of this form is to include/exclude segment **occurrences** in the program. If the employee does not have the segment of requested key data, this alone does not exclude the **employee** from the extract.

### Field Descriptions

The Stacked Segment Key form contains the following fields:

- **Requested Detail Line Field**—an inquiry-only field that displays a field name from the Extract Field Selection form if that field occurs in a multiple-occurrence segment whose key is not a date.

**Note:** Only the first requested field of a multiple occurrence segment is displayed for the selection of the segment key(s). All other fields from the same segment use the same comparison logic.

- **Required Keys**—displays the key field name related to the fields selected on the Extract Field Selection form. The key field name(s) used as the basis for selecting occurrences is placed here automatically by Solution View. **Do not change this field.**
- **NOT**—used to indicate that the selection criteria checks for a condition that does NOT exist. Check the box, or enter N in this field, to create selection criteria that checks for a condition that does NOT exist. The default entry of blank is interpreted in the positive, in other words, *is*.

## The Stacked Segment Key Form, continued

### Literal Compare Examples

- **To specify HED 001 (Use data only from HED 001)**

Required Keys Are Not EGL Literal Compare Values  
HED-NUMBER E 001

- **To specify HEDs from 001 to 500 (Use data from HED 001 through 500)**

Required Keys Are Not EGL Literal Compare Values  
HED-NUMBER E 001/500

- **To specify HED 500 or HED 510 (Use data from HED 500 and 510 only)**

Required Keys Are Not EGL Literal Compare Values  
HED-NUMBER E 500;510

- **To specify HEDs from 001 to 500, or HED 510 (Use data from HED 100 through 500 and also use data from HED 510)**

Required Keys Are Not EGL Literal Compare Values  
HED-NUMBER E 001/500;510

---

### NOTES

---

SELECTING SEGMENT KEYS, continued

Field Descriptions,

continued

- **EGL** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Values). Valid entries are:

- E or = (equal to)
- G or > (greater than)
- L or < (less than)

- **Literal Compare Value**—used to specify a literal value to be compared to the Required Keys Are field.

Ranges can be entered using slashes (/) to indicate an inclusive range, or semi-colons (;) to indicate an OR single or range condition. **The EGL field must be an E (=) to use ranges or OR conditions.**

**EXTRACT PROGRAMS DISPLAY MULTIPLE LINES PER EMPLOYEE IF MORE THAN ONE MATCH IS FOUND.**

**Note:** A key field may not contain spaces. Therefore, a literal entry of SPACES in the Literal Compare Values field is invalid.

- **Action** (option list SC22)—used to specify whether to include/exclude the segment occurrence being compared.

N in the field causes a segment that meets the condition to be retained if it also meets the next statement(s).

P in the field causes a segment that meets the condition to be retained.

## The Stacked Segment Key Form, continued

Solution View Extract Key Values: XSALXT    Option:   

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3-CODE

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

### Selecting the First Occurrence

### Selecting a Specific Occurrence

Solution View Extract Key Values: XSALXT    Option:   

The following REQUESTED FIELD is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence of the segment is desired enter the appropriate key information below.

Bypass this screen

Requested Detail Line Field: CONTROL-3-CODE

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	E	99	P
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

---

## NOTES

---

**SELECTING SEGMENT KEYS**, continued

**Using the Stacked Segment Key Form**      The procedure for using the Stacked Segment Key form to select a segment occurrence is:

To select the **first occurrence** of the stacked segment, click Bypass this form or press ENTER.

To select **any other occurrence**, perform the following steps:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Required Keys</b>  | 1. Do not change this field; it is automatically updated by the Extract Writer.   |
| <b>Not</b>  | 2. Optionally, indicate N to designate the negative effect of the entry in the EGL field.                                 |
| <b>EGL</b>  | 3. Type E (Equal), G (Greater), L (Less), or the symbolic equivalents, =, >, or <, to establish a conditional comparison. |
| <b>Literal Compare Value</b>  | 4. Type the literal value to be compared with the corresponding Required Keys Are field.                                  |
| <b>Action</b>   | 5. Type P (include in process) or N (check next comparison) to designate the selection of the specific occurrence.        |
|  | 6. Click Save or press ENTER.   |
|   | 7. Repeat steps 1 through 6 for each segment key requested.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and press ENTER to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**      Using the worksheet, please complete this form for your program.

## Calculation Routine Entry Form

Solution View Extract Writer: XSALXT      Option:

Calculation Routine Entry

Result Field		Factor/Field Name	
NXT-SALARY-EFFECTIVE	=	<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>
		<input type="text"/>	<input type="text"/>

Result: Date      Resulting Date: YYMMDD

---

### NOTES

---

## DEFINING CALCULATION ROUTINES

**Calculation Routine** This form defines calculations within the extract. A Calculation Routine

**Entry Form** Entry form is returned for each data field name you specified a Calc Type for on the Extract Field Selection form.

**Note:** A data field defined on a previous Calculation Routine Entry form may be used in subsequent calculations.

**Field Descriptions** The Calculation Routine Entry form contains the following fields:

- **Result, Result Field, Resulting format**—are all returned in Inquiry mode and are based on the fields you entered on the Report Field Selection form. The type of result and resulting format display in a box below the Factor/Field Names. The Result Field displays on the left side of the form.
- **Factor/Field Name**—used to define a formula to calculate by typing in factors and field names. Use this field to enter the field names or literal needed to perform the calculation.

You may select the field names to use in the formula by typing in the field name(s), or accessing the Field Name Table Menu.

- **Operation** (option list SC20)—defines the mathematical operations used in the calculation. This field is located to the right of the Factor/Field Name field and is not labeled. The valid entries are:
  - + (Add)
  - - (Subtract)
  - \* (Multiply)
  - / (Divide)

**Note:** The = operation located to the right of the Result Field is always present on this form; it is returned in Inquiry mode and may not be changed or removed. Also, the = operation may not be used in any other Operation field on this form.

## Calculation Routine Entry Form, continued

Calculation Operations	
Factor/Field Name	Result Field
-----	-----
7	CALC-RESULT +
5	*
3	
CALC-RESULT =	_____

---

### NOTES

---



## Calculation Routine Entry Form, continued

Solution View Extract Writer: XSALXT      Option:

Result Field	Factor/Field Name		
NXT-SALARY-EFFECTIVE	= SALARY-EFFECTIVE	Add	+
	00-06-00		

---

### NOTES

---

**DEFINING CALCULATION ROUTINES, continued**

**Using the Calculation Routine Entry Form**

To use the Calculation Routine Entry form:

<u>Selection:</u>	<u>Step:</u>
<b>Result Field</b>	<b>1.</b> Do not enter. Display-only field.
<b>Factor/Field Name</b>	<b>2.</b> Type the first factor or field name in the calculation.
<b>Operation</b>	<b>3.</b> Select the Calculation Option to be performed between the previous Factor/Field Name and the next Factor/Field Name.
<b>Factor/Field Name</b>	<b>4.</b> Type the next factor or field name calculation.  <b>Note:</b> Calculation fields previously defined in this report may be used in the current calculation.
	<b>5.</b> Repeat steps 3 and 4 for each arithmetic operation to be performed.
	<b>6.</b> Click Save or press ENTER.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and execute the form to return to the Action Form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

## Employee Selection Criteria Form

Solution View Extract Writer: XSALXT      Option: 

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		
	<input type="checkbox"/>	<input type="checkbox"/>		

Display Values

- EGLF Values
- Actions

### NOTES

## CREATING SELECTION CRITERIA LOGIC

### Employee Selection

The Employee Selection Criteria form is used to narrow the number of

### Criteria Form

employees that will be selected for the extract. For example, you may want to display only those employees with certain salary ranges, hire dates, and so forth, on the form. This is done by setting a condition on one or more fields. The field(s) does not have to be one of the fields that appears on the Extract Field Selection form.

Fields that you created for use in calculations may **NOT** be used on this form, or an error will result.

The use of the Employee Selection Criteria form is optional. This form only appears if it is chosen from the Extract Options form.

**Note: Criteria used to include/exclude employees on this form may not be inconsistent with segment key selections that were done on the Stacked Segment Key form.** For example, if you selected HED occurrences **not** equal to 600, then 600 should not be included here for employee selection.

### Field Descriptions

The Employee Selection Criteria form contains the following fields:

- **Field Name**—data field name(s) used as the basis for employee selection. The fields on this form do not have to be displayed on the report. Your security assignment determines which fields you can access.

To access the Field Name Table in a menu-driven format, select Field Selection Menu (M) from the Option Field *or* type a ? in the Field Name field to access the Field Name Table Menu to select the data field name(s). For each ? you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

- **Not**—to create selection criteria that checks for a negative condition, in other words, a condition that is not true.

For example, if you want only employees who do **NOT** have an annual salary greater than \$28,000 to appear on the report, you would indicate “Not” in this field (and the appropriate EGL, Literal Compare Values, and Action fields, discussed next).

## The Employee Selection Criteria Form Examples of Ranges, OR Conditions, and Found/Not Found Logic

### ■ Performance ratings of 1 or 2 or 3

Field Name      Not EGLF   Literal Compare Value  
RATING-VALUE      E   1;2;3

### ■ Birth dates ranging from 01-01-42 through 01-01-62

Field Name      Not EGLF   Literal Compare Value  
BIRTH-DATE      E   01-01-42/01-01-62

### ■ Only Pay Frequencies 1 through 3, or equal to 6

Field Name      Not EGLF   Literal Compare Value  
PAY-FREQUENCY      E   1/3;6

### ■ Salary Segment for January 1, 1993, Reclassification Increase

Field Name      Not EGLF   Literal Compare Value  
ANNUAL-AMOUNT-CHANGE      F   930101;9;I40

### ■ A value of spaces is valid. Example:

Field Name      Not EGLF   Literal Compare Value  
SHIFT-CODE      E   SPACES

---

## NOTES

---

## CREATING SELECTION CRITERIA LOGIC, continued

### Field Descriptions,

continued

- **EGLF** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are:

- E or = (equal)
- G or > (greater than)
- L or < (less than)
- F (found)

The F value is used to incorporate FOUND or NOT FOUND logic for segment selection. This logic allows you to include or exclude employees based on the presence of a specific segment on their record.

For the F value, the field name can be **ANY** field from the segment. To select a specific occurrence, enter the key field value(s) in the Literal Compare Value field.

- **Literal Compare Value**—used with the corresponding field name entry on this form to specify the selection criteria. The value in each field must be appropriate for the data field in the Field Name field.

**For EGLF field entries of E, G, and L**, numeric values may be entered with a decimal, commas, or dollar signs, if desired. Enter dates in the format MM-DD-YY or YYMMDD. Use century dates when a field displays on the system that way.

Slashes (/) or semi-colons (;) may be entered in the Literal Compare Value field as delimiters. A slash indicates a range and a semi-colon indicates an AND/OR condition. **When using ranges or the OR condition, the EGLF field must be an E (or =).**

A literal value of SPACES is valid here. This can be used to check a field for blanks.

**For Found/Not Found logic**, one or more key field values must be entered in the proper order, in other words, 1st key field value, 2nd key field value, and so forth. The key field values must be separated by a semi-colon (;).

Only one set of key field values is allowed per Found/Not Found entry. If the Literal Compare Value field is left blank, then the first occurrence of the segment is used.

## Employee Selection Criteria Form and the Codeset Selection Menu

Solution View Extract Writer: XSALXT      Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
SEX-CODE	<input type="checkbox"/>	E		
	<input type="checkbox"/>			

Display Values

EGLF Values  
 Actions

Solution View Extract Writer: XSALXT      Option:

Codeset Selection Menu For Sex Codes  
For Criteria Field: SEX-CODE

Female                       Male  
 Unclassified

----Complete----       Return To Selection Criteria

### NOTES

**CREATING SELECTION CRITERIA LOGIC, continued**

<b>Using the Codeset Selection Menu</b>	<p>The Codeset Selection Menu can be used with the Employee Selection Criteria form when entering data into the Literal Compare Value field for fields that are tied to option lists.</p> <p>If the field name entered in the Field Name field is tied to an option list, and the Literal Compare Value field is left blank, the process will automatically bring the user to the Codeset Selection Menu.</p> <p>Once there, the selections are made based on the option list description, for example, Male, and not the code itself, for example, M.</p>
<b>Number of Selections</b>	<p>You can make as many selections that will fit on one line on the Employee Selections Criteria form. If the line following the criteria entry is also blank, in other words, no field name in the Field Name entry field, then up to two lines of selections can be made.</p> <p>The length of an entry is the length of the code (field length), plus 1. The system will automatically place a semi-colon (;) between each selection made. If two lines of selections are made, the system will also place a C (Continue) into the first line's Action Code field.</p>
<b>Exiting the Codeset Selection Menu</b>	<p>To leave the Codeset Selection Menu at any time, use the Leave Menu option in the Option field or click Return to Selection Criteria.</p> <p>The Codeset Selection Menu will automatically return to the Employee Selection Criteria form when the last entry is processed.</p>

## Employee Selection Criteria Form Action Examples

**E** example: Employees who have an ACTIVITY-CODE of 001 **ARE NOT** included on the report.

Field Name	Not EGLF	Literal Compare Value	
Action			
ACTIVITY-CODE	E	001	E

**P** example: Employees who have a SEX-CODE of M (males) **ARE** included on the report.

Field Name	Not EGLF	Literal Compare Value	
Action			
SEX-CODE	E	M	P

**N and P** example: Employees who have a JOB-CODE of 12550 **AND** a JOB-CODE-EXTENT of 0002 **ARE** included on the report. Satisfying only one of these conditions does not include the employee on the report.

Field Name	Not EGLF	Literal Compare Value	
Action			
JOB-CODE	E	12550	N
JOB-CODE-EXTENT	E	0002	P

**P and P** example: Employees who have a JOB-CODE of 12550 **OR** a SALARY-GRADE of S10 **ARE** included on the report. Satisfying either one of these conditions includes the employee on the report.

Field Name	Not EGLF	Literal Compare Value	
Action			
JOB-CODE	E	12550	P
SALARY-GRADE	E	S10	P

**C and P** example: Employees who have a CONTROL-3-CODE equal to one of the values on line 1 or line 2 are included in this report.

Field Name	Not EGLF	Literal Compare Value	
Action			
CONTROL-3-CODE	E	1255;1926;2183;3065;3067;3402;3555	
C		3505;4721;6800	
P			

---

### NOTES

---

## CREATING SELECTION CRITERIA LOGIC, continued

### Field Descriptions,

continue

- **Action** (option list SC22)—used to determine whether to process the record being compared. Choose the Actions button to display the options.

Enter **C** to indicate that the values for comparison continue on the next line; field names and the NOT and EGLF indicators need not be repeated on subsequent lines which contain an action of C.

Enter **N** to indicate that other comparisons need to be checked.

Enter **P** to select the record based on this comparison.

Enter **E** to exclude a record based on this comparison.

The entries have the following effect in the program:

- **C** the values for this statement continue on the next line.
- **N** if this statement is true, check the next comparison.
- **P** if this statement is true, include the employee in the Selection process and then check the next line for another comparison.
- **E** if this statement is true, do not include the employee in the selection process.

## The Employee Selection Criteria Form Conditional Statement Examples

**Include employees having a SALARY of \$20,000.00 through \$25,000.00**

Field Name	Not EGLF	Literal Compare Value	Action
ANNUAL-SALARY	E	20000.00/25000.00	P

**Exclude employees having a Job Code of 12550 OR 16775**

Field Name	Not EGLF	Literal Compare Value	Action
JOB-CODE	E	12550;16775	E

**Include only Female employees having a Job Code of 13552 through 13559**

Field Name	Not EGLF	Literal Compare Value	Action
JOB-CODE	E	13552/13559	N
SEX-CODE	E	F	P

**Exclude employees with a Salary of \$20,000.00 through \$25,000.00 OR hired before January 1, 1992.**

Field Name	Not EGLF	Literal Compare Value	Action
ANNUAL-SALARY	E	20000.00/25000.00	E
HIRE-DATE	L	01-01-92	E

---

### NOTES

---

**CREATING SELECTION CRITERIA LOGIC**, continued

**Conditional Statements** Conditional Statements are used to perform an operation dependent on some condition. The conditions that may be used in a conditional statement are:

- |   |              |   |                  |
|---|--------------|---|------------------|
| ■ | Equal to     | ■ | Not Equal to     |
| ■ | Greater than | ■ | Not Greater than |
| ■ | Less than    | ■ | Not Less than    |

**Compound Conditional Statements** Compound Conditional Statements are used to test for several conditions.

Compound conditional statements include:

- **AND** Tests to see if all of several conditions exists. This means that more than one condition must be met by the employee in order to appear on the report.
  - An entry of N in the Action field designates an AND condition.
- **OR** Tests to see if one of several conditions exist. This means that at least one of the conditions must be met by the employee in order to appear on the report.
  - An entry of P in the Action field designates an OR condition when you are testing the values between one or more Field Names.
  - A semi-colon (;) inserted between a list of two or more Literal Compare Value designates an OR condition when testing the Field Name entry.
  - A slash (/) inserted between two Literal Compare Values designates an OR condition resulting in a range test for a specific Field Name entry. The range will include the starting value and the ending value.

## The Employee Selection Criteria Form

Solution View Extract Writer: XSALXT      Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
SALARY-GRADE	<input type="checkbox"/>	E	\$10;T15	N
ANNUAL-SALARY	<input type="checkbox"/>	E	10,000.00/59999.99	P
	<input type="checkbox"/>			

Display Values

EGLF Values  
 Actions

Solution View Extract Writer: XSALXT

Extract program XSALXT has been RELOADED and may now be run using the QUERY facility.

\*Note: QUERY must be executed in a batch run or under PC Solution.

On-line executions of QUERY are not allowed.

Return to the start of the Solution View process  
 Execute Query

---

### NOTES

---

CREATING SELECTION CRITERIA LOGIC, continued

**Using the Employee Selection Criteria Form**

To use the Employee Selection Criteria form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Field Name</b>   | 1. Type the name of the field to be used in the conditional statement.  |
| <b>Not</b>  | 2. Indicate Not to designate the negative effect of the conditional argument entered in the EGLF field (optional).  |
| <b>EGLF</b>   | 3. Type E (Equal), G (Greater), L (Less), or F (Found) to establish a conditional comparison.   |
| <b>Literal</b>  | 4. Type the literal value to be compared with   |
| <b>Compare Value</b>  | the corresponding Field Name entry.   |
| <b>Action</b>   | 5. Type C (continue values on next line), N (check next comparison), P (include in process) or E (exclude from the process) to designate the action to be taken if the conditional statement is true. |
|   | 6. Repeat steps 1 through 5 for each selection criteria condition that exists.  |
|  | 7. Click Save or press ENTER to move to the next form in the process.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the Prompt form displays. It informs you that your Extract program has been written and compiled successfully. Cyborg Scripting Language Code now exists for the Extract program. The next topic discusses output options.

**Practice**

Using the worksheet, please complete this form for your program.

## Output Options

```
SELECT OPTION ==> 2
BATCH JOB INITIATION MENU
1 Report
2 Query
```

```
QUERY BATCH JOB INITIATOR
Control 1-2: 999999 Code1-2: Key: Addl key:
QUERY KEY FROM TO
Hold output for online review? (Enter Y or N)
```

---

### NOTES

---

## OUTPUT OPTIONS

### What's Next?

The Solution Series does not allow viewing of the output for an extract because it is unable to route the output to the Employee Database. However, the Initiate Schedule Reports form can be used to output the extract to other files. The Extract program can then be run in batch and will display the extract data one page at a time.

### Accessing the Form

**Selection:**      **Step:**

**Navigator**



Reporting  
Report Scheduling



Initiate Scheduled Reports

**Result:** The Batch Job Initiation Menu Form displays.

### Using Submit to Select a Query Batch Job Initiator Form

The Submit program's first form prompts you for the type of report you wish to execute, Query or Report.

**Selection:**      **Step:**

**Select Option**

1.      Type 2 to indicate you are submitting a Query.



2.      Click Save or press ENTER.

**Result:** The Query Batch Job Initiator form displays.

### Field Descriptions

The Query Batch Job Initiator Form contains the following fields:

- **Control 1-2**—The Organization from which the Query data will be extracted.
- **Code 1-2**—Not applicable for extracts.
- **Key**— Not applicable for extracts.
- **Addl Key**— Optionally, a Report Group name to hold date parameters.
- **Query**—The name of the Query (in this case, the extract name) to be executed.
- **Key**—The key type used in the Query; for example, a Key value of 00 designates Employee Number.
- **From**—The starting key value.
- **To**—The ending key value.
- **Hold output for online review? (Enter Y or N)**—Choose N to send it directly to the batch system without viewing.

## Output Options, continued

```
QUERY BATCH JOB INITIATOR
Control 1-2: 999999 Code1-2:  Key:  Addl key:
QUERY XSALXT KEY 00 FROM 0000  TO 9999
Hold output for online review? N (Enter Y or N)
```

```
QUERY BATCH JOB INITIATOR
JOB JQRYXXXX SUBMITTED
```

---

### NOTES

---

**OUTPUT OPTIONS, continued**

**Using the Query Batch Job Initiator Form** The Query Batch Job Initiator Form is the second form in the Query Submit process. The fields on this form will be used to create the batch Control Record and determine the routing of the report output.

To use the Batch Job Initiator form:

- | <u>Selection:</u>   | <u>Step:</u>   |
|---|--|
| <b>Control 1-2</b>  | 1. Type a valid Control 1-2 number, (for example, 999999).                                       |
| <b>Key</b>  | 2. Type a Report Group name to activate the As Of date or date range parameter(s) (optional).    |
| <b>Query</b>  | 3. Type the name of the Query to be run (in this case, the extract name), (for example, XSALXT). |
| <b>Key</b>  | 4. Enter the key type for employee number, (for example, 00).                                    |
| <b>From</b>   | 5. Enter the starting key value for employee number, (for example, 0000).                        |
| <b>To</b>   | 6. Enter the ending key value for employee number, (for example, 9999).                          |
| <b>Hold output for online review?</b>   | 7. Type N to send directly to the batch system without viewing.                                  |
|  | 8. Click Save or press ENTER.  |

**Result:** The Submit process will do one of two things:

- Display the message “JOB JQRYxxxx SUBMITTED”, and submit the batch job stream to execute the Query process.
- Display the message “LOG OFF AND SUBMIT JOB JQRYxxxx”, which requires you to log off from The Solution Series and submit the Job Control Language (JCL) job stream JQRYxxxx.

## Output Example

1005	888-88-8005	MORRIS, ROBERT
028502010011	,086.40	
1010	888-88-8010	MOHR, MICHAEL T.
018602030011	,089.52	
1115	239-23-4141	SWEENEY, BARBARA
018701010010	,201.88	
2005	844-84-2005	BROWN, WILLIAM R
8701150015	,694.90	

### Using Extract Output

Cyborg Extract	————>	Word Processing
Output	————>	Spreadsheet Programs
	————>	Database Programs

---

## NOTES

---

**OUTPUT OPTIONS, continued**

**What's Next?**

Once the extract program has been created, you may run the program and use the output as input to many other types of software, either for use in-house or for an external system.

For instance, it could provide data required for pension administration needed by an actuarial. It could also provide salary figures and percentages for budgeting (see our Salary Administration module for available features). As another example, extracted data could be used to feed miscellaneous data to a parent company not using Cyborg.

# Program Inquiry

Solution View Tool Kit      Option:

Name of Program: XSALXT      Title:

<p>Program Type</p> <ul style="list-style-type: none"><li><input type="radio"/> Entry Screen</li><li><input type="radio"/> New User Fields</li><li><input type="radio"/> Query</li><li><input type="radio"/> Report</li><li><input type="radio"/> PC Download</li><li><input checked="" type="radio"/> Extract Routine</li></ul>	<p>Action</p> <ul style="list-style-type: none"><li><input type="radio"/> Add</li><li><input type="radio"/> Change</li><li><input type="radio"/> Delete</li><li><input checked="" type="radio"/> Inquiry</li><li><input type="radio"/> Program List</li><li><input type="radio"/> Program List By User</li><li><input type="radio"/> Copy Program</li></ul>
--	---

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Extract Writer: XSALXT      Option:

Extract Program General Information

Program Title: ANNUAL BUDGET EXTRACT

Module:      Security Code:

Author: S.O.      Last Modified: 09-19-02      By: S.O.

## NOTES

## INQUIRY, CHANGING, COPYING, DELETING

**Inquiry Capability** Once you have created an Extract program, the capability of viewing the form series through an inquiry function is provided. While in the inquiry mode, you may change an Extract program by form by selecting the Set Inquiry To Entry option in the Option field of the form(s) you want to change.

### Accessing the Inquiry Feature

To use the Inquiry feature:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of  
Program**

2. Type a previously-created Extract program name,  
(for example, XSALXT).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Extract.

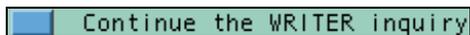
**Action**

4. Select Inquiry.



5. Click Save or press ENTER.

**Result:** An inquiry version of the first form in the series displays.



6. Click Continue the WRITER inquiry or press ENTER to move to each successive form in the series.

## Changing a Program

Solution View Tool Kit      Option:

Name of Program: XSALXT    Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input checked="" type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input checked="" type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Tool Kit      Option:

Name of Program: XSALXT    Title: ANNUAL BUDGET EXTRACT

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input checked="" type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input checked="" type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

**Warning Messages**  
 Change indicated: Title/Module/Security acceptable?

**Information Messages**

A Warning message means that Cyborg has found a problem with your information. Although Cyborg can proceed if this information is correct, the warning gives you a chance to review the information in case a mistake has been made.

An example of a Warning message would be entering an employee start date and birth date that make an

## NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Changing a Program** Once you have created an Extract program you have the option of changing the Extract program.

It is a good idea to make a copy of the original extract program using the Program Copier form which may be accessed from the Action form. You can then change the duplicate and have two different programs on file.

**Accessing the Change Option Form**

**Selection:**      **Step:**

- Navigator**
1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

- Name of Program**
2. Type a previously-created Extract program name, (for example XSALXT).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

- Title**
3. Type a new description of the Extract (optional).

- Program Type**
4. Select Extract.

- Action**
5. Select Change.

- Module**
6. Type a new module ID (optional).

- Security Code**
7. Type a new security code (optional).

-  8. Click Save or press ENTER.

**Note:** If you did not change the Title, Module, or Security fields and they were previously entered, a warning is displayed. This allows you to change the fields or leave them as they are and execute the form. Select the check box and click Save or press ENTER.

**Result:** The Extract Field Selection or Change Option form displays.

## Extract Change Options Form

Solution View Extract Writer: XSALXT      Option: [dropdown]

Extract Change Options

- Field Selection Entries
- Selection Criteria Entries
- Calculation Factor Entries
- Key Field Values Only

---

### NOTES

---

**INQUIRY, CHANGING, COPYING, DELETING, continued**

<b>Using the Extract Change Options Form</b>	<p>Select the Field Selection Entries box to change the original field selections.</p> <p>Select the Selection Criteria Entries box to change the Employee Selection Criteria.</p> <p>Select the Calculation Factor Entries box to change the Calculation Factor Entries.</p> <p>Scroll through the forms and recreate the Extract program. The forms you will use depend on which change(s) you indicated on the Extract Change Options form.</p> <p>You may also change an Extract program by using the Inquiry Option on the Action form. Select the Set Inquiry to Entry option in the Option field of the forms you want to change.</p>
<b>Subsequent Change Forms/Sequence Numbers</b>	<p>If you have indicated you want to change a Report program, you may also change the field or employee selection criteria sequence.</p>

## Copying a Program

Solution View Tool Kit Option:

Name of Program:  Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input checked="" type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input checked="" type="radio"/> Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

### NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Copying a Program** You may copy and rename any existing Extract program using the Program Copier form accessed from the Action form. You can use the copy to make modifications and have two different programs on file.

**Accessing the Program Copier Form** To access the Program Copier form:

- | <u>Selection:</u> | <u>Step:</u>   |
|-------------------|--|
| <b>Navigator</b>  | 1.  User Tools<br>User Tools<br> Solution View |

**Result:** The Action form displays.

- |                        |   |
|------------------------|---|
| <b>Name of Program</b> | 2. Type a previously-created Extract program name, (for example, XSALXT). |
|------------------------|---|

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

- |   |                               |
|---|-------------------------------|
| <b>Program Type</b>   | 3. Select Extract.            |
| <b>Action</b>   | 4. Select Copy Program.       |
|  | 5. Click Save or press ENTER. |

**Result:** The Extract Routine Program Copier form displays.

## Extract Routine Program Copier Form

Solution View Extract Writer: XSALXT      Option:

Extract Routine Program Copier

New Name:

Original Title:

Module:

Security Code:

### NOTES

**INQUIRY, CHANGING, COPYING, DELETING**, continued

**Field Descriptions**

The Extract Routine Program Copier form contains the following fields:

- **New Name**—used to enter the new extract name.

All extract program names must begin with X and end in XT. The new extract name must be unique, in other words, it cannot be a name that already exists.

- **Original Title**—displays the current title of the extract that is being copied. Optionally, you may change this title for the copied extract. Press TAB to allow the current entry to remain.

- **Module**—displays the current module ID associated with the program. Optionally, you may change this entry for the copied extract. Press TAB to allow the current entry to remain.

- **Security Code**—displays the current security code associated with the program. Optionally, you may change this entry for the copied extract. The Security Code must not be higher than your assigned security level. Press TAB or press ENTER to allow the current entry to remain.

## Program Copier Form

Solution View Extract Writer: XSALXT      Option:

Extract Routine Program Copier

New Name: XSABXT

Original Title: SEMI-ANNUAL BUDGET EXTRACT

Module:

Security Code:

### NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Program Copier Form**

To use the Program Copier form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>New Name</b>   | <b>1.</b> Type a name for the extract copy, (for example, XSABXT).  |
| <b>Original Title</b>   | <b>2.</b> Type a new title for the copied extract, for example, Semi-annual Budget Extract. Press TAB to bypass the field (optional). |
| <b>Module</b>   | <b>3.</b> Type a new module ID to be associated with the extract. Press TAB to bypass the field (optional).                           |
| <b>Security Code</b>  | <b>4.</b> Type a new Security Code for the extract (optional).  |
|  | <b>5.</b> Click Save or press ENTER.  |

**Altering the Copied Program**

You are now given the ability to alter the copied program. If Data Line entries are already on file (entered on the Extract Field Selection form) and no Selection Criteria or Key field records are on file, the Extract Field Selection form returns. You may now change this form and recreate the Extract program the same way you created it.

However, if you have entered the Selection Criteria, Calculation Routine Entry, or Stacked Segment Key forms, the Change Option form displays.

Indicate the forms you want to change on the Change Option form.

Scroll through the forms and recreate the Extract program. The forms you will use depend on which change(s) you indicated you wanted on the Extract Change Options form.

## Deleting a Program

Solution View Tool Kit      Option:

Name of Program: XSALXT    Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input checked="" type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input checked="" type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Extract Writer: XSALXT      Option:

Name of Program: X    Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:       Security Code:

\*\* EXTRACT program XSALXT has been deleted \*\*

### NOTES

**INQUIRY, CHANGING, COPYING, DELETING**, continued

**Deleting a Program**      You may delete an existing Extract program from the Action form.

**Accessing the Delete Feature**      To delete an Extract program:

- | <u>Selection:</u> | <u>Step:</u>   |
|-------------------|--|
| <b>Navigator</b>  | 1.  User Tools<br>User Tools<br> Solution View |

**Result:** The Action form displays.

- |                        |   |
|------------------------|---|
| <b>Name of Program</b> | 2. Type a previously-created Extract program name, (for example, XSALXT). |
|------------------------|---|

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

- |   |                               |
|---|-------------------------------|
| <b>Program Type</b>   | 3. Select Extract.            |
| <b>Action</b>   | 4. Select Delete.             |
|  | 5. Click Save or press ENTER. |

**Result:** A message displays to inform you that the program has been deleted.

## Section Summary

- **Planning Ahead: W.I.I.F.M.**
- **The Extract Writer Form Series**
- **Selecting the Action**
- **Extract Option Form**
- **Defining the Fields**
- **Selecting Segment Keys**

---

NOTES

---

**SECTION SUMMARY**

In this section, you learned how to create programs to generate an 80 or 150 character data file that may be used as input to another system. Specifically you learned:

- Planning Ahead: W.I.I.F.M.

---

---

---

- The Extract Writer Form Series

---

---

---

- Selecting the Action

---

---

---

- Extract Options Form

---

---

---

- Defining the Fields

---

---

---

- Selecting Segment Keys

---

---

---

**Section Summary**, continued

- **Defining Calculation Routines**
- **Creating Selection Criteria Logic**
- **Output Options**
- **Inquiry, Changing, Copying, Deleting**

---

NOTES

---

**SECTION SUMMARY**, continued

You also learned:

- Defining Calculation Routines

---

---

- Creating Selection Criteria Logic

---

---

- Output Options

---

---

- Inquiry, Changing, Copying, Deleting

---

---

## Section 6 Exercise

---

NOTES

---

**SECTION EXERCISE**

**Purpose** This exercise gives you practice using the information you have learned in this section.

Using the requirements worksheet and memo on the following pages, complete the Extract program specified by the memo and worksheet.

## EXTRACT WRITER

### MEMO

**TO:** Cyborg End User  
**FROM:** The Project-Leader  
**SUBJECT:** Extracting Budget Data

---

Management has approved the budget for next year and has asked to see the dollar effect on the budget based on projected salary increases entered on the 45-SCR form.

Since the budget is on an outside database, I will need to you to extract 45-SCR form information so that I can move it into the database to perform the calculations.

I will need the following information: employee last name, employee number, HR Control-Three value, salary increase date, plan percent change, plan annual amount change, plan budget effect-\$, and plan budget effect-%.

I realize that not every employee has a 45-SCR form segment. Therefore, anyone with blanks in the salary increase date field should be excluded.

Please feel free to call if you have any questions.



**Section 6: Creating a Program to Extract Data**

**EXTRACT WRITER**

Date: \_\_\_\_\_

<b>Program Name:</b> X _____ X T		<b>Module Specific?</b> Y / N								
<b>Program Title:</b>		<b>Security?</b> Y / N								
<b>As of Date:</b>		<b>Extract Length</b> 80 / 150								
<b>Employee Selection Criteria?</b> Y / N										
Sequence	Field Name	Segment Occurrence	Output Option			Calculation Type			Result Length	Result Decimals
			No Edit/Edit/No Output			Numeric/Date/Time-Span				
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
<b>Calculation Field Name:</b> <b>Formula: (+ * - /)</b>										
=										
=										
=										
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>				<b>Actions: (C, P, N, E)</b>				
<b>Field Name:</b>		<b>(not) E G L F</b>				<input type="checkbox"/>				
<b>Field Name:</b>		<b>(not) E G L F</b>				<input type="checkbox"/>				
<b>Field Name:</b>		<b>(not) E G L F</b>				<input type="checkbox"/>				

\* This worksheet has been left blank. You must complete it with information from the letter.

## SECTION 7: CREATING A PROGRAM TO ENTER DATA BY USING A FORM

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW .....	330
PLANNING AHEAD: W.I.I.F.M.....	332
THE FORM WRITER FORM SERIES.....	338
SELECTING THE ACTION .....	340
DEFINING THE FIELDS .....	342
CREATING HELP FORM TEXT .....	350
ACCESSING THE NEW FORM .....	352
INQUIRY, CHANGING, COPYING, DELETING.....	358
SECTION SUMMARY .....	370
SECTION EXERCISE.....	372

## Section Objectives

- **Determine the content and format of the new entry form**
- **Identify the form series that will be used to create the data entry form**
- **Define the form title, module ID, and security code**
- **Provide a form description and form prompts**
- **Define the existing fields to be used on the form**
- **Analyze the new data entry form to verify the field descriptions and the segment structure**
- **Inquire, Change, Copy, and Delete the form program**

---

NOTES

---

## SECTION OVERVIEW

**Purpose** In this section you will learn how to create programs for employee-level forms using existing segments and fields.

**Objectives** When you have completed this section you will be able to do the following:

- Determine the content and format of the new form
- Identify the form series that will be used to create the data form
- Define the form title, module ID and security code
- Provide a form description and form prompts
- Define the existing fields to be used on the form
- Analyze the new data form to verify the field descriptions and the segment structure
- Inquire, Change, Copy, and Delete the form program

## Requirements Memo

### MEMO

TO: Cyborg End-User  
FROM: The Boss  
DATE: August 4, 199X  
SUBJECT: Form Request

---

As the new manager of the Human Resources Department, I am very interested in obtaining a combination form that has fields from various other separate forms to be used by Department Managers.

Since this is just a trial run of the new form, would you create the form program for our use and let me try it for the next week or two? Then we should have enough experience with it to know whether or not we need to make any changes.

We'll want this basic employee information on a single form. Try it in this sequence:

Employee name, sex code, birth date, date of employment, department number and name, marital status, home phone and salary per period.

I'll be asking for three other forms in the near future. These other forms will be entered in sequence after this one has been keyed. Call me for those details.

I'd like to try this form as soon as you have it ready. Maybe we can load it with employee information from the test company at first. That way I can test each possible set of circumstances before we use it for real employees.

Call me if you need any further information. I'm at extension 643 in the corporate building.

---

### NOTES

---

## PLANNING AHEAD: W.I.I.F.M.

### What's In It For Me?

Form Writer is an end-user tool that enables you to create your own employee-level form using existing segments and fields. There is a twenty field limit per form. These twenty fields can be selected from ten different segments in addition to the E segment (Employee Information form). You can use delivered or user-defined fields.

This means that you can create your own combination form using the Solution View tools. You are not required to know the Cyborg Scripting Language programming language. You will be able to combine on one form the fields you use most often, or want to use to accomplish a given procedure.

### Planning the Form

Prior to completing the Form Writer form series that will create the form program, we suggest that you invest some time in reviewing the original request. This review allows you to define the requirements and the contents of your form program prior to entering the form parameters.

The graphic on the facing page is a sample memo that requests a new form to resolve some information needs. Using this example, it is clear that this manager wants to be able to combine existing fields of data on one form for more efficiency for Department Managers. Starting with the specifications that are stated on the memo, let us plan what will appear on the form.

### Requirements Worksheet

You might want to use a requirements worksheet to plan your form program. A worksheet would allow you to research the form content and detail specifications before you begin your entries. Once those requirements have been committed to paper, the process of using the Solution View form series becomes an easy next step.

On the next page, we will walk through a sample worksheet that you could use to specify the form requirements.



**PLANNING AHEAD: W.I.I.F.M.,** continued

**Using the Worksheet**

Your worksheet might look something like the sample on the facing page. The worksheet contains the following specifications:

- **Program Name**—What is the name of the program that will produce the new entry form? As you recall, all user-defined program names are six characters and must begin with X. For new form programs, you may choose to end the name with SCR to define the program as a form program.
- **Program Title**—What is the title that will appear at the top of the form? There is a 30-character limit for a form title.
- **Module Specific?**—Do you want to specify a Module ID to be associated with this form?
- **Security?**—Do you wish to specify the security to be associated with this form?
- **Sequence**—In what sequence should the fields appear on the form from left to right?
- **Field Name**—What are the field names that will be used in the form? There is a limit of twenty fields and ten keyed segments plus the E segment.
- **Prompts**—What prompt(s) are to be displayed at the bottom of the entry form? There is a limit of three lines.
- **HELP Description**—What is the description of this form to be displayed in the HELP function? There is a limit of four lines.

## Requirements Worksheet, continued

FORM WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> <u>X</u> <u>B</u> <u>D</u> <u>S</u> <u>C</u> <u>R</u>		<b>Module Specific?</b> Y / N <i>NO</i>	
<b>Program Title:</b> <i>EMPLOYEE BASIC DATA FORM</i>		<b>Security?</b> Y / N <i>NO</i>	
<b>Sequence</b>	<b>Field Name</b>	<b>Sequence</b>	<b>Field Name</b>
1	<i>EMPLOYEE-NAME</i>	2	<i>SEX-CODE</i>
3	<i>BIRTH-DATE</i>	4	<i>EMPLOYMENT-DATE</i>
5	<i>CONTROL-3-CODE</i>	6	<i>CONTROL-3</i>
7	<i>JOB-CODE</i>	8	<i>MARITAL-STATUS</i>
9	<i>EMPLOYEE-STATUS</i>	10	<i>HOME-PHONE</i>
11	<i>SALARY-PER-PERIOD</i>		
<b>Form Prompts:</b>			
<i>ENTER:            EF-SCR TO CONTINUE BASIC DATA</i>			
<i>                     55-SCR TO ENROLL INTO COMPANY PLAN</i>			
<i>                     40-SCR TO CALCULATE SALARY INFORMATION</i>			
<b>Help Description:</b>			
<i>This form is used to obtain basic employee information by managers.</i>			
<i>Added information can be found by using the XX, XY, and XZ forms.</i>			
<i>THIS IS A USER_DEFINED, NON-CYBORG FORM.</i>			

---

### NOTES

---

**PROVIDE TRANSITION** to the next topic, The Form Writer Form Series.

**PLANNING AHEAD: W.I.I.F.M., continued**

**Check Your Requirements**

The completed worksheet on the facing page is an example of specifications that will be used to define the form program to be created using the Form Writer form series. When you have completed your worksheet, you will have a list of the fields, prompt, and HELP display information.

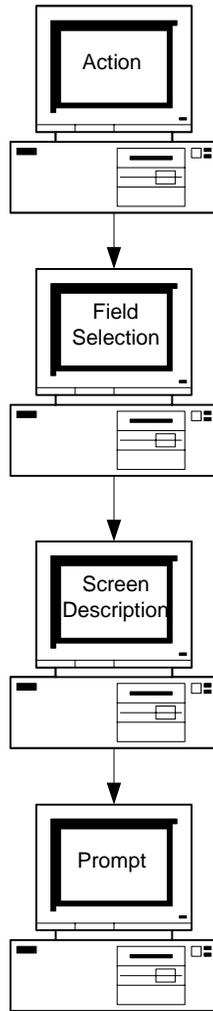
From what we have specified here, we are able to deduce:

- that the form is meant to display basic employee data
- that we have three other forms which will follow this one in sequence
- that the finished form program is still open to change

**What's Next?**

Now you are ready to use the Solution View tools to enter your program requirements. We will start by previewing the form series for a Form program.

## Form Writer Form Series



---

### NOTES

---

## THE FORM WRITER FORM SERIES

### Form Writer Forms

The Form Writer is used to create your own employee-level form using existing segments and fields. There is a twenty field limit per form. You can use ten keyed segments in addition to the E segment. You can use delivered or pre-defined user fields.

These are the forms, required and optional, that are available in Solution View to create your Form program:

- **Action form**—allows you to name the form program, give the form a title, and select the type of action to be performed.
- **Form Field Selection form**—used to specify the fields that will display on the new form and the prompts that will appear at the bottom of the new form.
- **Online Form Description Entry form**—(optional) used to specify a form description for display in the HELP function.
- **Prompt form**—tells you that the form program has been written and compiled successfully.

Now that you have previewed the form series that you will use to create the new form, let us look at each task in detail, see what tools are provided on each form, and how to use them.

## Action Form

The screenshot shows the 'Solution View Tool Kit' interface. At the top left is the title 'Solution View Tool Kit' and at the top right is an 'Option:' dropdown menu. Below this is a form with two main sections: 'Name of Program' and 'Title'. The 'Name of Program' field contains 'XBDSCR' and the 'Title' field contains 'EMPLOYEE BASIC DATA FORM'. Below these are two columns of radio button options. The left column is titled 'Program Type' and includes: 'Entry Screen' (selected), 'New User Fields', 'Query', 'Report', 'PC Download', and 'Extract Routine'. The right column is titled 'Action' and includes: 'Add' (selected), 'Change', 'Delete', 'Inquiry', 'Program List', 'Program List By User', and 'Copy Program'. Below these columns are two more fields: 'Module:' with a dropdown menu and 'Security Code:' with a small input box. At the bottom of the window, it says 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

### NOTES

## SELECTING THE ACTION

**Action Form** The first Form Writer form, the Action form, is used to identify the Writer program and takes you directly to the Form Writer form series.

### Accessing the Form

**Selection:**

**Step:**

**Navigator**



User Tools

User Tools



Solution View

**Result:** The Action form displays.

### Field Descriptions

The Action form field descriptions have already been presented in Section 3. In this section you will learn how to use those fields to access the Form Writer series of forms.

### Using the Action Form

To add a Form program:

**Selection:**

**Step:**

**Name of Program**

1. Type the one- to six-character name of the program, beginning with X, (for example, XBDSCR).

**Title**

2. Type a title for the report (optional).

**Program Type**

3. Select Entry Form.

**Action**

4. Select Add.

**Module**

5. Select the module ID to be associated with the report (optional).

**Security Code**

6. Type the Security Code for the report (optional).



7. Click Save or press ENTER.

**Result:** When the Action form is error-free, the Form Field Selection form displays.

### Practice

Using the worksheet, please complete this form for your program.

## Form Field Selection Form

Solution View Screen Field Selection: XBDSCR Option: 

Field Names: 



Screen Prompts  Use As Prompts 


Include Description

---

### NOTES

---

## DEFINING THE FIELDS

### Form Field Selection Form

The second form in the Form Writer form series, the Form Field Selection form, is used to define all the data fields you want on the entry form and define the prompts you want to see at the bottom of the form.

### Field Descriptions

The Form Field Selection form contains the following fields:

- **Field Names**—used to specify the names of the data fields to appear on the form. You can define up to twenty fields on your form from ten different segments plus the E segment. The data field name entry must be valid on the Field Name Table, either delivered or user-defined. Your security assignment determines which fields you can access.

To access the Field Name Table in a menu-driven format, select Field Selection Menu from the Option Field or type a ? in the Field Name field, to select the data field name(s). For each ? you will be allowed to select an entry from the Field Name Table Menu.

Once you are viewing the menu selections, select your field names by placing X next to each one. You may also place a 1, 2, 3, and so forth, in the entry boxes of the data field names and the data fields will appear on the report in the specified order.

You can access field help while in the Field Name Table Menu by placing a ? in the entry box next to the field. Select Leave Field Help from the Option field to return to the Field Name Table Menu.

After selecting your data field name entries from the Field Name Menu, select Leave Menu from the Option field to return to the Query Field Selection form.

- **Form Prompts**—optionally used to create prompt message text that appears on the bottom of the newly created form.

If you wish to include the message text as a navigational prompt, select the Use As Prompts check box to create an entry on the context-sensitive menu. The prompt text must be in the format “XXXSCR Form title” (where XXX are the first three characters of the form program name).

[If you have a non-windows system, you may wish to precede the form program name and title with ENTER, followed by a colon and a space.]

- **Include Description**—lets the Form Writer know that you want to enter a description of the form that will appear when you run the HELP program for the form. If you want to enter a description, check the box.

## Tools for finding Field Names

Search Field Name Table for Field Name SEARCH

The SEARCH program allows you to search for a field name stored on the Field Name Table. SEARCH displays a list of names based on a partial entry in the Key fields.

Complete the text boxes below, then hit enter:

Enter the WORD/VERB/COMMAND:

**SEARCH**

**FLABEL**

Screen Label To Field Name Cross Ref. FLABEL

This program will list the Field Name and associated label of a screen.

Complete the text boxes below, then hit enter:

Enter Screen Name:

---

**NOTES**

---

**DEFINING THE FIELDS**, continued

**Tools for finding Field Names**, continued

**Search Field Name Table (SEARCH)**—The program SEARCH can also be used without accessing the Field Name Table in the menu driven format.

This program can be accessed by selecting:

**Selection:**      **Step:**

- |                  |   |               |
|------------------|---|---------------|
| <b>Navigator</b> |  | User Tools    |
|                  |   | User Tools    |
|                  |  | Solution View |

The system will prompt for the word or field name that you wish to find. Once the form is executed, a list of field names will be returned based on your entry.

**Form Label to Field Name Cross-Reference (FLABEL)**—The program FLABEL is a good tool if you know which form you want to pull information from. The FLABEL program will list the form label as it appears on the form and then the corresponding field name to the right. To access this utility select:

- |                  |   |                                |
|------------------|---|--------------------------------|
| <b>Navigator</b> |  | Development Tools              |
|                  |   | Fields and Verbs               |
|                  |  | Display Field Names for a form |

The system will prompt for a form name (for example, 40-SCR). Once the form is executed, the system will return the cross-reference list.

Other utilities include Field Documentation Display (FIELDS) and Field Name Table Cross-Reference Menu (F-MENU).

## Tools for finding Field Names, continued

---

### NOTES

---

**DEFINING THE FIELDS**, continued

**Tools for finding Field Names,** continued To get the definition of a field the user should select the relevant field on the form and click What's This.

**Selection:**      **Step:**



**1.**      With the relevant field on the form selected, click What's This.

**Result:** The pointer changes to an arrow with a question mark beside it.

**6.**      Click the field again.

**Result:** A Windows Help form appears with the field details.

## Screen Field Selection Form

Solution View Screen Field Selection: XBDSCR Option:

Field Names:	EMPLOYEE-NAME	SEX-CODE
	BIRTH-DATE	EMPLOYMENT-DATE
	CONTROL-3-CODE	CONTROL-3
	JOB-CODE	MARITAL-STATUS
	EMPLOYEE-STATUS	HOME-PHONE
	SALARY-PER-PERIOD	

Screen Prompts

Use As Prompts

EF-SCR Continue Employee Basic Data
55-SCR Enroll into Company Plan
40-SCR Calculate Salary Information

Include Description

### NOTES

**DEFINING THE FIELDS**, continued

**Using the Form Field Selection Form**

To use the Form Field Selection form:

**Selection:**  
**Field Names**

**Step:**

1. Type the name of the first field to be displayed on the new entry form.
2. Repeat step 1 until all field names have been entered.

**Use As Prompts**

3. Choose Use As Prompts if message text is a navigational prompt.

**Form Prompts**

4. Optionally, type one to three lines of prompt text to be displayed in the context-sensitive menu.

**Include Description**

5. Select the check box to include a description.



6. Click Save or press ENTER.

Errors on this form must be resolved before proceeding. You may either fix the error, or click Cancel to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.

## Online Form Description Form

Solution View Screen Writer: XBDSCR      Option:

Online Screen Description Entry

Screen Description  
This form is used to enter basic employee information during the new hire process. Added information can be entered by using the EF-SCR, 55-SCR, and 40-SCR forms.  
THIS IS A USER-DEFINED, NON-CYBORG FORM

Solution View Screen Writer: XBDSCR

Screen program XBDSCR has been RELOADED and may now be run.

- Return to the start of the Solution View process
- Execute screen program XBDSCR

---

### NOTES

---

## CREATING HELP FORM TEXT

**Online Form Description Entry Form** The Online Form Description Entry form is used to enter a description of the form that is displayed when the HELP program is run.

**Field Descriptions** The Online Form Description Entry form contains the following fields:

- **Screen Description**—allows you to enter a description of this new form which will be displayed by the HELP program function. Four lines are available for this purpose.

An example might be, “This form is used to enter basic employee information during the new hire process.”

**Using the Online Form Description Entry Form:** To use the Online Form Description Entry form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Form Description</b>   | 1. Type a one- to four-line explanation of the form’s purpose for use by the HELP form display. |
|  | 2. Click Save or press ENTER.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or click Cancel to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** If this form is error-free, the program source code is written and reloaded, and the prompt form displays. The Prompt form informs you that your Form program has been written and compiled successfully. Cyborg Scripting Language Code now exists for the Form program.

**Practice** Using the worksheet, please complete this form for your program.

## Accessing the New Form

EMPLOYEE BASIC DATA FORM		AUSTIN, STEVEN F		
EMPLOYEE-NAME	SEX CODE	BIRTH-DATE	EMPLOYMENT DATE	CONTROL 3-CODE
AUSTIN, STEVEN F	M	07-15-1959	09-03-1983	01
CONTROL-3	JOB CODE	MARITAL-STATUS	EMPLOYEE-STATUS	HOME PHONE
Midwest	17857	Divorced	Active-Hrly PT	9719923
SALARY PER-PERIOD				
692.31				

- Save This Form
- Cancel This Form
- Select An Employee...
- Show Selection

---

- Continue Employee Basic Data
- Enroll into Company Plan
- Calculate Salary Information

### NOTES

## ACCESSING THE NEW FORM

**Selecting Segment Keys** You may now access the new form program directly by using the Form field in the Command Dialog Box, the Command Line, or the button on the Prompt form to execute the program.

If the form includes any multiple occurrence segment fields, you will be presented with the **first** occurrence of all non-dated keyed segments or the **current** occurrence of all dated key segments.

**Edits and Calculations** Edits and calculations associated with the field on its original form **DO NOT** become part of that field's characteristics when you use it on a new form. Any table references that the field made on its original form are not made on the new form. These features can be added using the Cyborg Scripting Language tools.

**Windows Format** To access your new form in a Windows format (GUI) (displaying, for example, drop-down option lists), you need to create a form appearance table using the FormBuilder utility. This is discussed in Cyborg's courses on Cyborg Scripting Language.



**ACCESSING THE NEW Form**, continued

**Selecting Segment Keys**,  
continued

Optionally, you may use the exclamation point (!) in the Key field to get a listing of existing keys.

Type an exclamation point into the first position of any data field for that segment. By doing this, the form returns, in standard Inquiry mode, to allow for the selection of a particular occurrence of the segment, or the entry of new Key field data to establish another occurrence of the segment.

If an occurrence of a particular segment does not exist for an employee, the exclamation point is automatically populated into the first character of the first field for that segment. This insures that the form will go to the Inquiry mode for key field population.

## Selecting Segment Keys

EMPLOYEE BASIC DATA FORM		AUSTIN, STEVEN F	
JOB EFFECTIVE	KEY SEPARATOR	CHANGE TYPE	JOB CODE
06-30-1981	9	130	17857
12-17-1978	9	A02	@

EMPLOYEE BASIC DATA FORM		AUSTIN, STEVEN F		
EMPLOYEE-NAME	SEX CODE	BIRTH-DATE	EMPLOYMENT DATE	CONTROL 3-CODE
AUSTIN, STEVEN F	M	07-15-1959	09-03-1983	01
CONTROL-3	JOB CODE	MARITAL-STATUS	EMPLOYEE-STATUS	HOME PHONE
Midwest	15405	Divorced	Active-Hrly PT	9719923
SALARY PER-PERIOD				
692.31				

### NOTES

**ACCESSING THE NEW Form**, continued

**Selecting Segment Keys,**  
continued

Make your selection from the listing using the at sign (@) to select from the inquiry list and continue with the next key field.

When all segment keys have been selected, you may proceed with the data entry for the entry form.

You may NOT delete existing forms with the Delete This Entry function because they may contain multiple segments on a single form. Instead, you must delete each segment occurrence from its own form.

## Form Inquiry

Solution View Tool Kit

Option:

Name of Program:  Title:

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Screen Writer: XBDSCR

Option:

Screen Program General Information

Program Title: EMPLOYEE BASIC DATA FORM

Module: Security Code:

Author: S.O. Last Modified: 09-19-02 By: S.O.

This form is used to enter basic employee information during the new hire process. Added information can be entered by using the EF-SCR, 55-SCR, and 40-SCR forms.  
THIS IS A USER-DEFINED, NON-CYBORG FORM

Continue the WRITER inquiry

## NOTES

## INQUIRY, CHANGING, COPYING, DELETING

### Inquiry Capability

Once you have created a Form Writer program, the capability of viewing the form series through an inquiry function is provided. While in the inquiry mode, you may change a Form Writer program by selecting the Set Inquiry to Entry in the Option field of the form(s) you want to change.

### Accessing the Inquiry Feature

To use the Inquiry feature:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Form program name, (for example, XBDSCR).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Entry Form.

**Action**

4. Select Inquiry.



5. Click Save or press ENTER.

**Result:** An inquiry version of the first form in the series displays.

6. Press ENTER to move to each successive form in the series.

## Changing a Form

The screenshot shows the 'Solution View Tool Kit' window. At the top right is an 'Option:' dropdown menu. Below it are two input fields: 'Name of Program:' with the text 'XBDSCR' and 'Title:'. The main area is divided into two columns of radio button options. The left column, titled 'Program Type', includes: 'Entry Screen' (selected), 'New User Fields', 'Query', 'Report', 'PC Download', and 'Extract Routine'. The right column, titled 'Action', includes: 'Add', 'Change' (selected), 'Delete', 'Inquiry', 'Program List', 'Program List By User', and 'Copy Program'. At the bottom, there is a 'Module:' dropdown menu and a 'Security Code:' input field. A footer note states: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

The dialog box is titled 'Warning Messages' and contains a '+' icon and a checkbox labeled 'Change indicated: Title/Module/Security acceptable?'. Below this is an 'Information Messages' section with a question mark icon. The text explains that a warning message indicates a problem found by Cyborg, and while it can proceed, the warning allows for a review. An example is given: 'An example of a Warning message would be entering an employee start date and birth date that make an...'

## NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Changing a Program**

Once you have created a Form Writer program you have the option of changing the program. You may use the Inquiry option on the Action form and select the Set Inquiry to Entry option in the Option field of the form(s) you want to change, or use the Change option on the Action form.

You may want to make a copy of the original form program using the Program Copier form. You can then change the duplicate and have two different form programs on file.

**Accessing the Change Option Form**

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Form program name, (for example, XBDSCR).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Title**

3. Type a new description of the Entry Form (optional).

**Program Type**

4. Select Entry Form.

**Action**

5. Select Change.

**Module**

6. Type a new module ID (optional).

**Security Code:**

7. Type a new security code (optional).



8. Click Save or press ENTER.

**Note:** If you did not change the Title, Module, or Security Code fields and they were previously entered, a warning is displayed. This allows you to change the fields or leave them as they are and execute the form.

**Result:** You are now given the ability to alter the Form program.

## Copying a Form

The screenshot shows the 'Solution View Tool Kit' interface. At the top left is the title 'Solution View Tool Kit' and at the top right is an 'Option:' dropdown menu. Below this is a form with several fields and sections:

- 'Name of Program:' with the text 'XBDSCR' and a 'Title:' field.
- 'Program Type' section with radio buttons for: Entry Screen (selected), New User Fields, Query, Report, PC Download, and Extract Routine.
- 'Action' section with radio buttons for: Add, Change, Delete, Inquiry, Program List, Program List By User, and Copy Program (selected).
- 'Module:' dropdown menu and 'Security Code:' text box.

At the bottom of the window, it says: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

### NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Copying a Program**

You may copy and rename any existing Form Writer program using the Program Copier form accessed from the Action form. You can use the copy to make modifications and have two different programs on file.

**Accessing the Program Copier Form**

To access the Program Copier form:

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created Form program name, (for example, XBDSR).

**Note:** If you are not sure which programs are on file, select either the Program List by User or Program List option to view them.

**Program Type**

3. Select Entry Form.

**Action**

4. Select Copy Program.



5. Click Save or press ENTER.

**Result:** The Program Copier form displays.

## Program Copier Form

The screenshot shows a software interface titled "Solution View Screen Writer: XBDSCR". At the top right, there is a label "Option:" followed by a dark blue dropdown menu. Below this is a light green rectangular area containing the text "Entry Screen Program Copier". Inside this area, there are four input fields: "New Name:" with the value "X", "Original Title:" with the value "EMPLOYEE BASIC DATA FORM", "Module:" with an empty dropdown menu, and "Security Code:" with an empty text box.

---

### NOTES

---

## INQUIRY, CHANGING, COPYING, DELETING, continued

**Field Descriptions**      The Program Copier form contains the following fields:

- **New Name**—used to enter the new Form Writer name.

All form program names must begin with X. The new form name must be unique, in other words, it cannot be a name that already exists.

- **Original Title**—displays the current title of the Form Writer program that is being copied. Optionally, you may change this title for the copied form. Press TAB to allow the current entry to remain.
- **Module**—displays the current module ID associated with the program. Optionally, you may change this entry for the copied form. Press TAB to allow the current entry to remain.
- **Security Code**—displays the current security code associated with the program. Optionally, you may change this entry for the copied form. The Security Code must not be higher than your assigned security level. Press TAB or press ENTER to allow the current entry to remain.

## Program Copier Form, continued

Solution View Screen Writer: XBDSCR      Option: ▼

Entry Screen Program Copier

New Name: XEISCR

Original Title: EMPLOYEE INFORMATION II

Module: ▼

Security Code:

Solution View Screen Field Selection: XEISCR      Option: ▼

Field Names:	EMPLOYEE-NAME	SEX-CODE
	BIRTH-DATE	EMPLOYMENT-DATE
	CONTROL-3-CODE	CONTROL-3
	JOB-CODE	MARITAL-STATUS
	EMPLOYEE-STATUS	HOME-PHONE
	SALARY-PER-PERIOD	

Screen Prompts

Use As Prompts      EF-SCR Continue Employee Basic Data

55-SCR Enroll into Company Plan

40-SCR Calculate Salary Information

Change Description

To delete a field name clear the desired entry

## NOTES

**INQUIRY, CHANGING, COPYING, DELETING, continued**

**Using the Program Copier Form**

To use the Program Copier form:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>New Name</b>   | <b>1.</b> Type a name for the form copy, (for example, XEISCR).             |
| <b>Original Title</b>   | <b>2.</b> Type a new title for the copied form (optional).                  |
| <b>Module</b>   | <b>3.</b> Select a new module ID to be associated with the form (optional). |
| <b>Security Code</b>  | <b>4.</b> Type a new Security Code for the form (optional).                 |
|  | <b>5.</b> Click Save or press ENTER.  |

**Result:** You are now given the ability to alter the copied program.

## Deleting a Form

Solution View Tool Kit

Option:

Name of Program:  Title:

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Screen Writer: XBDSCR

Option:

Name of Program:  Title:

Program Type

- Entry Screen
- New User Fields
- Query
- Report
- PC Download
- Extract Routine

Action

- Add
- Change
- Delete
- Inquiry
- Program List
- Program List By User
- Copy Program

Module:  Security Code:

\*\* SCREEN program XBDSCR has been deleted \*\*

### NOTES



## Section Summary

- **Planning Ahead: W.I.I.F.M.**
- **The Form Writer Form Series**
- **Selecting the Action**
- **Defining the Fields**
- **Creating Help Form Text**
- **Accessing the New Form**
- **Inquiry, Changing, Copying, Deleting**

---

### NOTES

---

## SECTION SUMMARY

In this section, you learned how to create programs for employee-level entry forms using existing segments and fields. Specifically you learned:

- Planning Ahead: W.I.I.F.M.

---

---

---

- The Form Writer Form Series

---

---

---

- Selecting the Action

---

---

---

- Defining the Fields

---

---

---

- Creating Help Form Text

---

---

---

- Accessing the New Form

---

---

---

- Inquiry, Changing, Copying, Deleting

---

---

---

## Section 7 Exercise

---

### NOTES

---

## SECTION EXERCISE

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Using the requirements worksheet and the memo on the following pages, complete the Entry Form program specified by the memo and worksheet.
2. Use the new form to make an entry for employee 1001.

## FORM WRITER

### MEMO

**TO:** Cyborg End User  
**FROM:** The Boss  
**SUBJECT:** Create a New Benefits Form

---

The Benefits Department has requested a special form to track information pertinent to the company paid life insurance plan. The data they need to track is currently stored on The Solution Series, but on several different forms. Therefore, they would like to consolidate it all onto one form in order to make it easier to access and maintain.

Please create an entry form with the following information: birth date, annual salary, adjusted seniority date, smoker ID, sex code, union code, and union.

The 59DSCR form will be entered after this form has been completed, so you may want to set up a form prompt.

Don't forget to document the form for HELP.

Feel free to contact me if you have any questions.

FORM WRITER

Date: \_\_\_\_\_

<b>Program Name: X <u>B</u> <u>A</u> <u>S</u> <u>C</u> <u>R</u></b>		<b>Module Specific?</b>	<b>Y / N YES</b>
<b>Program Title:</b>		<b>Security?</b>	<b>Y / N NO</b>
<b>Sequence</b>	<b>Field Name</b>	<b>Sequence</b>	<b>Field Name</b>
	BIRTH-DATE		
	ANNUAL-SALARY		
	ADJUSTED-SENIORITY		
	SMOKER-ID		
	SEX-CODE		
	UNION-CODE		
	UNION		
<b>Form Prompts:</b>			
<b>Help Description:</b>			

\* This worksheet has been started for you. You must complete it with additional information from the letter.



## **SECTION 8: CREATING A PROGRAM TO DEFINE NEW FIELDS AND CREATE AN ENTRY FORM**

---

---

### TABLE OF CONTENTS

SECTION OVERVIEW .....	378
PLANNING AHEAD: W.I.I.F.M. ....	380
THE NEW FIELDS DEFINITION FORM SERIES .....	388
SELECTING THE ACTION .....	390
TITLING AND DESCRIBING THE NEW FORM .....	392
DEFINING THE FIELDS .....	396
ACCESSING THE NEW FORM .....	400
NEW FORM SAMPLE .....	402
DELETING THE NEW FIELDS DEFINITION FORM .....	404
SECTION SUMMARY .....	406
SECTION EXERCISE.....	408

## Section Objectives

- **Determine the content and format of the new entry form**
- **Identify the form series that will be used to create the user defined fields and data entry form**
- **Optionally, override the segment assigned by the Writer program**
- **Define the form title, module ID, and security code**
- **Provide a form description and form prompts**
- **Define the field name, length, type, key, and option list**
- **Analyze the new data entry form to verify the field definitions and the segment structure**
- **Delete the form program**

---

NOTES

---

## SECTION OVERVIEW

### Purpose

In this section you will learn how to define new fields and create Cyborg Scripting Language form programs using your user-defined fields.

### Objectives

When you have completed this section you will be able to do the following:

- Determine the content and format of the new entry form
- Identify the form series that will be used to create the user defined fields and data entry form
- Optionally, override the segment assigned by the WRITER program
- Define the form title, module ID, and security code
- Provide a form description and form prompts
- Define the field name, length, type, key, and option list
- Analyze the new data entry form to verify the field definitions and the segment structure
- Delete the form program

## Requirements Memo

### MEMO

TO: Cyborg End-User  
FROM: The Benefits Administrator  
DATE: August 5, 199X  
SUBJECT: System Tracking of Awards Program

---

As you may know, our awards program begins in January of next year. I will need access to a new online form to enter individual employee award information.

Since the awards can be monetary in nature, I'll be contacting the payroll manager to set up a new earnings HED for the payout of awards. Monetary awards are given at 5, 10, 15, and 20 year intervals. Awards can also be gifts or letters of recognition.

The online form will need to display a date field, followed by the award type code and description, and then an amount field for any dollars awarded. Let's create a new option list for the award type, maybe two positions in length. Give us at least a fifteen character description.

When you have it programmed, please give me a call so I can check it out. If it works properly, we'll notify all department managers that we have the tracking form installed. They can then submit candidates and recommended awards starting January 1st.

Let me know if you need any additional information.

---

### NOTES

---

## PLANNING AHEAD: W.I.I.F.M.

### What's In It For Me?

New Fields Definition Writer is an end-user tool that enables you to create new fields and a data entry form to maintain the fields. The facility creates the program and field definitions from entries you make.

You can use New Fields Definition Writer to create both employee-level and company-level forms with up to 15 entry fields. This means that you can create your own entry form with your own fields using the Solution View tools. You are not required to know the Cyborg Scripting Language programming language.

We encourage Relational Database Management System users to employ this method rather than Cyborg Scripting Language programming to create new fields. New Fields Definition Writer automatically adds both key and non-key fields and the special FIND- verb to the Query Maintenance Facility Cross Reference Table.

### Planning the Form

Prior to completing the New Fields Definition Writer form series that will create the form program, we suggest that you invest some time in reviewing the original request. This review allows you to define your new fields and the contents of your form program prior to entering the field and form parameters.

The sample memo on the facing page requests new fields and a new form to resolve some information needs. Using this example, it is clear that this manager wants to be able to use a new online form with user-defined fields to track the new employee awards. Starting with the specifications that are stated on the memo, let us plan what fields will appear on the entry form.

### Requirements Worksheet

You might want to use a requirements worksheet to plan your form program. A worksheet would allow you to research the field and form content and detailed specifications before you begin your entries. Once those requirements have been committed to paper, the process of using the Solution View form series becomes an easy next step.

On the next page, we will walk through a sample worksheet that you could use to specify the field and form requirements.

# Requirements Worksheet

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X _____				<b>Module Specific?</b> Y / N			
<b>Program Title:</b>				<b>Security?</b> Y / N			
<b>Data (circle one):</b> Employee / Company				<b>Segment Code Override:</b>			
<b>Help Description:</b>							
<b>Form Prompts:</b>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field?	Is this a Required Non-Key Field?	Option List Name

**FIELD TYPE choices:** Alphanumeric, Date, Option List, Option List Description

Valid **Company** level segments are:  
Segment type C, Codes Bx and Cx, where x is one alphanumeric character

Valid **Employee** level segments are:  
Segment type L, codes Lx, Mx, and Nx, where x is one alphanumeric character

Segment codes are automatically assigned in descending order:

Company (C segment):  
C9, C8, C7, ...CZ, CY,...B9, B8,...

Employee (L segment):  
N9, N8,...NZ, NY,...M9, M8,...MZ, MY

---

## NOTES

---

**PLANNING AHEAD: W.I.I.F.M., continued**

**Using the Worksheet** Your worksheet might look something like the sample on the facing page. The worksheet contains the following specifications:

- **Program Name**—What is the name of the program that will produce the new fields and form? As you recall, all user-defined program names are six characters and must begin with X. For form programs, you may choose to end the name with SCR to define the program as a form program.
- **Program Title**—What is the title that will appear at the top of the form? There is a 30-character limit for a form title.  
  
Single quotes and apostrophes are not allowed in the title. Brackets ( [ ] ), parentheses ( ), and greater-than and less-than signs (< >) are also not allowed.
- **Data**—Will this form contain employee OR company level data?
- **Module Specific?**—Do you want to specify a Module ID to be associated with this form?
- **Security?**—Do you wish to specify the security to be associated with this form?
- **Segment Code**—Do you want to override the segment code value for these fields? You may override the automatic assignment if you are already using any of these segment codes. If so, enter it here.

The New Fields Definition Writer creates the new fields and automatically assigns them to a new segment. The graphic on the facing page lists the valid Company and Employee Level segments.

Segment codes are automatically assigned in descending order. They are assigned as shown in the graphic on the facing page.

**Note:** You also have an override capability for selecting a specific segment code when you access the General Information form. This function is helpful for those of you who created some user-defined segments prior to using Solution View.

- **HELP Description**—What is the description of this form to be displayed in the HELP function? There is a limit of four lines.
- **Form Prompts**—What prompt(s) are to be displayed at the bottom of the form? There is a limit of three lines.

## Requirements Worksheet, continued

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X _____						<b>Module Specific?</b> Y / N	
<b>Program Title:</b>						<b>Security?</b> Y / N	
<b>Data</b> (circle one): <b>Employee / Company</b>						<b>Segment Code Override:</b>	
<b>Help Description:</b>							
<b>Form Prompts:</b>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field?	Is this a Required Non-Key Field?	Option List Name

**FIELD TYPE choices:** Alphanumeric, Date, Option List, Option List Description

Valid **Company** level segments are:  
Segment type C, Codes Bx and Cx, where x is one alphanumeric character

Valid **Employee** level segments are:  
Segment type L, codes Lx, Mx, and Nx, where x is one alphanumeric character

Segment codes are automatically assigned in descending order:

Company (C segment):  
C9, C8, C7, ...CZ, CY,...B9, B8,...

Employee (L segment):  
N9, N8,...NZ, NY,...M9, M8,...MZ, MY

---

### NOTES

---

PLANNING AHEAD: W.I.I.F.M., continued

- Using the Worksheet**
- **Sequence**—In what sequence should the fields appear on the form from left to right?
  - **New Field Name**—What are the field names that will be used in the entry form? List key fields first in the case of a multiple occurrence segment. The total length of all fields cannot exceed 68 characters; for Company segments, the limit is 81 characters.  
  
**Note:** These **must not be** previously defined field names. They may not have imbedded spaces; use hyphens to link words in a field name.
  - **Length of Field**—What is the length of this field? (Option list Descriptions and Date fields require no entry here. They are 20 and 6 positions respectively.)
  - **Field Type**—What type of field is this? See the worksheet legend for options.
  - **If Numeric, Number of Decimals**—For numeric fields, how many decimals should be used (0 to 6)? For numeric fields with no decimal places, do you want to show a leading \$?
  - **Is this a Key Field?**—Is this field defined as a key field for a multiple occurrence? If yes, this field must be sequenced first. The total characters in all key fields must not exceed 12.
  - **Is this a Required Non-Key Field?**—Is this a non-key field that must be filled when entries are made?
  - **Option List Name**—If this is an option list field, to which existing option list should it be attached? If this is a new option list, indicate the option list name here.

**Note:** The process required to create a new option list is detailed in course 2302—Online Cyborg Scripting Language Programming and in the on-line documentation.

## Requirements Worksheet, continued

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> <u>X A W A R D</u>					<b>Module Specific?</b> Y / N <i>NO</i>		
<b>Program Title:</b> <i>SERVICE AWARD FORM</i>					<b>Security?</b> Y / N <i>NO</i>		
<b>Data</b> (circle one): <b>Employee / Company</b> <i>EMPLOYEE</i>					<b>Segment Code Override:</b>		
<b>Help Description:</b> <i>This form is used to record service awards granted to an employee. Awards may be letters of recognition, monetary awards, or gifts.</i>							
<i>THIS IS A USER-DEFINED, NON-CYBORG FORM.</i>							
<b>Form Prompts:</b>							
<i>ENTER: HH-SCR to pay monetary value of award.</i>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field?	Is this a Required Non-Key Field?	Option List Name
1	<i>AWARD-DATE</i>						
2	<i>AWARD-TYPE</i>	<i>02</i>				<i>YES</i>	<i>HR85</i>
3	<i>AWARD-DESCRIPTION</i>	<i>15</i>	<i>CD</i>				<i>HR85</i>
4	<i>AWARD-AMOUNT</i>	<i>08</i>	<i>N2</i>	<i>2</i>			

**FIELD TYPE choices:** Alphanumeric, Date, Option List, Option List Description

<p>Valid <b>Company</b> level segments are: Segment type C, Codes Bx and Cx, where x is one alphanumeric character</p> <p>Valid <b>Employee</b> level segments are: Segment type L, codes Lx, Mx, and Nx, where x is one alphanumeric character</p>
---

<p>Segment codes are automatically assigned in descending order:</p> <p>Company (C segment): C9, C8, C7, ...CZ, CY,...B9, B8,...</p> <p>Employee (L segment): N9, N8,...NZ, NY,...M9, M8,...MZ, MY</p>
--

---

### NOTES

---

**PLANNING AHEAD: W.I.I.F.M., continued**

**Check Your Requirements**

The completed worksheet on the facing page is an example of specifications that will be used to define the form program to be created using the New Field Definition Writer form series. When you have completed your worksheet, you will have a list of the fields and their characteristics, as well as prompt and HELP display information.

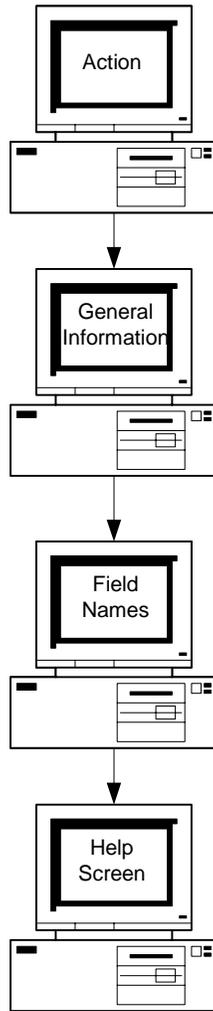
From what we have specified here, we are able to deduce the following:

- that this form will have three entry fields
- that there is one key field, the date
- that there is a required option list field
- that there is a new option list, which will have to be populated prior to using the new form

**What's Next?**

Now you are ready to use the Solution View tools to enter your program requirements. We will start by previewing the form series for a New Fields Definition program.

## Form Writer Form Series



---

### NOTES

---

## THE NEW FIELDS DEFINITION FORM SERIES

### New Fields Definition Writer Forms

New Fields Definition Writer is an end-user tool that enables you to create new fields and a data entry form to maintain the fields. The facility creates the program and field definitions from entries you make. One form requires general information about the new form, such as its name and title. Another form asks for the field list for the new form.

You can use New Fields Definition Writer to create both employee-level and company-level forms with up to three lines of entry fields.

*New Fields Definition Writer forms may only be created by persons having unlimited System Control Repository access on the Cyborg system. The fields you create through Solution View are automatically added to Field Name Tables.*

**Note: You cannot create Table forms with this facility.**

These are the forms that are available in Solution View to create your form program.

- **Action form**—allows you to name the form program, give the form a title, and select the type of action to be performed
- **General Information form**—used to specify the form title, description and prompts
- **New Field Name Entries form**—used to define the fields to be displayed on the new form
- **Help form**—displays to signify that the form program has compiled successfully and prompts you to use the new form

Now that you have previewed the form series that you will use to create the new fields and form, let us look at each task in detail, see what tools are provided on each form, and how to use them.

## Action Form

The screenshot shows a dialog box titled "Solution View Tool Kit" with a green background. At the top right, there is a label "Option:" followed by a dropdown menu. Below this, there are two text input fields: "Name of Program:" with the text "XAWARD" and "Title:" with the text "SERVICE AWARD FORM".

Below the text fields are two columns of radio button options:

- Program Type:**
  - Entry Screen
  - New User Fields
  - Query
  - Report
  - PC Download
  - Extract Routine
- Action:**
  - Add
  - Change
  - Delete
  - Inquiry
  - Program List
  - Program List By User
  - Copy Program

At the bottom of the dialog, there are two more fields: "Module:" followed by a dropdown menu, and "Security Code:" followed by a small square input field.

At the very bottom of the dialog, there is a line of text: "Solution View is a Registered Trademark of Cyborg Systems, Inc."

### NOTES

## SELECTING THE ACTION

**Action Form** The first New Fields Definition Writer form, the Action form, is used to identify the Writer program and takes you directly to that form series.

**Accessing the Form** To access the Action form:

**Selection:**

**Step:**

**Navigator**



User Tools

User Tools



Solution View

**Result:** The Action form displays.

**Using the Action Form**

To add a New Fields Description program:

**Selection:**

**Step:**

**Name of Program**

1. Type the one- to six-character name of the program, beginning with X, (for example, XAWARD).

**Title**

2. Type a title for the New Fields Description form, (for example, Service Award Form) (optional).

**Program Type**

3. Select New User Fields.

**Action**

4. Select Add.

**Module**

5. Select the module ID to be associated with the report (optional).

**Security Code**

6. Type the Security Code for the report (optional).



7. Click Save or press ENTER.

**Result:** When the Action form is error-free, the General Information form displays.

**Practice**

Using the worksheet, please complete this form for your program.

## General Information Form

Solution View - New Fields Definition \*\*\*\*\* General Information \*\*\*\*\*

Screen Name	Screen Title	Security Code	Module Code	Employee Data
XAWARD	SERVICE AWARD FORM	**	PP	Y

Screen Description

Use As Prompts

### NOTES

## TITLING AND DESCRIBING THE NEW FORM

**General Information Form** The General Information form requests basic information about the form you are creating. Some fields on this form are optional.

**Field Definitions** The General Information form contains the following fields:

- **Form Name**—displays the name that was entered on the Action form. This field is automatically filled.
- **Form Title**—used to enter a one- to thirty-character title for the form program. This title displays at the top of the entry form.
- **Security Code**—(optional) displays the Security Code entered on the Action form.
- **Module Code**—(optional) displays the Module Code entered on the Action form.
- **Employee Data**—used to specify either Y (the default) for an employee-level form, or N to create a company level form.
- **Form Description**—(optional) used to enter HELP form documentation by typing a one- to four-line explanation of the form's purpose.
- **Form Prompts**—(optional) used to create prompt message text that appears on the bottom of the newly created form.

If you wish to include the message text as a navigational prompt, choose the Use As Prompts check box to create an entry on the context-sensitive menu. The prompt text must be in the format “XXXSCR Form title” (where XXX are the first three characters of the form program name).

If you have a non-windows system, you may wish to precede the form program name and title with Enter, followed by a colon and a space.

Overriding the Segment Code

**Use this form to override the segment code, if applicable. Type the override segment code in the Additional Key field of the Command Line before executing the form.**

## General Information Form, continued

Solution View - New Fields Definition \*\*\*\*\* General Information \*\*\*\*\*

Screen Name	Screen Title	Security Code	Module Code	Employee Data
XAWARD	SERVICE AWARD FORM	**	PP	Y

Screen Description

This form is used to record service awards granted to an employee. Awards may be letters of recognition, monetary awards, or gifts.

THIS IS A CUSTOMER-DEFINED, NON-CYBORG FORM

Use As Prompts

HH-SCR Earnings and Deductions

NOTES

**TITLING AND DESCRIBING THE NEW FORM, continued**

**Using the General Information Form**

To use the General Information form:

<u>Selection:</u>	<u>Step:</u>
<b>Additional Key Field</b>	1. Optionally, type an override segment code value.
<b>Form Name</b>	2. Bypass this field.
<b>Form Title</b>	3. Bypass this field.
<b>Security Code</b>	4. Bypass this field.
<b>Module Code</b>	5. Bypass this field.
<b>Employee Data</b>	6. Retain Y (the default) for an employee-level form, or type N to create a company level form.
<b>Form Description</b>	7. Type a one- to four-line explanation of this form's purpose (optional).
<b>Use As Prompts</b>	8. Choose Use As Prompts if message text is a navigational prompt.
<b>Form Prompts</b>	9. Type one to three lines of text that you want displayed at the bottom of the form (optional).
	10. Click Save or press ENTER.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option Field, and execute the form to return to the Action form. Then delete the program and re-enter it correctly.

**Result:** When this form is error-free, the next form in the series displays.

**Practice**

Using the worksheet, please complete this form for your program.



## DEFINING THE FIELDS

### New Field Name Entries Form

The third form in the New Fields Definition Writer form series is the New Field Name Entries form which allows you to define the fields that you want on the form. You must assign unique field names which may not have imbedded spaces; use hyphens to link words in a field name. Starting your field names with X prevents conflict with delivered CYBORG field names.

### Field Descriptions

The New Field Name Entries form contains the following fields:

- **New Field Name**—used to enter a 1- to 20-character field name. List key fields first.
- **Length**—used to specify the number of characters that the field requires (limit of 30). This field is required for all fields **except DT** fields (dates) **and CD** fields (option list descriptions). You may leave the Length entry field blank for DT and CD fields to assign the default values.
  - The Date field (DT) default is 10.
  - The Option list Description (CD) default is 20.

**Note:** The total characters in all key fields must not exceed 12 (stored length).

- **Type**—used to designate the field type, such as date, numeric, option list. Valid types are:
  - **blank** - alphanumeric
  - **CD** - Option List Description
  - **DT** - Date
  - **N0** - Numeric/0 Decimals
  - **N2** - Numeric/2 Decimals
  - **N4** - Numeric/4 Decimals
  - **N6** - Numeric/6 Decimals
  - **N\$** - Numeric/Leading \$
- **Key**—used to enter **K** to designate a field as a key, making the segment that is created a multiple occurrence segment. List key fields first in order of importance. More than one field can be combined to create a key. Enter **G** to designate a **required** non-key field when the form is entered.
- **Option list**—designates the field as an option list field, and allows you to specify the option list which validates this field.

## New Field Name Entries Form, continued

Solution View - New Fields Definition \*\*\*\*\* New Field Name Entries \*\*\*\*\*

New Field Name	Length	Type	Key	Codeset	Type:
XAWARD-DATE		DT	K		- Alphanumeric
XAWARD-TYPE	02		G	HR85	CD - Codeset Description
XAWARD-DESCRIPTION	15	CD		HR85	DT - Date
XAWARD-AMOUNT	08	N2			NO - Numeric/0 Decimals
					N2 - 2 Decimals
					N4 - 4 Decimals
					N6 - 6 Decimals
					N\$ - Leading \$

Key: Enter K for each key field.  
They must precede all non-key fields.  
Enter G for all fields that are required entries.

SERVICE AWARD FORM XAWARD

This form is used to record service awards granted to an employee. Awards may be letters of recognition, monetary awards, or gifts.  
THIS IS A CUSTOMER-DEFINED, NON-CYBORG FORM

Complete the text boxes below, then hit enter:

If needed, alter Org Level:

For inquiry only, enter 'I':

Enter EMPL #:  or SSN:

or NAME:

Enter XAWARD-DATE:

### NOTES

**DEFINING THE FIELDS**, continued

**Using the New Field Name Entries Form**

To use the New Field Name Entries form:

- | <u>Selection:</u>   | <u>Step:</u>   |
|---|--|
| <b>New Name Field</b>   | <b>1.</b> Type a unique, 1- to 20-character field name. (One way of assuring this is to prefix all new field names with <b>X</b> .) List all Key fields before non-key fields. |
| <b>Length</b>   | <b>2.</b> Type the number of characters that the field requires. Bypass this field to use the system default for a date or option list field.                                  |
| <b>Type</b>   | <b>3.</b> Type the code (listed on the right side of the form) that defines the field type.  |
| <b>Key</b>  | <b>4.</b> Type <b>K</b> if the field is part of the segment key. Type <b>G</b> if this is a required non-key field.  |
| <b>Option list</b>  | <b>5.</b> Type the name of the option list which validates the field.  |
|   | <b>6.</b> Repeat steps 1 through 5 for each unique field name that you want on the form.   |
|  | <b>7.</b> Click Save or press ENTER.   |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then delete the program and re-enter it correctly.

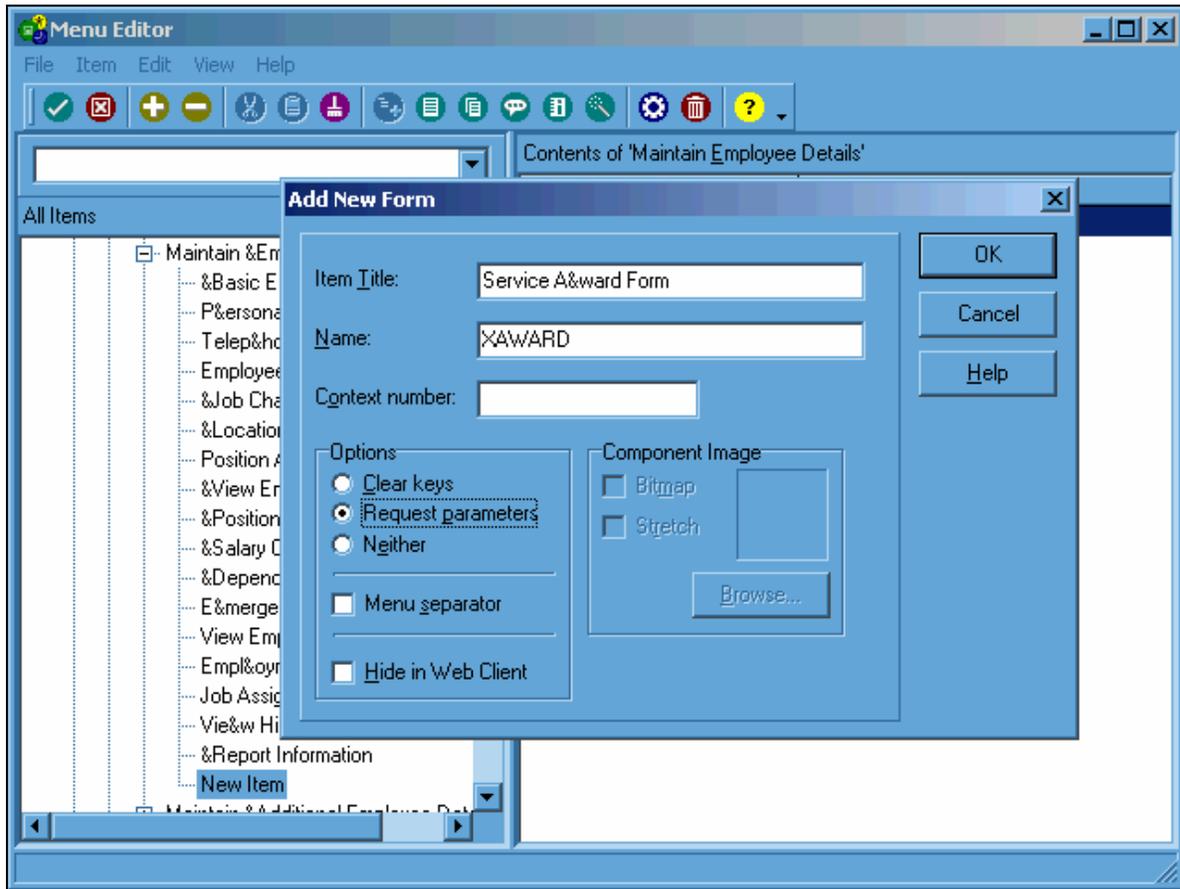
**Result:** If this form is error-free, the program source code is written and compiled, and the Help form displays.

**Note:** When using a PC and the PRODUCTION-VERSION switch is on, you will be shown the RELOAD is OK message instead of being taken to the Help form.

**Practice**

Using the worksheet, please complete this form for your program.

## Menu Editor



### NOTES

## ACCESSING THE NEW FORM

### Updating FILECL

After creating new option lists to be used by the entry form program, execute the UPDTCL program before using the new form for the first time. FILECL is a delivered indexed file that contains option lists, event menu records, field definitions, and some security information.

### Creating Menu Item

You need to create a new menu item to access the new form.

**Selection:**

**Navigator**

**Step:**

1.  Development Tools  
System Control Repository Utilities  
 Customize Menus...

**Result:** The Menu Editor displays.

**Select menu structure**

2. Select a place in the menu structure to insert the new item, (for example Menus + H&R + Employee &Resourcing + Maintain &Employee Details).



3. Insert the new item.

**Item Title.**

4. Type the title of the new item, (for example “Service A&ward Form”).

**Note:** The ‘&’ will make the ‘w’ in ‘Award’ a hot key.

**Name**

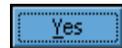
5. Type the name of the new item. This must be the name of the new form created, (for example XAWARD).

**Options**



7. Click OK or press ENTER.

8. Exit the menu editor.



9. Click Yes or press ENTER when the Confirm dialog appears.

10. Exit The Solution Series.

11. Log on to The Solution Series.

**Result:** The new menu item should be available and you should be able to launch the new form.

### Direct Form Access

As an option from this point on, you may access the new form directly by using the Command Entry dialog.

You may delete any existing occurrence by using the Delete This Entry function.

## New Form Sample

SERVICE AWARD FORM AUSTIN, STEVEN F

Award Date:

Award Type:

Award Amount:

SERVICE AWARD FORM AUSTIN, STEVEN F

Award Date:

Award Type:

Award Amount:

- Save This Form
- Cancel This Form
- Select An Employee...
- Show Selection
- Earnings and Deductions

### NOTES

## NEW FORM SAMPLE

### New Fields Definition Form

The form displayed on the facing page is the result of a New Fields Definition program that was created using Solution View.

### Verify Form & Fields

After creating the data entry form program, it may be desirable to verify the features of the new form to gather information for documenting the form's use. This verification includes, but is not limited to:

- All key fields are labeled with a (>) preceding the entry field.
- Valid values can be entered into each field:
  - Date fields accept data as MM-DD-CCYY.
  - Numeric fields accept the proper number of integers and decimals. Results display on the form properly.
  - Required fields must be entered, otherwise an error occurs.
  - Option list fields are edited against an option list, and display the correct literal description.
- For multiple occurrence segments, test the following functions from the button bar:
  -  **Move down**—move to the next oldest entry if the occurrence is stored by date, or the next highest numbered or lettered occurrence.
  -  **Move up**—move back up the occurrences until you reach the first entry.
  -  **Top of selection list**—move to the most recent occurrence or the top of the stacked segment.
  -  **Select**—provides a summary list of all the form's occurrences.

## Deleting a Program

Screenshot of the Solution View Tool Kit interface. The window title is "Solution View Tool Kit". At the top right, there is a dropdown menu labeled "Option:". Below this, there are two text input fields: "Name of Program:" containing "XAWARD" and "Title:". Underneath, there are two columns of radio button options. The "Program Type" column includes: Entry Screen, New User Fields (selected), Query, Report, PC Download, and Extract Routine. The "Action" column includes: Add, Change, Delete (selected), Inquiry, Program List, Program List By User, and Copy Program. At the bottom, there is a "Module:" dropdown menu and a "Security Code:" text input field. A footer message reads: "Solution View is a Registered Trademark of Cyborg Systems, Inc."

Screenshot of the Solution View Tool Kit interface after the deletion. The window title is "Solution View Tool Kit". At the top right, there is a dropdown menu labeled "Option:". Below this, there are two text input fields: "Name of Program:" containing "X" and "Title:". Underneath, there are two columns of radio button options. The "Program Type" column includes: Entry Screen, New User Fields, Query, Report, PC Download, and Extract Routine. The "Action" column includes: Add, Change, Delete, Inquiry, Program List, Program List By User, and Copy Program. At the bottom, there is a "Module:" dropdown menu and a "Security Code:" text input field. A confirmation message is displayed in the center: "\*\* NEWSCR program XAWARD has been deleted \*\*".

### NOTES

## DELETING THE NEW FIELDS DEFINITION FORM

**Deleting a Program** You may delete an existing New Fields Definition Writer program using the Action form.

**Note:** The system will delete the System Control Repository records for the form program, but retains the Employee Database data. Use the form first to delete any Employee Database data established before deleting the program.

### Accessing the Delete Feature

**Selection:**

**Step:**

**Navigator**

1.  User Tools  
User Tools  
 Solution View

**Result:** The Action form displays.

**Name of Program**

2. Type a previously-created New Fields Definition program name, (for example, XAWARD).

**Note:** These programs cannot be listed using the action form List option. Instead, you may use the Display Utility to list existing form programs.

**Program Type**

3. Select New User Fields.

**Action**

4. Select Delete.



5. Click Save or press ENTER.

**Result:** A message displays to inform you that the program has been deleted.

## Section Summary

- **Planning Ahead: W.I.I.F.M.**
- **The New Fields Definition Form Series**
- **Selecting the Action**
- **Titling and Describing the New Form**
- **Defining the Fields**
- **Accessing the New Form**
- **New Form Sample**
- **Deleting the New Fields Definition Form**

---

NOTES

---

## SECTION SUMMARY

In this section, you learned how to define new fields and create Cyborg Scripting Language form programs using your user-defined fields. Specifically you learned:

- Planning Ahead: W.I.I.F.M.

---

---

---

- The New Fields Definition Form Series

---

---

---

- Selecting the Action

---

---

---

- Titling and Describing the New Form

---

---

---

- Defining the Fields

---

---

---

- Accessing the New Form

---

---

---

- New Form Sample

---

---

---

- Deleting the New Fields Definition Form

---

---

---

## Section 8 Exercise

---

NOTES

---

## SECTION EXERCISE

**Purpose** This exercise gives you practice using the information you have learned in this section.

1. Using the requirements worksheet and memo on the following pages, complete the New Fields Definition program specified by the memo and worksheet.
2. Use the newly created form to make an entry for employee 1234.

## NEW FIELDS DEFINITION WRITER

### MEMO

**TO:** Cyborg End User  
**FROM:** The Project Leader  
**SUBJECT:** Creating a New Attendance Form

---

The Attendance Department has requested a special form to track information on unpaid absences. The data they need to track is not currently stored on The Solution Series. Therefore, they would like us to create several new fields for this purpose and display them for entry on a new form entitled "Employee Unpaid Absence". This will make it fairly simple to access and maintain.

Please create fields for the following data: starting date, ending date, absence code using option list TA01, absent hours, absent day of the week using option list TA04, and two yes/no fields called "In", and "Payroll notified".

As far as I know, we are not currently using any of the option lists that I have specified above, so please contact me about modifying them. You may need to populate the TA04 option list to test the form. Go ahead and use these two codes to start with: APPROVED-SUPERVISOR, and APPROVED MANAGER.

We'll want to use a form prompt to remind the entry operator to notify Payroll about the pay to be withheld.

Remember to document the form for HELP.

Feel free to contact me if you have any questions.

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X <u>A</u> <u>B</u> S C R						<b>Module Specific?</b> Y / N <i>NO</i>	
<b>Program Title:</b>						<b>Security?</b> Y / N <i>NO</i>	
<b>Date</b> (circle one): <b>Employee / Company</b> <i>EMPLOYEE</i>						<b>Segment Code Override:</b>	
<b>Help Description:</b>							
<b>Form Prompts:</b> <i>Payroll must be notified so that the correct payment of earnings can be made.</i>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field (K)?	Is this a Required Non-Key Field (G)?	Option List Name
	XSTART-DATE						
	XEND-DATE						
	XABS-CODE	2					
	XABSENCE-TYPE						
	XABSENT-HOURS						
	XDAY-OF-WEEK-CODE	3					
	XDAY-OF-WEEK						
	XCALLED-IN	1					<i>HR00</i>
	XAPPR-CODE	4					
	XAPPROVAL						
	XPAYROLL-NOTIFIED	1					<i>HR00</i>

\*This worksheet has been started for you. You must complete it with additional information from the letter.

**Section 7: Creating a Program to Enter Data by Using a Form**

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X ___ S C R						<b>Module Specific?</b> Y / N	
<b>Program Title:</b>						<b>Security?</b> Y / N	
<b>Date (circle one):</b> Employee / Company						<b>Segment Code Override:</b>	
<b>Help Description:</b>							
<b>Form Prompts:</b>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field (K)?	Is this a Required Non-Key Field (G)?	Option List Name

\*This worksheet has been left blank. You must complete it with information from the letter.



## APPENDIX A: EXERCISE ANSWERS

---

### TABLE OF CONTENTS

SECTION 3 EXERCISE ANSWERS.....	414
SECTION 4 EXERCISE ANSWERS.....	415
SECTION 5 EXERCISE ANSWERS.....	419
SECTION 6 EXERCISE ANSWERS.....	428
SECTION 7 EXERCISE ANSWERS.....	431
SECTION 8 EXERCISE ANSWERS.....	433

### SECTION 3 EXERCISE ANSWERS

1. Sign on to The Solution Series and access the Solution View Action form.

Follow the specific log on instructions for your site or the classroom setting.

2. From the Action form, display a list of all Query reports that have been created by all users.

Select Query from the Program Type field, Program List from the Action field, and execute the form.

3. Name two of the actions you can perform by using the Action form Option field.

Previous Form

Set Inquiry to Entry

Field Selection Menu

Goodby; Log Off System

Leave Field Help

Reload Program

Leave Menu

Restart Process

**SECTION 4 EXERCISE ANSWERS**

Using the requirements worksheet complete the Query report specified by the memo and worksheet. Run your completed report using the Query facility.

Solution View Tool Kit Option:

Name of Program:  Title:

<p>Program Type</p> <ul style="list-style-type: none"> <li><input type="radio"/> Entry Screen</li> <li><input type="radio"/> New User Fields</li> <li><input checked="" type="radio"/> Query</li> <li><input type="radio"/> Report</li> <li><input type="radio"/> PC Download</li> <li><input type="radio"/> Extract Routine</li> </ul>	<p>Action</p> <ul style="list-style-type: none"> <li><input checked="" type="radio"/> Add</li> <li><input type="radio"/> Change</li> <li><input type="radio"/> Delete</li> <li><input type="radio"/> Inquiry</li> <li><input type="radio"/> Program List</li> <li><input type="radio"/> Program List By User</li> <li><input type="radio"/> Copy Program</li> </ul>
---	---

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Query Writer: XAGNCY Option:

Query Options

<p>As Of Date <input type="text"/></p> <p>If omitted Query will show current data.</p>	<p>General Options</p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Selection Criteria</li> <li><input checked="" type="checkbox"/> Count Employees</li> </ul>
--	--

SECTION 4 EXERCISE ANSWERS, continued

Solution View Query Field Selection: XAGNCY Option:

Field Name:  Total Option:

Solution View Query Writer: XAGNCY Option:

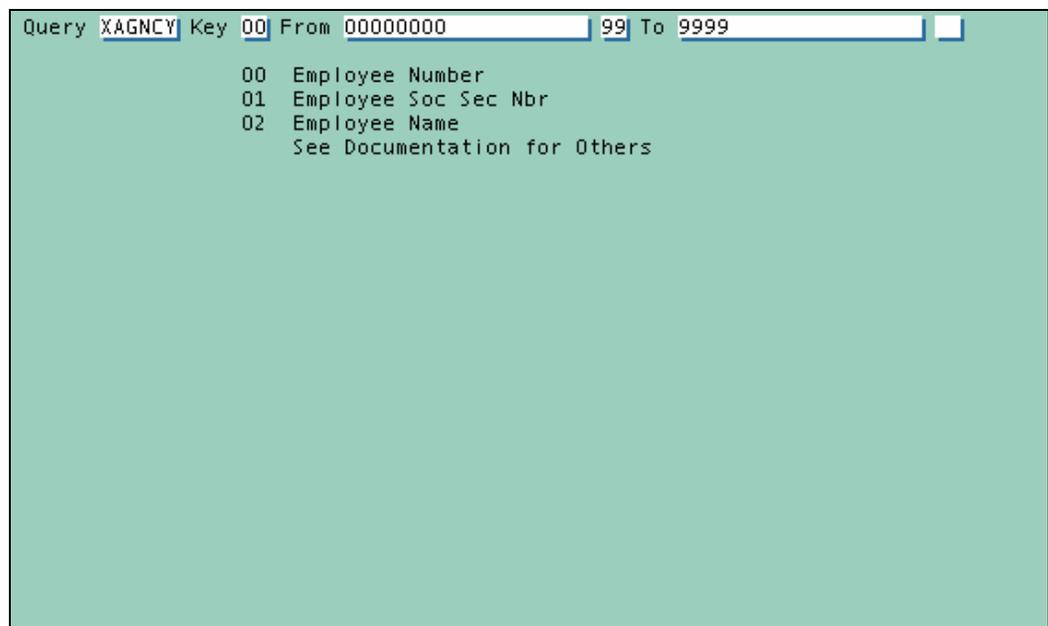
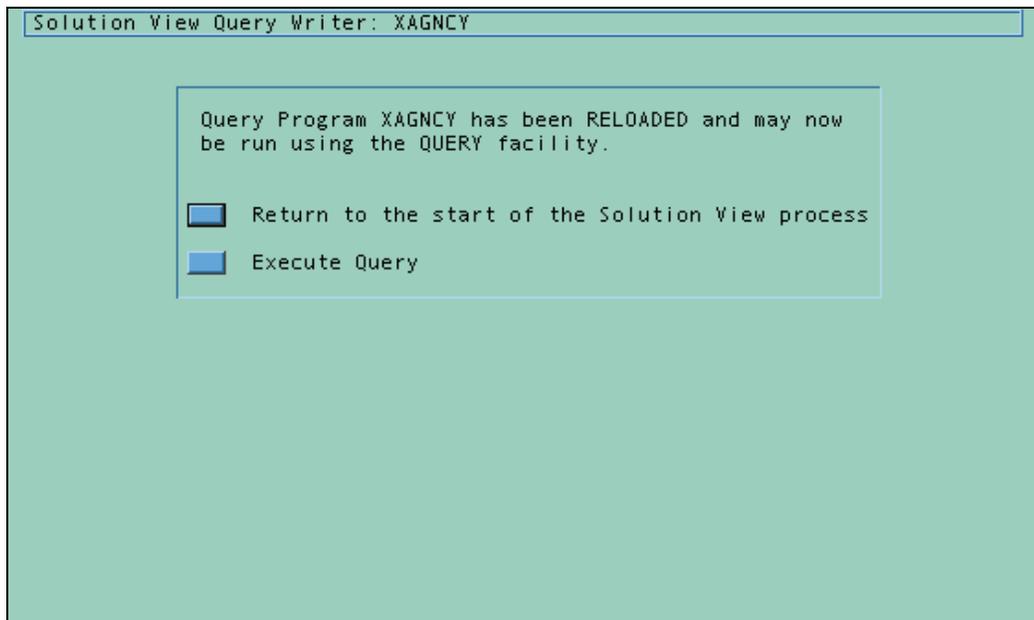
Field Name	Not	EGLF	Literal Compare Value	Action
HIRE-SOURCE	<input type="checkbox"/>	E	01	N
RESULTING-EMP-STATUS	<input type="checkbox"/>	E	01;03	P
	<input type="checkbox"/>			

Display Values

EGLF Values

Actions

## SECTION 4 EXERCISE ANSWERS, continued



SECTIONS 4 EXERCISE ANSWERS, continued

Query XAGNCY Key 00 From 7010 99 To 9999

EMPLOYMENT AGENCY STATISTICS

EMPLOYEE NAME-10	JOB-TITLE	AGENCY FEE	RELOCATION EXPENSE
MEYER, JUN	ACCOUNTANT	1,000	
MOORE, SAM	DIRECTOR OF SALES	63,000.00	2,000
MORITZ, KA	PURCHASING MANAGER	58,000.00	1,575
JONES, JER	NETWORK TECHNICIAN	31,350.00	2,000
JOHNSEN, R	DIRECTOR OF SALES	72,267.48	12,000 8,500
SWEENEY, BA	PURCHASING COORDINATOR	48,250.00	1,500
COLLINS, A	PURCHASING MANAGER	58,250.00	8,050
GRIMES, TH	DIRECTOR OF SALES	66,149.72	10,000 7,000
REYNOLDS,	ACCOUNTANT	48,750.00	1,000
BARNES, JO	DIRECTOR OF SALES	89,360.40	2,000

Totals for QUERY program XAGNCY ready to be displayed. Press your ENTER key.

TOTALS FOR QUERY PROGRAM XAGNCY:

EMPLOYEE COUNT:	10
AGENCY-FEE:	41,125
RELOCATION-EXPENSE:	15,500

----Complete----

- 00 Employee Number
- 01 Employee Soc Sec Nbr
- 02 Employee Name

See Documentation for Others

**SECTION 5 EXERCISE ANSWERS**

Using the requirements worksheet, complete the Report program specified by the memo and worksheet. Schedule, run, and view your completed report.

Solution View Tool Kit Option:

Name of Program:  Title:

<p>Program Type</p> <ul style="list-style-type: none"> <li><input type="radio"/> Entry Screen</li> <li><input type="radio"/> New User Fields</li> <li><input type="radio"/> Query</li> <li><input checked="" type="radio"/> Report</li> <li><input type="radio"/> PC Download</li> <li><input type="radio"/> Extract Routine</li> </ul>	<p>Action</p> <ul style="list-style-type: none"> <li><input checked="" type="radio"/> Add</li> <li><input type="radio"/> Change</li> <li><input type="radio"/> Delete</li> <li><input type="radio"/> Inquiry</li> <li><input type="radio"/> Program List</li> <li><input type="radio"/> Program List By User</li> <li><input type="radio"/> Copy Program</li> </ul>
---	---

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Report Writer: XSALPT Option:

Report Options

<p>As Of Date Options</p> <ul style="list-style-type: none"> <li><input checked="" type="radio"/> Use Current Date</li> <li><input type="radio"/> Use Entered Date</li> <li><input type="radio"/> Use Entered Range</li> </ul>	<p>General Options</p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Selection Criteria</li> <li><input checked="" type="checkbox"/> Count Employees</li> </ul>
--	--

SECTION 5 EXERCISE ANSWERS, continued

Solution View Report Field Selection: XSALPT Option:

Seq Nbr	Field Name	Control Break	Sort Seq	Print/Total	Calc Type	Result Length	Result Decimals
A 00	EMPLOYEE-NAME			P			
A 05	EMPLOYEE-NUMBER		2	P			
A 10	RATING-VALUE			P			
A 15	SALARY-EFFECTIVE			P			
A 20	SALARY-DATE-SPAN		1	P	T	06	
A 25	POSITION-IN-RANGE			P			
A 30	ANNUAL-SALARY			B			
A 35	PROJ-MERIT-INC-SAL			B	N	08	2

Display Values

Control Break     Print/Total     Calculation Types     All

Solution View Report Writer: XSALPT Option:

Result Field	Factor/Field Name	
SALARY-DATE-SPAN	= CURRENT-DATE	Subtract -
	SALARY-EFFECTIVE	

SECTION 5 EXERCISE ANSWERS, continued

Solution View Report Writer: XSALPT      Option: ▼

Calculation Routine Entry

Result Field	=	Factor/Field Name		Operator	
PROJ-MERIT-INC-SAL	=	ANNUAL-SALARY		Multiply *	<span style="border: 1px solid black; padding: 2px;">▼</span>
		105%			<span style="border: 1px solid black; padding: 2px;">▼</span>
		<span style="border: 1px solid black; padding: 2px;"> </span>			<span style="border: 1px solid black; padding: 2px;">▼</span>
		<span style="border: 1px solid black; padding: 2px;"> </span>			<span style="border: 1px solid black; padding: 2px;">▼</span>
		<span style="border: 1px solid black; padding: 2px;"> </span>			<span style="border: 1px solid black; padding: 2px;">▼</span>
		<span style="border: 1px solid black; padding: 2px;"> </span>			<span style="border: 1px solid black; padding: 2px;">▼</span>

Result: Numeric    Resulting Field Length: 08    Number of Decimals: 2

Solution View Report Writer: XSALPT      Option: ▼

Employee Selection Criteria

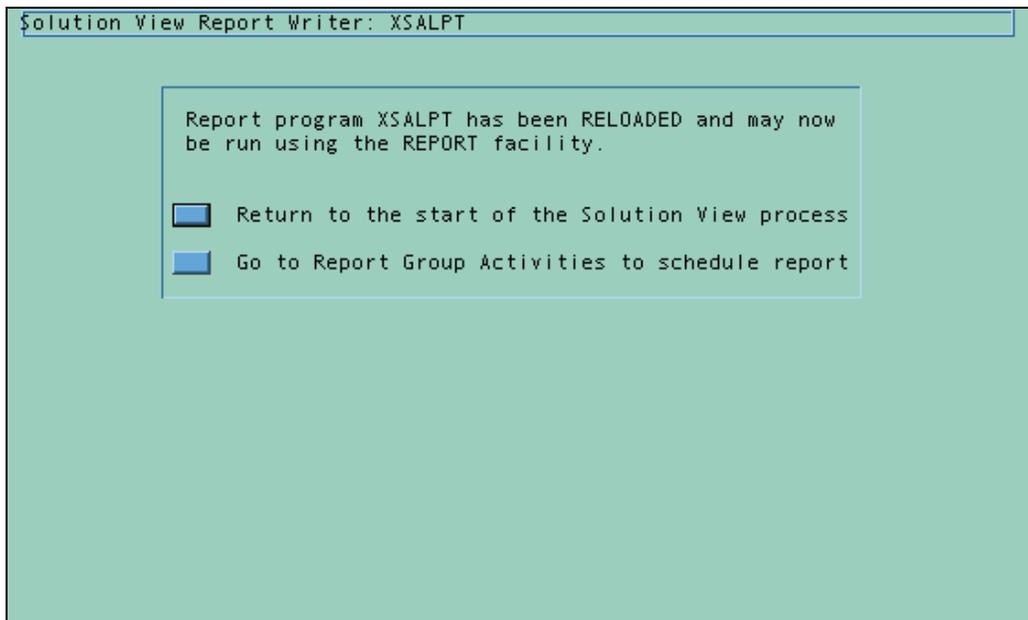
Field Name	Not	EGLF	Literal Compare Value	Action
RATING-VALUE	<input type="checkbox"/>	E	2	N
POSITION-IN-RANGE	<input checked="" type="checkbox"/>	G	80.00	P
<span style="border: 1px solid black; padding: 2px;"> </span>	<input type="checkbox"/>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>
<span style="border: 1px solid black; padding: 2px;"> </span>	<input type="checkbox"/>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>
<span style="border: 1px solid black; padding: 2px;"> </span>	<input type="checkbox"/>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>
<span style="border: 1px solid black; padding: 2px;"> </span>	<input type="checkbox"/>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>	<span style="border: 1px solid black; padding: 2px;"> </span>

Display Values

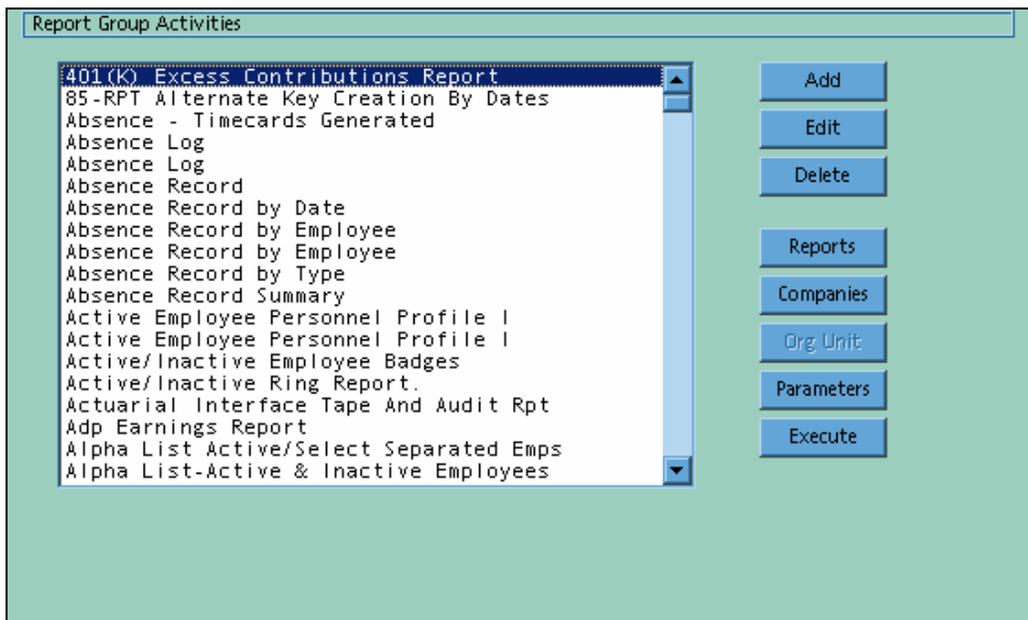
EGLF Values

Actions

SECTION 5 EXERCISE ANSWERS, continued



After the prompt form displays, schedule and run the report.



SECTION 5 EXERCISE ANSWERS, continued

Add Report Group

Report Group> SAMPLE

Title: MERIT INCREASE PROJECTION

Source Records: Labor/Hist/Emp Mastr

Report Group View

Public

Private

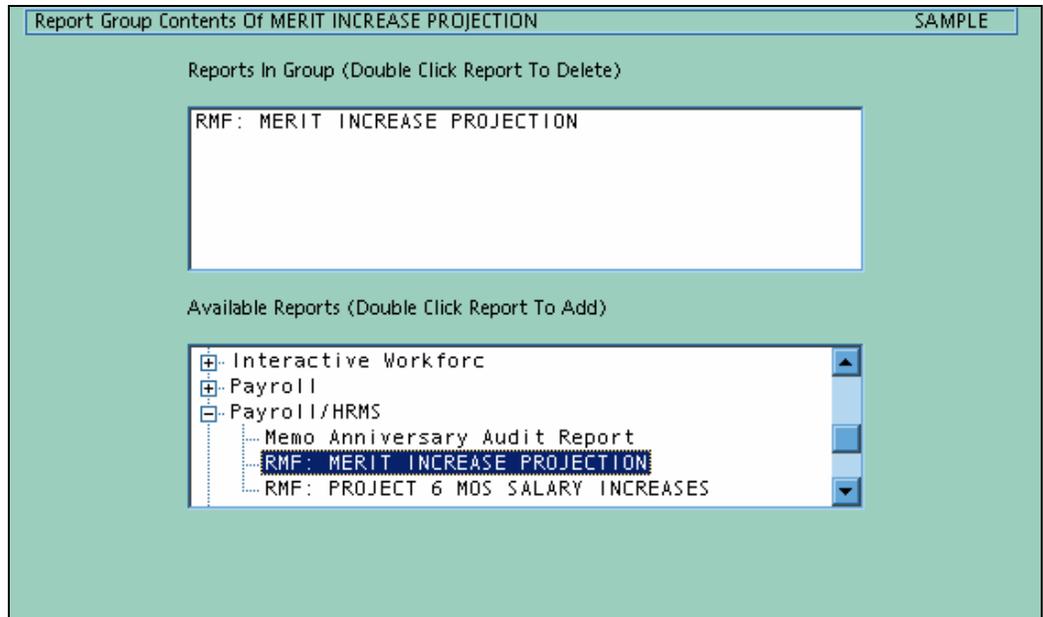
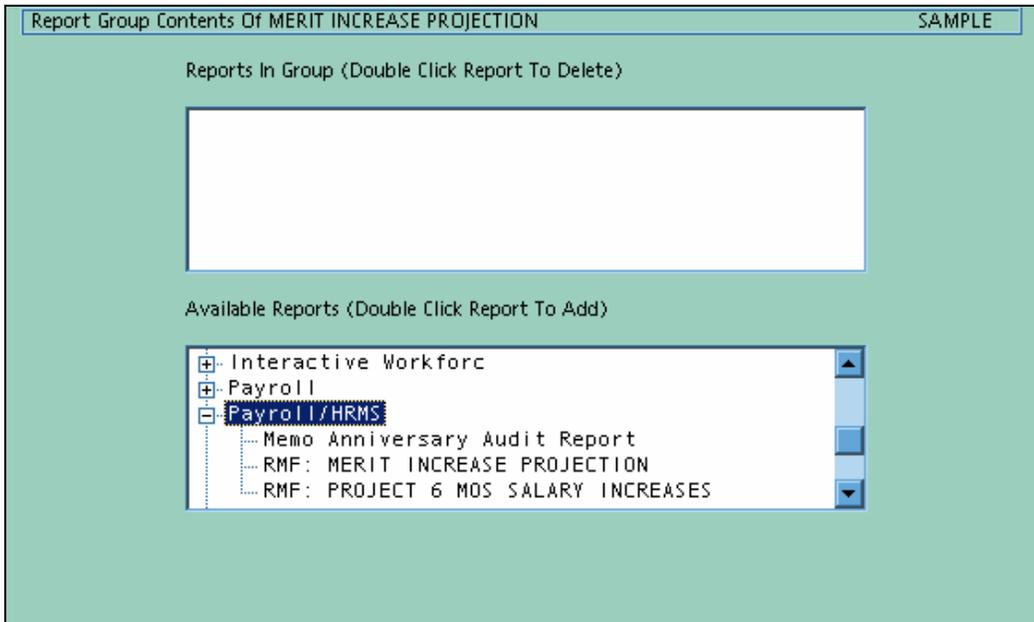
Report Group Contents Of MERIT INCREASE PROJECTION SAMPLE

Reports In Group (Double Click Report To Delete)

Available Reports (Double Click Report To Add)

- Benefits Admin
- Canadian Empl Equity
- EEO
- Employee Attendance
- Employee Comp
- Employee Development

SECTION 5 EXERCISE ANSWERS, continued



SECTION 5 EXERCISE ANSWERS, continued

```
SELECT OPTION ==> 1
                BATCH JOB INITIATION MENU
                1 Report
                2 Query
```

```
REPORT BATCH JOB INITIATOR
Enter Report Group name: SAMPLE
Hold output for online review? Y (Enter Y or N)
Normal, roll-up, or consolidate? (Enter space, R, or C)
```

**SECTION 5 EXERCISE ANSWERS**, continued

```
REPORT BATCH JOB INITIATOR  
JOB JRPTXXX  SUBMITTED
```

Use the Initiate Scheduled Reports program to run the report according to the instructions at your site or classroom setting and use the View Held Report program to preview the report.

```
ACTION ==> V D-Delete, P-Print, V-View ID ==> 011  
  
ID# PGS  DATE    TIME  
011 0004 09-19-2002 12:12:26
```

SECTION 5 EXERCISE ANSWERS, continued

SIDE ==> L-Left, R-Right PAGE ==> 0002 LINE ==> 01 Pages=0004

CORPORATION 99 ABC SOLUTIONS, INC. RET/TERM MERIT INCREASE PROJECTION  
 DIVISION 5555 RETIREE/TERM ORGANIZATION AS OF: 09-19-2002

EMPLOYEE-NAME	EMPLOYEE NUMBER	RATING VALUE	SALARY EFFECTIVE	SALAR DATE-SP
COSTELLO, SUSANNE	2006	2	02-01-1987	15-07-
HAYES, JOHNS A	1806	2	12-01-1986	15-09-
CREMMINS, ALAN E	2009	2	11-12-1986	15-10-
MANNING, WILLIAM Z	1165	2	10-01-1986	15-11-
LANNON, PATRICE	2007	2	09-01-1986	16-00-
HAMMER, JAMES B	1236	2	08-01-1986	16-01-
ADAMS, RICHARD	1117	2	06-01-1986	16-03-
ANDREWS, HENRY A	2013	2	03-09-1986	16-06-
WARD, CHESTERON	2012	2	03-01-1986	16-06-
PENDARVIS, MARTIN M	2010	2	01-03-1986	16-08-
COMPTON, SUSAN A	1116	2	01-01-1985	17-08-
ISLEY, JEANETTE J	1432	2	01-01-1985	17-08-

\*CONTROL-1-2 995555

SIDE ==> R-Left, R-Right PAGE ==> 0002 LINE ==> 01 Pages=0004

MERIT INCREASE PROJECTION REPT PAGE 1  
 AS OF: 09-19-2002 XSAL TIME 12:12 DATE 09-19-2002

G	SALARY EFFECTIVE	SALARY DATE-SPAN	POSITION IN-RANGE	ANNUAL SALARY	PROJ-MERIT INC-SAL
	02-01-1987	15-07-18	71.25	18,280.08	19,194.08
	12-01-1986	15-09-18	39.03	9,194.12	9,653.83
	11-12-1986	15-10-07	66.67	34,016.32	35,717.14
	10-01-1986	15-11-18	4.61	25,099.92	26,354.92
	09-01-1986	16-00-18	45.56	44,200.00	46,410.00
	08-01-1986	16-01-18	22.23	21,172.84	22,231.48
	06-01-1986	16-03-18	12.34	38,221.44	40,132.51
	03-09-1986	16-06-10	70.83	16,900.00	17,745.00
	03-01-1986	16-06-18	51.71	27,508.00	28,883.40
	01-03-1986	16-08-16	25.36	16,380.00	17,199.00
	01-01-1985	17-08-18	40.08	13,841.36	14,533.43
	01-01-1985	17-08-18	4.20	16,372.20	17,190.81
				281,186.28	295,245.60

SECTION 6 EXERCISE ANSWERS

Using the requirements worksheet, complete the Extract program specified by the memo and worksheet.

The screenshot shows the 'Solution View Tool Kit' dialog box. At the top right is an 'Option:' dropdown menu. Below it are two text input fields: 'Name of Program:' containing 'XBUDXT' and 'Title:' containing 'BUDGET REPORT DATA'. The main area is divided into two columns of radio button options. The left column, titled 'Program Type', includes: 'Entry Screen', 'New User Fields', 'Query', 'Report', 'PC Download', and 'Extract Routine' (which is selected). The right column, titled 'Action', includes: 'Add' (selected), 'Change', 'Delete', 'Inquiry', 'Program List', 'Program List By User', and 'Copy Program'. At the bottom, there is a 'Module:' dropdown menu set to 'Salary Administratio...' and a 'Security Code:' text input field. A footer note states: 'Solution View is a Registered Trademark of Cyborg Systems, Inc.'

The screenshot shows the 'Solution View Extract Writer: XBUDXT' dialog box. At the top right is an 'Option:' dropdown menu. Below it is an 'Extract Options' text input field. The main area is divided into two columns of radio button options. The left column, titled 'As Of Date Options', includes: 'Use Current Date' (selected), 'Use Entered Date', and 'Use Entered Range'. The right column, titled 'General Options', includes: 'Selection Criteria' (checked), '80 Character Record' (selected), and '150 Character Record'. At the bottom, there are two buttons: 'Composite View' and 'Single Field View'.

SECTION 6 EXERCISE ANSWERS, continued

Solution View Extract Field Selection: XBUDXT Option: ▼

	Seq Nbr	Field Name	Output Field	Calc Type	Result Length	Result Decimals
A	00	EMPLOYEE-NAME-10	0			
A	02	EMPLOYEE-NUMBER	0			
A	04	CTRL-THREE	0			
A	06	SALARY-INCREASE-DATE	0			
A	08	PLAN-PERCENT-CHANGE	0			
A	10	PLAN-ANNUAL-AMT-CHG	0			
A	12	PLAN-BUDGET-EFFECT-\$	0			
A	14	PLAN-BUDGET-EFFECT-%	0			

Go To Next Step

Display Values

Output Field   
  Calculation Types   
  Both

Solution View Extract Writer: XBUDXT Option: ▼

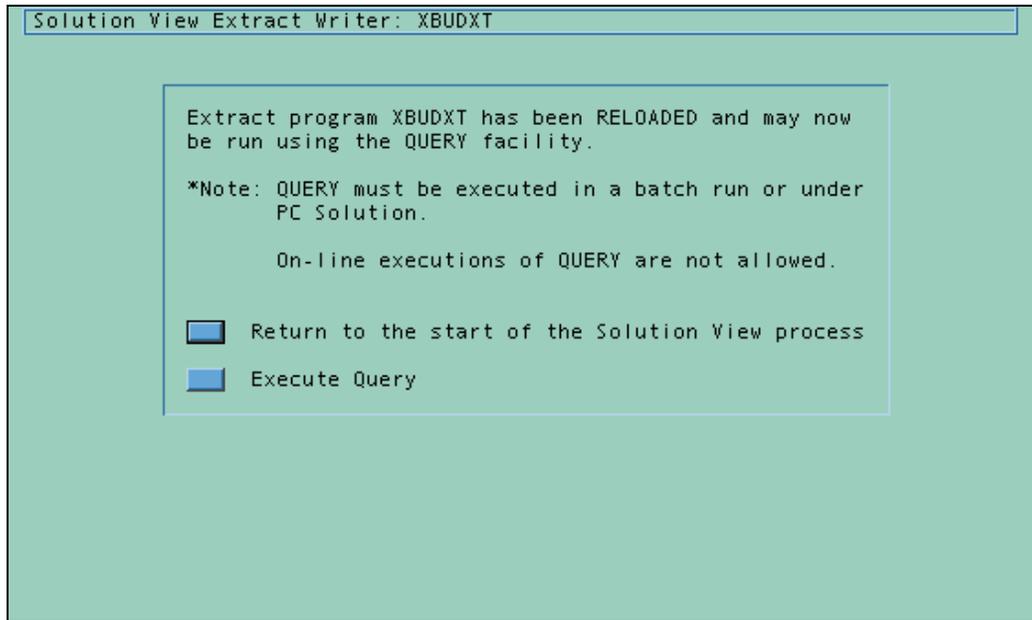
Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
SALARY-INCREASE-DATE	<input type="checkbox"/>	E	000000	E
	<input type="checkbox"/>			

Display Values

EGLF Values   
  Actions

SECTION 6 EXERCISE ANSWERS, continued



The following is an example of the raw output that would result from executing the extract program.

REYNOLDS, 2001	338809-01-87	8.00	1,491.68	497.28	2.67
BARNES, JO2002	338810-01-87	6.50	5,059.60	1,264.90	1.63
CMEYLA, JA2003	338803-01-87	8.00	1,067.38	889.30	6.67
KWONG, STE2004	338809-01-87	8.00	1,572.48	524.10	2.67
BROWN, WIL2005	338806-10-87	6.00	941.72	525.75	3.35
COSTELLO, 2006	338802-01-87	6.25	1,053.00	965.29	5.73
LANNON, PA2007	338809-10-87	8.50	3,756.92	1,158.28	2.62
HALL, RHON2008	338805-01-87	5.25	81,026.40	54,020.30	3.50
CREMMINS, 2009	338810-10-87	8.00	2,721.42	612.29	1.80
PENDARVIS, 2010	338808-20-87	7.50	1,228.76	447.05	2.73
WARD, CHES2012	333304-01-87	5.00	1,375.40	1,031.55	3.75
ANDREWS, H2013	338804-09-87	6.00	1,014.00	737.99	4.37
GRANT, KEI2014	303004-15-87	6.00	1,185.60	843.08	4.27
MARGOLIS, 2015	338804-01-87	6.00	2,430.00	1,822.50	4.50
SHEA, JEFF2016	303004-11-87	6.00	1,241.76	896.80	4.33

**SECTION 7 EXERCISE ANSWERS**

Using the requirements worksheet, complete the Entry Form program specified by the memo and worksheet. Use the new form to make an entry for employee 1001.

Solution View Tool Kit      Option:

Name of Program:     Title:

<b>Program Type</b> <input checked="" type="radio"/> Entry Screen <input type="radio"/> New User Fields <input type="radio"/> Query <input type="radio"/> Report <input type="radio"/> PC Download <input type="radio"/> Extract Routine	<b>Action</b> <input checked="" type="radio"/> Add <input type="radio"/> Change <input type="radio"/> Delete <input type="radio"/> Inquiry <input type="radio"/> Program List <input type="radio"/> Program List By User <input type="radio"/> Copy Program
--	--

Module:       Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View Screen Field Selection: XBASCR    Option:

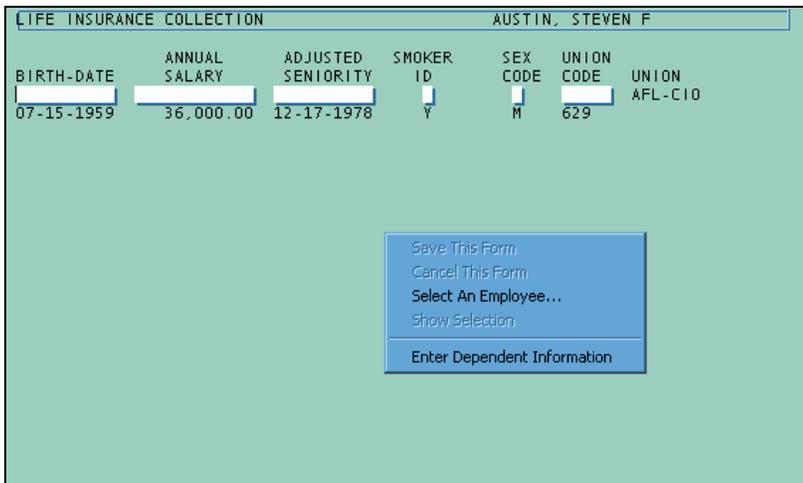
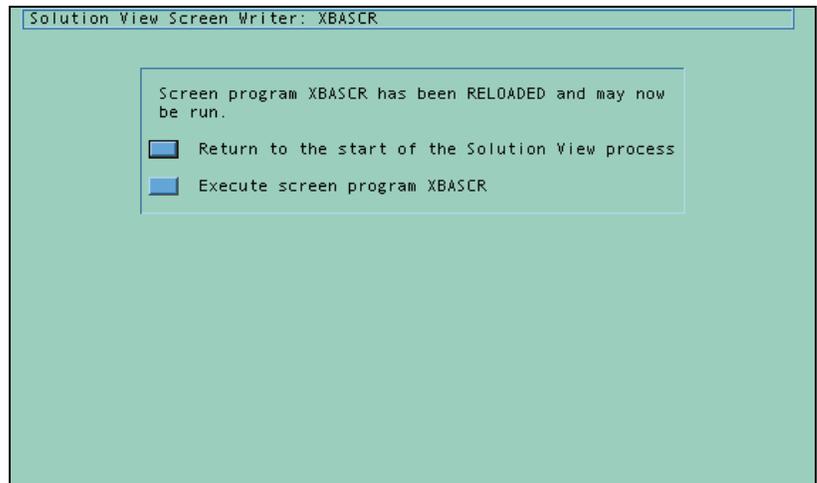
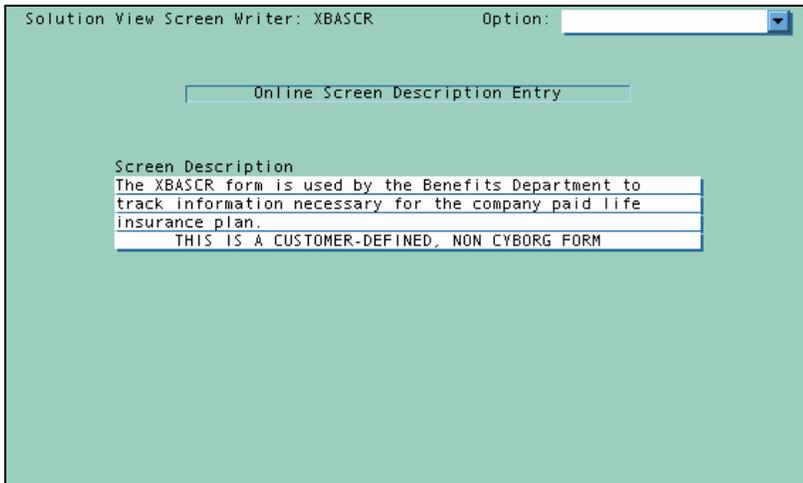
Field Names:	<input type="text" value="BIRTH-DATE"/>	<input type="text" value="ANNUAL-SALARY"/>
	<input type="text" value="ADJUSTED-SENIORITY"/>	<input type="text" value="SMOKER-ID"/>
	<input type="text" value="SEX-CODE"/>	<input type="text" value="UNION-CODE"/>
	<input type="text" value="UNION"/>	<input type="text"/>
	<input type="text"/>	<input type="text"/>

**Screen Prompts**

Use As Prompts   

Include Description

SECTION 7 EXERCISE ANSWERS, continued



**SECTION 8 EXERCISE ANSWERS**

Using the requirements worksheet, complete the New Fields Definition program specified by the memo and worksheet. Use the newly created form to make an entry for employee 1234.

Solution View Tool Kit Option:

Name of Program:  Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input checked="" type="radio"/> Add
<input checked="" type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

Solution View - New Fields Definition \*\*\*\*\* General Information \*\*\*\*\*

Screen Name	Screen Title	Security Code	Module Code	Employee Data
XABSCR	EMPLOYEE UNPAID ABSENCE	**	PP	Y

Screen Description

This form is used to record unpaid absences for an employee. This information may be used to determine absence trends, for example, the employee tends to be absent on the day just prior to a holiday.

Use As Prompts

Payroll must be notified so that the correct payment of earnings can be made.

SECTION 8 EXERCISE ANSWERS, continued

Solution View - New Fields Definition ***** New Field Name Entries *****				
New Field Name	Length	Type	Key	Codeset
XSTART-DATE		DT	K	
XEND-DATE		DT	G	
XABS-CODE	02		G	TA01
XABSENCE-TYPE		CD		TA01
XABSENT-HOURS	04	N2	G	
XDAY-OF-WEEK-CODE	03			TA02
XDAY-OF-WEEK		CD		TA02
XCALLED-IN	01		G	HR00
XAPPR-CODE	04		G	TA04
XAPPROVAL		CD		TA04
XPAYROLL-NOTIFIED	01		G	HR00

Type: - Alphanumeric  
 CD - Codeset Description  
 DT - Date  
 N0 - Numeric/0 Decimals  
 N2 - 2 Decimals  
 N4 - 4 Decimals  
 N6 - 6 Decimals  
 N\$ - Leading \$

Key: Enter K for each key field.  
 They must precede all non-key fields.  
 Enter G for all fields that are required entries.

EMPLOYEE UNPAID ABSENCE XABSCR

This form is used to record unpaid absences for an employee. This information may be used to determine absence trends, for example, the employee tends to be absent on the day just prior to a holiday.

Complete the text boxes below, then hit enter:

If needed, alter Org Level:

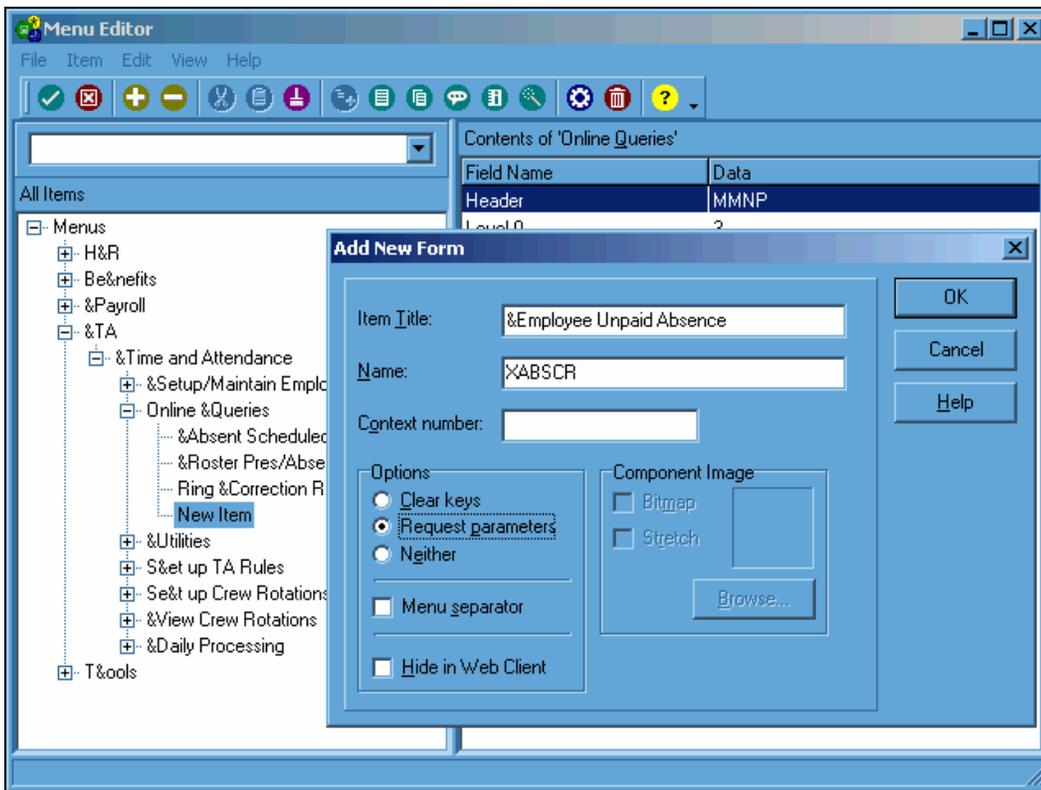
For inquiry only, enter 'I ':

Enter EMPL #:  or SSN:

or NAME:

Enter XSTART-DATE:

SECTION 8 EXERCISE ANSWERS, continued



EMPLOYEE UNPAID ABSENCE	AUSTIN, STEVEN F
Start Date>	<input type="text"/>
End Date:	<input type="text"/>
Abs Code:	<input type="text"/>
Absent Hours:	<input type="text"/>
Day Of Week Code:	<input type="text"/>
Called In:	<input type="text"/>
Appr Code:	<input type="text"/>
Payroll Notified:	<input type="text"/>

Payroll must be notified so that the correct payment of earnings can be made.

**Note:** Option list TA04 is not populated. Use the Option list Edit facility to add codes to that option list.

Proceed to use the new form after option list TA04 has been populated.

---

**NOTES**

---

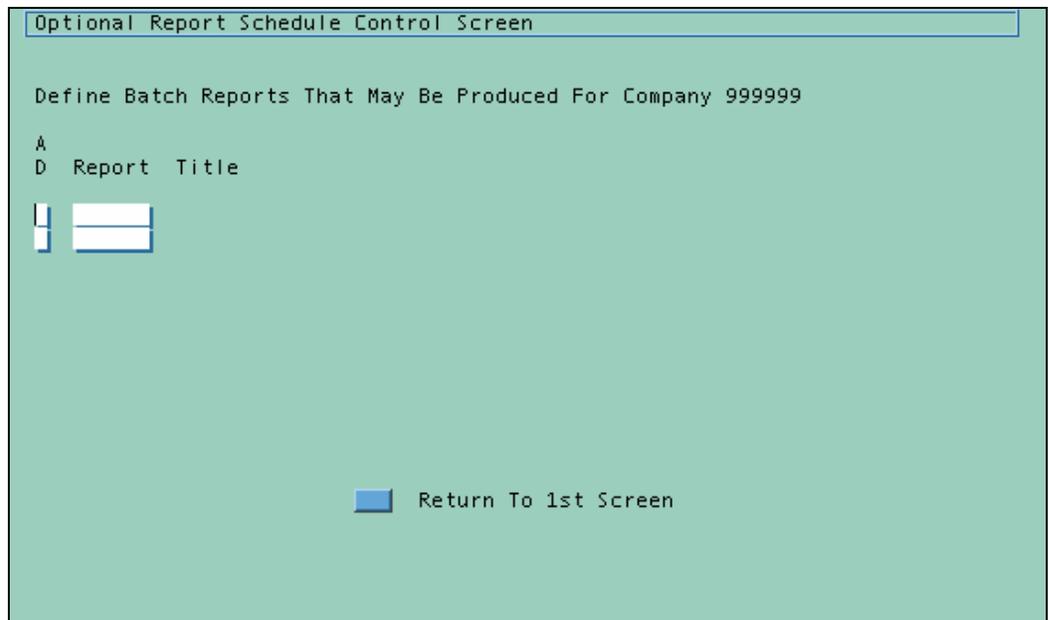
## APPENDIX B: EXTRA FOR EXPERTS

---

### TABLE OF CONTENTS

BATCH REPORTING VALIDATION.....	439
DISPLAYING A REPORT SCHEDULE.....	443
PURGING A REPORT SCHEDULE.....	445
PURGING AN ORGANIZATION SCHEDULE.....	447
CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION.....	449
THE SINGLE FIELD VIEW FORM.....	471
FIELD NAME CROSS REFERENCE FIELD HELP.....	473
MERGING/IMPORTING TEXT FILES.....	475
CONSOLIDATED AND ROLL-UP REPORTING.....	477
ENTERING HEADING INFORMATION.....	479
DESIGNATING THE CONSOLIDATED ORGANIZATION.....	481
USING ROLL-UP AND CONSOLIDATED REPORTING.....	483
CREATING A USER QUICK-REFERENCE FIELD MENU.....	484
ACCESSING A USER QUICK-REFERENCE FIELD MENU.....	489
DELETING ENTRIES FROM THE USER QUICK REFERENCE FIELD MENU.....	491
CHANGING OPERATOR TITLES FOR QUICK REFERENCE FIELD MENU.....	492

## Organization Report Validation



## BATCH REPORTING VALIDATION

### Organization Report Validation

The Report process assumes all reports are valid and are to be produced for each Organization, unless an **Optional Report Schedule Control** form is entered.

Using this optional form, you may specify which reports are valid for each Organization on your Employee Database. The form is set up by Organization.

When this form is present, the report process checks to be sure a report is valid before allowing it to be produced for the Organization.

Before accessing the form, verify that you are in the correct Organization for which you are establishing report validation.

### Accessing the Form

**Selection:**      **Step:**

**Navigator**

1.  Reporting Report Scheduling
-  Specify Reports for Organization

**Result:** The Optional Report Schedule Control form displays.

**Company**

2. Enter the company name (for example, 999999).



3. Click Save or press ENTER.

**Result:** The 2nd panel of the Optional Report Schedule Control form displays.

### Field Descriptions

The form contains three fields:

- **A/D**—specifies whether you are adding (A), or deleting (D) a line of data.
- **Report**—contains the name of all valid reports for the Organization displayed in the Command Line. Leaving this area blank indicates that all reports are valid for the Organization.
- **Title**—displays the title of the report after execution of the form.

## Organization Report Validation

Optional Report Schedule Control Screen

Define Batch Reports That May Be Produced For Company 999999

A  
D Report Title

A	1A-RPT
A	1G-RPT

Optional Report Schedule Control Screen

Define Batch Reports That May Be Produced For Company 999999

A  
D Report Title

<input type="checkbox"/>	1A-RPT	Alpha List-Active & Inactive Employees
<input type="checkbox"/>	1G-RPT	Birthday Listing
<input type="checkbox"/>		
<input type="checkbox"/>		

**BATCH REPORTING VALIDATION**, continued

**Completing the Form**

To establish Organization report validation:

**Selection:**

A/D

**Step:**

1. Type A or D indicating whether you are adding or deleting a report from this Organization.

**Report**

2. Type the name of the report that is valid for this Organization.

**Repeat:** Complete Steps 1 and 2 for each report that is valid to be run for this Organization.



3. Click Save or press ENTER.

**Note:** When including multiple reports for report validation, press ENTER and additional blank lines will appear.

## Displaying a Report Schedule

Display Utility

Object: Rpt Schedule

Object Key: WEEKLY

1st Line Only

```
PEWEEKLY          HPROJECT 6 MOS SALARY INCREASES
PEWEEKLYXPROPT   RMF: PROJECT 6 MOS SALARY
```

## DISPLAYING A REPORT SCHEDULE

### Viewing Existing Report Schedules

To view existing Report Schedules, use the Display Utility.

### Accessing the Display Utility

To access the Display Utility:

**Selection:**      **Step:**

**Navigator**



Development Tools

System Control Repository Utilities



List Control Repository Object

**Result:** The Display form displays.

### Field Descriptions

**Object**—the type of object to be edited, Report Schedule which has an object code of PE.

**Object Key**—the name of the specific object to be edited, for example, WEEKLY to designate a Report Schedule called WEEKLY.

### Viewing Existing Report Schedules

To view existing report schedules:

**Selection:**      **Step:**

**Object**

1. Choose a valid Object, (for example, Report Schedule).

**Object Key**

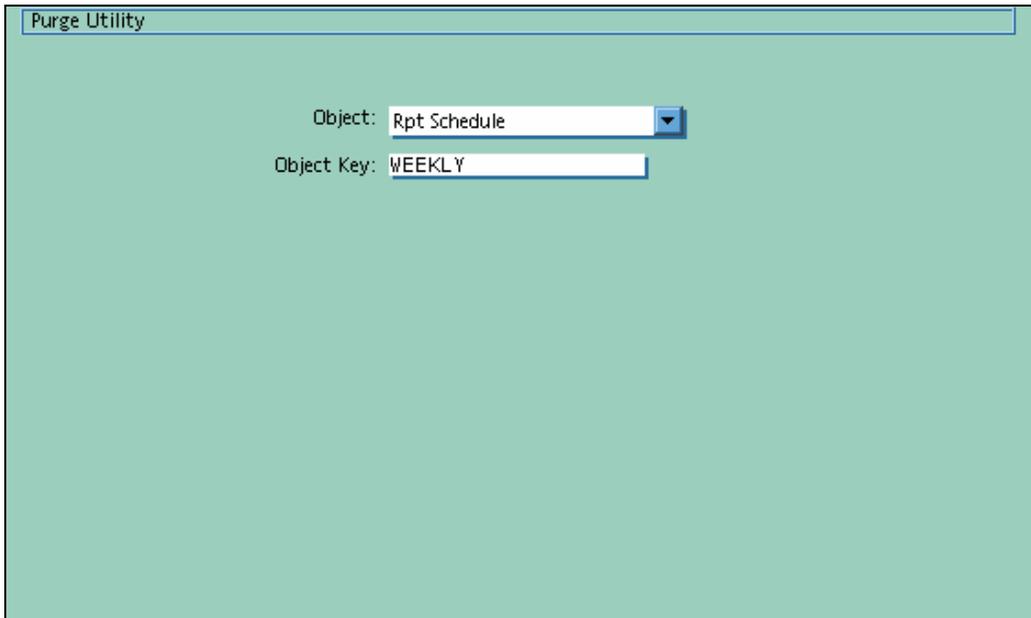
2. Type a valid Object Key, (for example, WEEKLY).



3. Click Save or press ENTER.

**Result:** The contents of the Report Schedule displays.

## Purging a Report Schedule

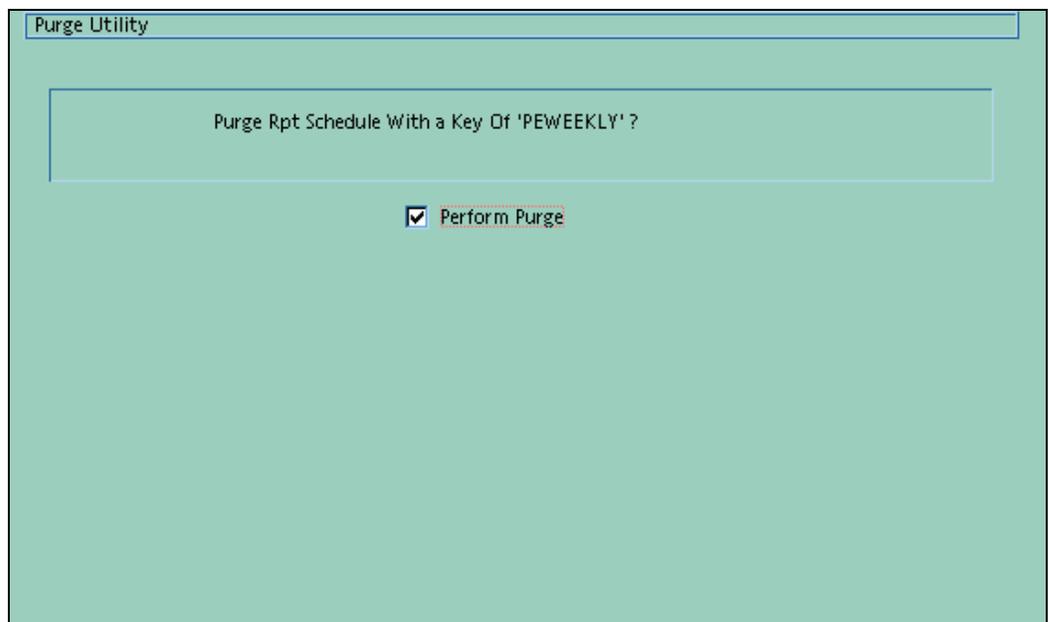


Purge Utility

Object: Rpt Schedule

Object Key: WEEKLY

This screenshot shows the 'Purge Utility' dialog box. It has a title bar that says 'Purge Utility'. Inside the dialog, there are two input fields. The first is labeled 'Object:' and contains the text 'Rpt Schedule'. The second is labeled 'Object Key:' and contains the text 'WEEKLY'.



Purge Utility

Purge Rpt Schedule With a Key Of 'PEWEEKLY' ?

Perform Purge

This screenshot shows the 'Purge Utility' dialog box after the user has entered the object and key. A large rectangular box in the center contains the text 'Purge Rpt Schedule With a Key Of 'PEWEEKLY' ?'. Below this box, there is a checkbox labeled 'Perform Purge' which is checked.

## PURGING A REPORT SCHEDULE

### Purging Existing Report Schedules

To purge existing Report Schedules, use the PURGE program.

### Accessing the Purge Form

**Selection:**      **Step:**

**Navigator**



Development Tools

System Control Repository Utilities



Purge

**Result:** The PURGE Form displays.

### Field Descriptions

**Object**—the type of object to be purged, for example, Report Schedule which has an object code of PE.

**Object Key**—the name of the object to be purged, for example, WEEKLY.

### Purging An Existing Report Schedule

**Selection:**      **Step:**

**Object**

1. Choose a valid Object, (for example, Report Schedule).

**Object Key**

2. Type an existing Object Key, (for example, WEEKLY).



3. Click Save or press ENTER.

**Result:** The message “Purge Rpt Schedule Key Of ‘PEWEEKLY’ ?” displays.

**Perform Purge**

4. Select Perform Purge check box to purge.

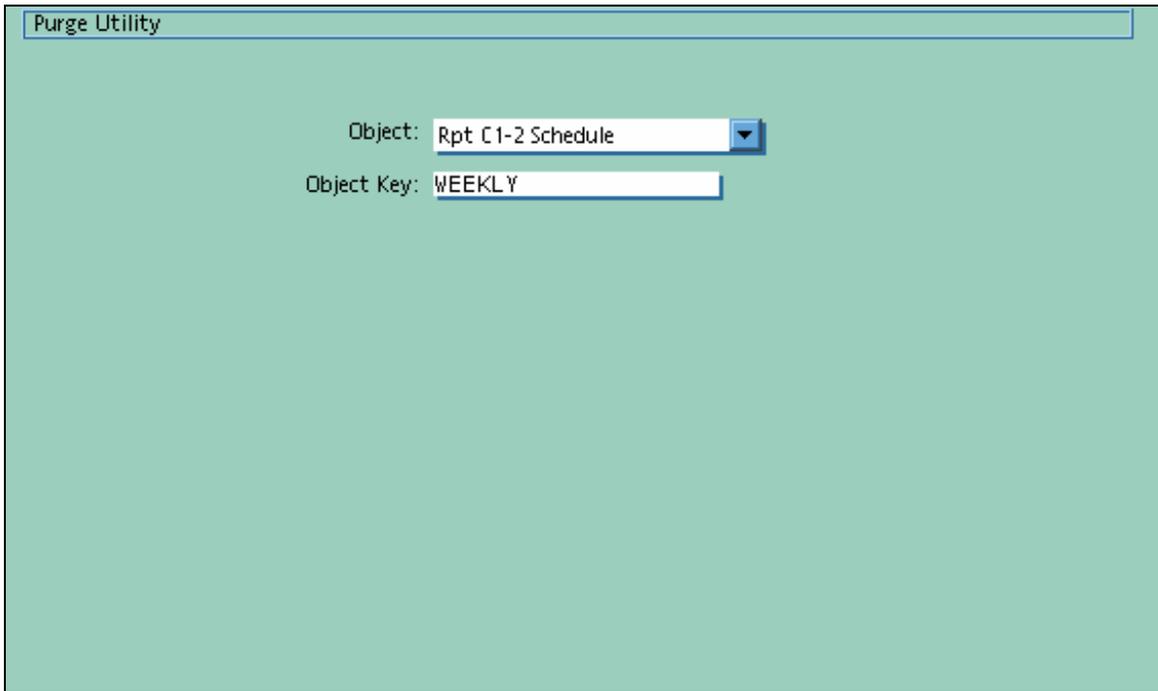


5. Click Save or press ENTER.

**Result:** The message “---COMPLETE---” displays indicating that the purge has been completed.

**Note:** Report Schedules may also be purged by using the Report Scheduler form to delete all reports listed for that report schedule.

## Purging an Organization Schedule



The screenshot shows a dialog box titled "Purge Utility" with a light green background. It contains two input fields: "Object:" with a dropdown menu showing "Rpt C1-2 Schedule" and "Object Key:" with a text box containing "WEEKLY".

**PURGING AN ORGANIZATION SCHEDULE**

**Purging Existing Schedule C1-2s** To purge existing Schedule C1-2s For Reports forms, use the Purge form.

**Accessing the Purge Form**

**Selection:**                      **Step:**

**Navigator**



Development Tools

System Control Repository Utilities



Purge

**Field Descriptions**

**Object**—the type of object to be purged, for example, Rpt C1-2 Schedule which has an object code of PD.

**Object Key**—the name of the object to be purged, for example, WEEKLY.

**Completing the Form**

**Selection:**                      **Step:**

**Object**

**1.** Type a valid Object, (for example, Rpt C1-2 Schedule).

**Object Key**

**2.** Type an existing Object Key, (for example, WEEKLY).



**3.** Click Save or press ENTER.

**Result:** The message “Purge Rpt C1-2 Schedule With A Key Of ‘PDWEEKLY’ ?” displays.

**Perform Purge**

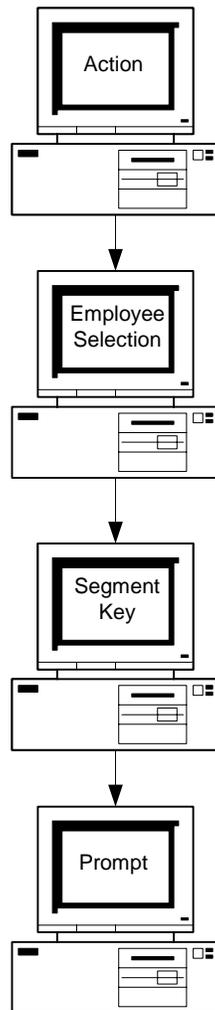
**4.** Select Perform Purge check box to purge.



**5.** Click Save or press ENTER.

**Result:** The message “---COMPLETE---” displays indicating that the purge has been done.

## PC Selection Writer Form Series



## CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION

### PC Selection Forms

The PC Selection Download Facility is used in conjunction with the PC Application File Data Selection form to gather a specific group of employees and their data for a PC Application File. The facility creates a small program that supplies additional employee selection criteria to the PC Application File Data Selection information used to gather employee/company data.

The facility cannot be used alone, and if used, **MUST** be entered prior to entering the PC Application File Data Selection form.

These are the forms that are available in Solution View, that are available to gather a specific group of employees and their data for a PC Application File:

- **Action form**—used to select the type of report to be written and the action to be performed on that report
- **Employee Selection Criteria form**—used to create selection logic
- **Stacked Segment Key form**—optional, used to select one or more segment keys for a multiple-occurrence segment where the key is not a date
- **Prompt form**—informs you that your program has been written and compiled successfully

Now that you have previewed the form series that you will use to create additional employee selection criteria, let us look at each task in detail and see what tools are provided on each form and how to use them.

## Action Form

Solution View Tool Kit Option:

Name of Program:  Title:

Program Type	Action
<input type="radio"/> Entry Screen	<input checked="" type="radio"/> Add
<input type="radio"/> New User Fields	<input type="radio"/> Change
<input type="radio"/> Query	<input type="radio"/> Delete
<input type="radio"/> Report	<input type="radio"/> Inquiry
<input checked="" type="radio"/> PC Download	<input type="radio"/> Program List
<input type="radio"/> Extract Routine	<input type="radio"/> Program List By User
	<input type="radio"/> Copy Program

Module:  Security Code:

Solution View is a Registered Trademark of Cyborg Systems, Inc.

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**ACTION Form** The first PC Selection Writer form, the Action form, is used to identify the Solution View program.

**Accessing the Form**

**Selection:**      **Step:**

**Navigator**



User Tools  
User Tools



Solution View

**Result:** The Action form displays.

**Field Descriptions**

The Action form field definitions have already been presented in Section 3. This section shows you how to use those fields to access the PC Selection Writer series of forms.

**Using the Action Form**

To create a new program using the PC Selection Writer:

**Selection:**      **Step:**

**Name of Program**

**1.** Type the name of the program. Extract names must begin with **X**.

**Title**

**2.** Type a title for the PC Select program (optional).

**Program Type**

**3.** Select PC Download.

**Action**

**4.** Select Add.

**Module**

**5.** Select the module ID to be associated with the PC Select program (optional).

**Security Code**

**6.** Type the Security Code for the PC Select program (optional).



**7.** Click Save or press ENTER.

**Result:** If the Action form is error-free, the Employee Selection Criteria form displays.

## The Employee Selection Criteria Form

Solution View PC Download Writer: XSAMPX      Option: ▼

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="text"/>	<input type="text"/>

Display Values

EGLF Values

Actions

## CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION, continued

**Employee Selection Criteria Form**      The Employee Selection Criteria form is used to narrow the number of employees that will be included in the PC Select. This is done by setting a condition on one or more fields.

**Field Descriptions**      The Employee Selection Criteria form contains the following fields:

- **Field Name**—data field name(s) used as the basis for employee selection. Key fields may **NOT** be selected here. The fields on this form do not have to be displayed on the report.

To access the Field Name Table in a menu-driven format, select Field Selection Menu from the Option Field *or* type a ? in the Field Name field to access the Field Name Table Menu to select the data field name(s). For each ? you will be allowed to select an entry from the Field Name Menu.

Once you are viewing the menu selections, select your field names by typing X next to each one.

After selecting your data field name entries from the Field Name Menu, select Leave Menu from the Option field to return to the Employee Selection Criteria form.

- **Not**—to create selection criteria that checks for a negative condition, in other words, a condition that is not true.

For example, if you want only employees who do NOT have an annual salary greater than \$28,000 to appear on the report, you would indicate “Not” in this field (and the appropriate EGLF, Literal Compare Value, and Action fields, discussed next).

## **Employee Selection Criteria Form Examples of Ranges and OR Conditions and Found/Not Found Logic**

### ■ Performance ratings of 1 or 2 or 3

Field Name      Not EGLF   Literal Compare Value  
RATING-VALUE      E   1;2;3

### ■ Birth dates ranging from 01-01-42 through 01-01-62

Field Name      Not EGLF   Literal Compare Value  
BIRTH-DATE      E   01-01-42/01-01-62

### ■ Only Pay Frequencies 1 through 3, or equal to 6

Field Name      Not EGLF   Literal Compare Value  
PAY-FREQUENCY      E   1/3;6

### ■ Salary Segment for January 1, 1993, Reclassification Increase

Field Name      Not EGLF   Literal Compare Value  
ANNUAL-AMOUNT-CHANGE      F   930101;9;I40

### ■ A value of spaces is valid. Example:

Field Name      Not EGLF   Literal Compare Value  
SHIFT-CODE      E   SPACES

---

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Field Descriptions,** ■ **EGLF** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are as follows:

- E or = (equal)
- G or > (greater than)
- L or < (less than)
- F (found)

The F value is used to incorporate FOUND or NOT FOUND logic for segment selection. This logic allows you to include or exclude employees based on the presence of a specific segment on their record.

For the F value, the field name can be **ANY** field from the segment. To select a specific occurrence is desired, enter the key field value(s) in the Literal Compare Value field.

- **Literal Compare Value**—values used with the corresponding field name entry on this form to specify the selection criteria. The value in each field must be appropriate for the data field in the Field Name field.
- **For EGLF field entries of E, G, and L**, numeric values may be entered with a decimal, commas, or dollar signs, if desired. Dates may be entered in the format YYMMDD or MM-DD-YY. Use century dates when a field displays on the system that way.

Slashes (/) or semi-colons (;) may be entered in the Literal Compare Value field as delimiters. A slash indicates a range and a semi-colon indicates an AND/OR condition. **When using ranges or the OR condition, the EGLF field must be an E (or =).**

A literal value of SPACES is valid here. This can be used to check a field for blanks.

- **For Found/Not Found logic**, one or more key field values must be entered in the proper order, in other words, 1st key field value, 2nd key field value, and so forth. The key field values must be separated by a semi-colon (;).

Only one set of key field values is allowed per Found/Not Found entry. If the Literal Compare Value field is left blank, then the first occurrence of the segment is used.

## Employee Selection Criteria and the Codeset Selection Menu

Solution View PC Download Writer: XSAMPX    Option:

Employee Selection Criteria

Field Name	Not	EGLF	Literal	Compare Value	Action
SEX-CODE	<input type="checkbox"/>	<input checked="" type="checkbox"/> E			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			
	<input type="checkbox"/>	<input type="checkbox"/>			

Display Values

- EGLF Values
- Actions

Solution View PC Download Writer: XSAMPX    Option:

Codeset Selection Menu For Sex Codes  
For Criteria Field: SEX-CODE

Female                       Male  
 Unclassified

-----Complete-----     Return To Selection Criteria

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Using the Codeset Selection Menu**

The Codeset Selection Menu can be used with the Employee Selection Criteria form when entering data into the Literal Compare Value field for fields that are tied to option lists.

If the field name entered in the Field Name field is tied to an option list, and the Literal Compare Value field is left blank, the process will automatically bring the user to the Codeset Selection Menu.

Once there, the selections are made based on the option list description, (for example, Male), and not the code itself, (for example, M).

**Number of Selections**

As many selections can be made that will fit on one line on the Employee Selections Criteria form. If the line following the criteria entry is also blank, in other words, no field name in the Field Name entry field, then up to two lines of selections can be made.

The length of an entry is the length of the code (field length), plus one. The system will automatically place a semi-colon (;) between each selection made. If two lines of selections are made, the system will also place a C (Continue) into the first line's Action Code field.

**Exiting the Codeset Selection Menu**

To leave the Codeset Selection Menu at any time, use the Leave Menu option in the Option field.

The Codeset Selection Menu will automatically return to the Employee Selection Criteria form when the last entry is processed.

## Employee Selection Criteria Form Action Code Examples

**E** example: Employees who have an ACTIVITY-CODE of 001 **ARE NOT** included on the report.

Field Name	Not	EGLF	Literal	Compare	Value	Action
ACTIVITY-CODE	E		001			E

**P** example: Employees who have a SEX-CODE of M (males) **ARE** included on the report.

Field Name	Not	EGLF	Literal	Compare	Value	Action
SEX-CODE	E	M				P

**N and P** example: Employees who have a JOB-CODE of 12550 **AND** a JOB-CODE-EXTENT of 0002 **ARE** included on the report. Satisfying only one of these conditions does not include the employee on the report.

Field Name	Not	EGLF	Literal	Compare	Value	Action
JOB-CODE	E	12550				N
JOB-CODE-EXTENT	E	0002				P

**P and P** example: Employees who have a JOB-CODE of 12550 **OR** a SALARY-GRADE of S10 **ARE** included on the report. Satisfying either one of these conditions includes the employee on the report.

Field Name	Not	EGLF	Literal	Compare	Value	Action
JOB-CODE	E	12550				P
SALARY-GRADE	E	S10				P

**C and P** example: Employees who have a CONTROL-3-CODE equal to one of the values on line 1 or line 2 are included in this report.

Field Name	Not	EGLF	Literal	Compare	Value	Action
CONTROL-3-CODE	E	1255;1926;2183;3065;3067;3402;3555 3505;4721;6800				C P

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Field Descriptions,**     ■     **Action** (option list SC22)—used to determine whether to process the description record being compared. Choose the Actions button to display the options.

- Enter **C** to indicate that the values for comparison continue on the next line; field names and the NOT and EGL indicators need not be repeated on subsequent lines which contain an action of C.
- Enter **N** to indicate that other comparisons need to be checked.
- Enter **P** to select the record based on this comparison.
- Enter **E** to exclude a record based on this comparison.

The entries have the following effect in the program:

- **C**     the values for this statement continue on the next line
- **N**     if this statement is true, check the next comparison
- **P**     if this statement is true, include the employee in the selection process and then check the next line for another comparison
- **E**     if this statement is true, do not include the employee in the selection process

## The Employee Selection Criteria Form Conditional Statement Examples

**Include employees having a SALARY of \$20,000.00 through \$25,000.00**

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	P

**Exclude employees having a Job Code of 12550 OR 16775**

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	12550;16775	E

**Include only Female employees having a Job Code of 13552 through 13559**

Field Name	Not	EGLF	Literal Compare Value	Action
JOB-CODE		E	13552/13559	N
SEX-CODE		E	F	P

**Exclude employees with a Salary of \$20,000.00 through \$25,000.00 OR hired before January 1, 1992**

Field Name	Not	EGLF	Literal Compare Value	Action
ANNUAL-SALARY		E	20000.00/25000.00	E
HIRE-DATE		L	01-01-92	E

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Conditional Statements**      The types of conditional logic that can be performed with the Employee Selection Criteria form are as follows:

Conditional Statements are used to perform an operation dependent on some condition. The conditions that may be used in a conditional statement are:

- |                |                    |
|----------------|--------------------|
| ■ Equal to     | ■ Not Equal to     |
| ■ Greater than | ■ Not Greater than |
| ■ Less than    | ■ Not Less than    |

**Compound Conditional Conditions**      Compound Conditional Statements are used to test for several conditions. Compound conditional statements include:

- **AND** Tests to see if all of several conditions exist. This means that more than one condition must be met by the employee in order to appear on the report.
  - An entry of N in the Action field designates an AND condition.
  
- **OR** Tests to see if one of several conditions exists. This means that at least one of the conditions must be met by the employee in order to appear on the report.
  - An entry of P in the Action field designates an OR condition when you are testing the values between one or more FIELD NAMES.
  - A semi-colon (;) inserted between a list of two or more Literal Compare Value designates an OR condition when testing the Field Name entry.
  - A slash (/) inserted between two Literal Compare Values designates an OR condition resulting in a range test for a specific Field Name entry. The range will include the starting value and the ending value.

## The Employee Selection Criteria Form

Solution View PC Download Writer: XSAMPX      Option: ▼

Employee Selection Criteria

Field Name	Not	EGLF	Literal Compare Value	Action
SALARY-GRADE	<input type="checkbox"/>	E	S10;T15	N
ANNUAL-SALARY	<input type="checkbox"/>	E	10,000.00/59999.99	N
CONTROL-3-CODE	<input type="checkbox"/>	E	02	P
	<input type="checkbox"/>			

Display Values

EGLF Values  
 Actions

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Using the Employee Selection Criteria Form**

The procedure for using the Employee Selection Criteria form is:

- | <u>Selection:</u>   | <u>Step:</u>  |
|---|---|
| <b>Field Name</b>   | <b>1.</b> Type the name of the field to be used in the conditional statement.   |
| <b>Not</b>  | <b>2.</b> Indicate Not to designate the negative effect of the conditional argument entered in the EGLF field (optional).   |
| <b>EGLF</b>   | <b>3.</b> Type E (Equal), G (Greater), L (Less), or F (Found) to establish a conditional comparison.  |
| <b>Literal Compare Value</b>  | <b>4.</b> Type the literal value to be compared with the corresponding Field Name entry.  |
| <b>Action</b>   | <b>5.</b> Type C (continue values on next line), N (check next comparison), P (include in process), or E (exclude from the process) to designate the action to be taken if the conditional statement is true. |
|   | <b>6.</b> Repeat steps 1 through 5 for each selection criteria condition that exists.   |
|  | <b>7.</b> Click Save or press ENTER.  |

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process in the Option field, and execute the form to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the next form in the series displays.

## The Stacked Segment Key Form

Solution View PC Download Key Values: XSAMPX Option: [Dropdown]

The following SELECTION CRITERIA field is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence is desired enter the appropriate key information below.

Bypass this screen

Requested Selection Criteria Field: CONTROL-3-CODE

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>
	<input type="checkbox"/>	<input type="checkbox"/>		<input type="checkbox"/>

EGL Values

Equal To: E or =  
Greater Than: G or >  
Less Than: L or <

Actions

N = Next: Check next key field  
P = Proceed: No further checks performed

## CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION, continued

### Stacked Segment Key Form

The Stacked Segment Key form is only displayed when a field name entered on the Employee Selection Criteria form is in a multiple occurrence (stacked) segment whose primary KEY field is not a date. This is the only time this form is used. The form is displayed automatically for each segment (not field) you listed on the Employee Selection Criteria form; you do not need to request it.

As with Query and Report programs, Solution View assumes you want to display the requested field's data from the first occurrence in the stacked segment. If another occurrence is desired, the criteria must be entered on this form. You may bypass this form by pressing ENTER, or clicking Bypass this form. (This action assumes the default, the first occurrence.)

### Field Descriptions

The Stacked Segment Key form contains the following fields:

- **Requested Detail Line Field**—an inquiry-only field that displays a field name from the Employee Selection Criteria form if that field occurs in a multiple-occurrence segment whose key is not a date.

Only the first requested field of a multiple occurrence segment is displayed for the selection of the segment key(s). All other fields from the same segment use the same comparison logic.

- **Required Keys**—displays the key field name related to the fields selected on the Employee Selection Criteria form. The key field name(s) used as the basis for selecting occurrences are placed here automatically by Solution View. **Do not change this field.**
- **Not**—used to indicate that the selection criteria checks for a condition that does NOT exist. Check the box, or enter N in this field to create selection criteria that checks for a condition that does NOT exist. The default entry of blank is interpreted in the positive, in other words, *is*.

## The Stacked Segment Key Form

Literal Compare  
Examples

- **To specify HED 001** (Use data only from HED 001)

**Required Keys Are** Not EGL Literal Compare Values  
**HED-NUMBER** E 001

- **To specify HEDs from 001 to 500** (Use data from HED 001 through 500)

**Required Keys Are** Not EGL Literal Compare Values  
**HED-NUMBER** E 001/500

- **To specify HED 500 or HED 510** (Use data from HED 500 and 510 only)

**Required Keys Are** Not EGL Literal Compare Values  
**HED-NUMBER** E 500;510

- **To specify HEDs from 001 to 500, or HED 510** (Use data from HED 100 through 500 and also use data from HED 510)

**Required Keys Are** Not EGL Literal Compare Values  
**HED-NUMBER** E 001/500;510

**CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION**, continued

**Field Descriptions,**  
continued

- **EGL** (option list SC21)—used to designate the condition to be met by the value in the next field (Literal Compare Value). Valid entries are:
  - **E** or = (equal)
  - **G** or > (greater than)
  - **L** or < (less than)
- **Literal Compare Value**—used to specify a literal value to be compared to the Required Keys Are field.

Ranges can be entered using slashes (/) to indicate an inclusive range, or semi-colons (;) to indicate an OR range condition. **The EGL field must be an E (=) to use ranges.**

**Note:** A key field may not contain spaces. Therefore, a literal entry of SPACES in the Literal Compare Value field is invalid.

- **Action** (option list SC22)—allows you to specify whether to process the segment occurrence being compared.

P in the field causes a record that meets the condition to be retained.

N in the field causes a record that meets the condition to be retained if it also meets the next statement(s).

## The Stacked Segment Key Form

Solution View PC Download Key Values: XSAMPX Option: ▼

The following SELECTION CRITERIA field is in a stacked segment whose primary KEY field is NOT a date. If other than the first occurrence is desired enter the appropriate key information below.

Bypass this screen

Requested Selection Criteria Field: CONTROL-3-CODE

Required Keys	Not	EGL	Literal Compare Value	Action
LOCATION-NUMBER	<input type="checkbox"/>	E	01	P
	<input type="checkbox"/>			
	<input type="checkbox"/>			
	<input type="checkbox"/>			

**EGL Values**

Equal To: E or =  
 Greater Than: G or >  
 Less Than: L or <

**Actions**

N = Next: Check next key field  
 P = Proceed: No further checks performed

Solution View PC Download Writer: XSAMPX

Selection Routine XSAMPX has been RELOADED and may now be included in the PS-SEG process.

Return to the start of the Solution View process

Execute PS-SEG process

## CREATING A PROGRAM TO PERFORM SELECTION CRITERIA ON EXTRACT DATA FOR PC SOLUTION, continued

### Using the Stacked Segment Key Form

The procedure for using the Stacked Segment Key form to select a segment occurrence is:

To select the **first occurrence** of the stacked segment, click Bypass this form or press ENTER.

To select any other occurrence, perform the following steps:

<u>Selection:</u>	<u>Step:</u>
<b>Required Keys</b>	1. Do not change this field, since it is automatically updated by the Report Writer.
<b>Not</b>	2. Indicate N to designate the negative effect of the entry in the EGL field (optional).
<b>EGL</b>	3. Type E (Equal), G (Greater), L (Less), or the symbolic equivalents, =, >, or <, to establish a conditional comparison.
<b>Literal Compare Value</b>	4. Type the literal value to be compared with the corresponding Required Keys field.
<b>Action</b>	5. Type P (include in process), or N (check next comparison) to designate the selection of the specific occurrence.
	6. Click Save or press ENTER.
	7. Repeat steps 1 through 6 for each segment key requested.

Errors on this form must be resolved before proceeding. You may either fix the error, or choose the Cancel button to return to pre-error condition.

To escape an errored form, select Restart Process (X) in the Option field, and press ENTER to return to the Action form. Then change or delete the program.

**Result:** When this form is error-free, the prompt form displays. The Prompt form informs you that your PC Select program has been written and compiled successfully. English Language Code now exists for the PC Select program.

## The Single Field View Form

Solution View Extract Writer: XSAMXT      Option:

Extract Options

As Of Date Options

- Use Current Date
- Use Entered Date
- Use Entered Range

General Options

- Selection Criteria
- 80 Character Record
- 150 Character Record

Solution View Extract Field Selection: XSAMXT Option:

Add  
 Change  
 Delete  
 Position

Seq Nbr:       Field Name:

Output Field

- None
- Print (Edit)
- Output (Unedited)

Calculation Types

- None
- Date
- Numeric
- Time Span

Length:       Decimals:

## THE SINGLE FIELD VIEW FORM

**Single Field View Form** The Single Field View form is used to enter a single entry (1 line) of the Composite View Field Selection form for either a report or extract routine.

All fields for that one line of entry from the Composite View form display in a true GUI format rather than option list-related entry fields as is the case with the Composite View form.

**Form Usage** After you have typed a single entry on this form, click Composite View at the bottom of the form to return to the Composite View form. Click Single Field View on the Composite View form to return to the Single Field View form to make additional field entries.

These pushbuttons provide you with an easy and fast method to display the “big picture” (Composite View) and return to the focus of entering one field at a time.

**Note:** The Single Field View pushbutton appears on the Composite View form only when you have chosen the Single Field View from the previous Option form.

The Process/Next Step pushbutton is used to indicate that no other entries will occur on the Single Field View form. (For standard form processing, click Save.) When the Process/Next Step pushbutton is used, the process then turns control over to the editing routines of the Composite View form. If any edit errors occur at this time in the process, the Composite View form displays, but not the Single Field View form.

**Note:** Once the Process/Next Step pushbutton has been used on the Single Field View form (turning control over to the Composite View program), the pushbutton to return to the Single Field View form no longer appears on the Composite View form.

## Field Name Cross Reference Field Help

Solution View Extract Writer: XSAMXT      Option: ▼

Selected Menu: Basic Employee Data      Page: 1  
 Segment Type: E    Segment Code: E

E-SEGMENT	SIN-1	SIN-1-3
SOCIAL-INSURANCE-NBR	SOCIAL-SEC-NBR	SOCIAL-SECURITY-NBR
SSN-1-3	SIN-2	SIN-3
SIN-4	SSN-4	SIN-5
SIN-5-7	SSN-5-6	SIN-6
SIN-7	SSN-7	SIN-8
SSN-8-11	SIN-9	SIN-9-11
SIN-10	SIN-11	SSN-12
PAY-FREQUENCY	PAY-FREQUENCY-CODE	RSX-PAY-FREQUENCY
PAYMENT-CODE	<b>PAYMENT-TYPE</b>	PAYMENT-TYPE-15
STATUS	STATUS-CODE	RSX-SEX
SEX	SEX-CODE	EEO-RACE
LANGUAGE-PREFER	LANGUAGE-PREFER-DESC	LANGUAGE-PREFERENCE
RACE-CODE	RSX-EEO-RACE	ET-APPLICANT-SOURCE
RSX-UNION	UNION	UNION-CODE

Solution View Extract Writer: XSAMXT      Option: ▼

Selected Menu: Basic Employee Data  
 Segment Type: E    Segment Code: E  
 Requested Field: PAYMENT-TYPE

This alphanumeric field contains 20 characters.  
 It is a value found in Code Set PP40.

----Complete----

## FIELD NAME CROSS REFERENCE FIELD HELP

### **Field Name Cross Reference Field Help**

The Field Name Table Cross Reference menu is invoked from the Solution View process either by the Field Selection Menu option from the Option field in the upper right corner of the form, *or* by typing a question mark (?) in the first position of the Field Name entry field.

Once there, you can display field help documentation by typing a question mark (?) into the selection box preceding the specific field. When this is done, the system documentation for the selected field is displayed.

You may display field documentation for more than one field at a time by typing a question mark (?) into two or more fields simultaneously.

### **Returning to the Field Selection Form**

The field help form will automatically return to the field selection form once all the information for the field(s) has been displayed.

To return to the field selection form before the end of the documentation is reached, select Leave Fld Menu Help from the Option field in the upper right hand corner of the form.

---

**NOTES**

---

## MERGING/IMPORTING TEXT FILES

Refer to your specific software program's manuals if you plan to merge extracted data from Solution View. Each program has specific requirements for the merge procedure as well as delimiter requirements.

### Delimiter Requirements

Delimiters are used to recognize the beginning and end of each field and record. For example, by default, WordPerfect assumes that fields are separated by commas and that each record ends with a carriage return.

Quattro Pro, for example, can import a file that used commas and quotes to separate text in rows. The delimiters are used to establish columns in the spread sheet page. A comma-and-quote delimited file has the following characteristics:

- Data is entered in lines, similar to the rows of a spreadsheet
- Types of data are separated on each line with commas

Quattro Pro can also import a file delimited by commas only. Instead of requiring quotes around labels, Quattro Pro determines whether to enter the data as a label or a value depending on its first character.

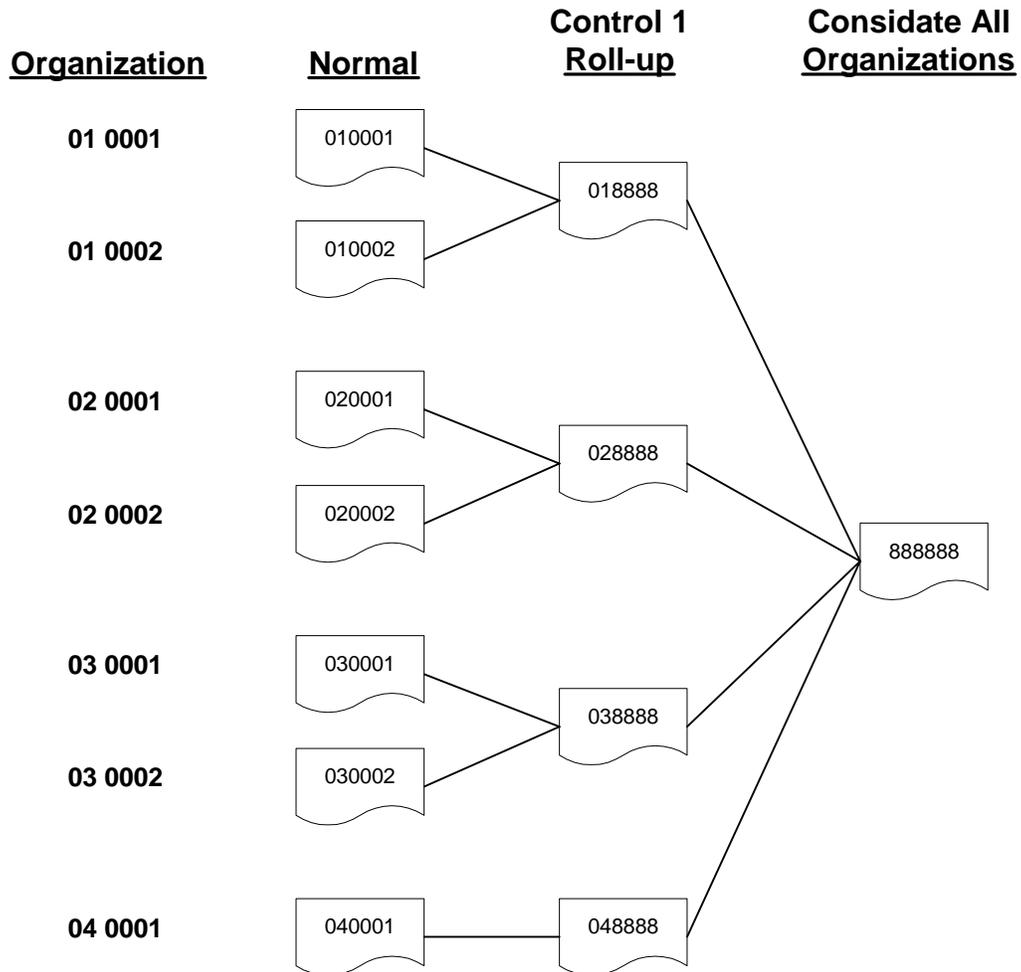
### Performing the Merge

After the file has been properly marked for delimiters, you can merge it into a primary file, such as a monthly newsletter to all employees.

Your software manuals contain specific instructions and tips for the merge process. For example, WordPerfect recommends that the first time you do a merge, you add just a few records to the secondary file, create the primary file, and perform a merge. This "reduced" merge will give you an opportunity to verify that the merge is correct without wasting a lot of time and paper.

## Control-1 Roll-up/Consolidate All Organizations Reporting

**Example:**            7 Organizations  
                          1 Report (1A-RPT)



## CONSOLIDATED AND ROLL-UP REPORTING

### Overview

- Roll-up Reporting
  - Roll-ups and reports all like Control 1 values on the Employee Database (or those specified on the Report Schedule form)
- Consolidated Reporting
  - Consolidates all Organizations on the Employee Database (or those specified on the Report Schedule form) as one report
- This enhancement is usable with any *Solution Series* report, either delivered or created using Solution View
- The feature is fully supported by the SUBMIT and VIEW programs, as well as the usual batch report process

## Entering Headingc Information

Set Up A New Organization

New Org:

Enter a new valid Organization

Note: An entry here will display the Organization Name And Address form to continue the setup of information about the new org.

Organization Name And Address

Organization Name

Org:

OL2:

Organization Address

Address:

City/State:

ZIP Code:

Control Level Descriptions

Org:

OL2:

PL3:

PL4:

PL5:

PL6:

Organization Options

Common Tax Org:

Report Frequency:

Country:

Local Currency:

No Pay Warning:

Routing Number:

Org Category:

Mag Stripe/Bar Cde:

Clear Then Adjust

Frick Tape

Tax MC Override

Reciprocal Rules

Months Retained:

Hist:  Labor:

Build Alt Key

Phonetic  Emple ID

## ENTERING HEADING INFORMATION

Establish the following forms for each Roll-up or Consolidated Organization. In the previous example, 018888, 028888, 038888, 048888, and 888888 would be set up.

**Set Up A New Organization Form (AA-NEW)** ■

Enter the Organization value. (A unique Organization that will be used for reporting purposes only.)

**Organization Name And Address Form (AA-SCR)** ■

Enter the desired report heading information.

**Organization Options Form (AF-SCR)** ■

Set up this company for Consolidated Reporting. This applies to both the Roll-up and Consolidated Organizations. This company:

- cannot have employees added or transferred to it.
- will never have pay calculated in P4CALC.
- does not have to be added to RPT20.

## System Options

System Options

User Code Life Span: 00-03-00

Overall Max Lines: 60

RTPRNT Max Lines: 54

Control 1-2

Sign-on Default: 999999

Consolidate To: 888888

Other Options

Production Version

Reject On ? HELP  
When No Entry Made

Alternate Language

GUI Client

Distributed Admin

Node ID:

Active

Main Document Path

C:\Cyb50-idx\Docs

## DESIGNATING THE CONSOLIDATED ORGANIZATION

### System Options Form (SCOPTS)

- The Consolidate To field is used to overlay the Organization (if using Consolidated Reporting) or the Control 2 field (if using Roll-up reporting) on the report extract records.

This form is accessible only by the Security Officer.

## Using Roll-up And Consolidated Reporting

```
REPORT BATCH JOB INITIATOR

Enter Report Group name: WEEKLY
Hold output for online review? N (Enter Y or N)
Normal, roll-up, or consolidate? C (Enter space, R, or C)
```

### Sample Batch Control Record

```
1...5...0...5...0...5...0...5...0...5...0...5...0...5
P REPORTJ00100          REPORT WEEKLY          F001          C
                        ▲
                        ▲
```

## USING ROLL-UP AND CONSOLIDATED REPORTING

### Initiating Reporting

- Initiate Consolidated or Roll-up reporting using one of the two methods:

- **Report Batch Job Indicator**

Indicate your choice of Roll-up (R) or Consolidate (C).

- **Batch Control Record**

Indicate your choice of Roll-up (R) or Consolidate (C) in Column 52.

### Indicating the Organizations

- Indicate which Organizations to process:

- In addition to your production Organizations, you must also enter your Consolidated or Roll-up Organizations on the Report Schedule form.



**CREATING A USER QUICK-REFERENCE FIELD MENU**, continued

3. Type USER or User in the Menu Type field. Press ENTER. (To create a menu for a Module instead of a specific user, type a Module Code in the Menu Type field instead of USER when using this procedure.)

The screenshot shows a dialog box titled "Create User Quick Reference Menu". Inside, there is a text field labeled "Menu Type:" containing the text "USER". Below this, there is a larger text area with the following instructions: "Enter 'User' or 'USER' to establish a quick reference menu for your operator ID." followed by "or" and "Enter two characters to establish a quick reference menu by module ID." At the bottom of the dialog, there is a button labeled "Return To Field Menu".

4. The Main Menu displays. Select a menu option and press ENTER.

The screenshot shows a dialog box titled "Field Menu" with "Page: 1" in the top right corner. Inside, there is a section titled "Main Menu" containing a list of radio button options: "Clear Selection", "Company Fields", "Table Fields", "Other Record Fields", "Employee Fields", "Time Entry/Adj Flds", and "Work Fields". The "Employee Fields" option is selected and highlighted with a red box. At the bottom right of the dialog, there is a button labeled "Exit Quick Menu".

CREATING A USER QUICK-REFERENCE FIELD MENU, continued

- The first page of the selected field name menu displays. Select a menu option again. Press ENTER.

Field Menu Page: 1

Selected Menu: Employee Fields

- Previous Menu
- Employee Identifier
- Basic Employee Data
- Employee Transfer
- Name and Address
- Applicant Job Data
- Employee HED's
- Basic Applicant Data
- Employee Pay Data
- Applicant Name+Addr
- W2 Name and Address
- Pay Allocations
- Labor/Hist HED Amts

- The selected field names display with entry boxes to the left of each field. Type X to the left of each field you want to have on your User Quick Reference Menu. You may select up to 45 data fields for your menu from any of the field menus. Click Save or press ENTER.

Field Menu Page: 1

Selected Menu: Basic Employee Data  
Segment Type: E Segment Code: E

- Main Menu
- Previous Menu
- Clear Option

<input type="checkbox"/> E-SEGMENT	29PPE	003000	<input type="checkbox"/> SIN-1	29PPE	003001
<input type="checkbox"/> SIN-1-3	29PPE	003002	<input type="checkbox"/> SOCIAL-INSURANCE-NBR	29PPE	003003
<input checked="" type="checkbox"/> SOCIAL-SEC-NBR	29PPE	003004	<input checked="" type="checkbox"/> SOCIAL-SECURITY-NBR	29PPE	003005
<input type="checkbox"/> SSN-1-3	29PPE	003006	<input type="checkbox"/> SIN-2	29PPE	004000
<input type="checkbox"/> SIN-3	29PPE	005000	<input type="checkbox"/> SIN-4	29PPE	006000
<input type="checkbox"/> SSN-4	29PPE	006001	<input type="checkbox"/> SIN-5	29PPE	007000
<input type="checkbox"/> SIN-5-7	29PPE	007001	<input type="checkbox"/> SSN-5-6	29PPE	007002
<input type="checkbox"/> SIN-6	29PPE	008000	<input type="checkbox"/> SIN-7	29PPE	009000
<input type="checkbox"/> SSN-7	29PPE	009001	<input type="checkbox"/> SIN-8	29PPE	010000
<input type="checkbox"/> SSN-8-11	29PPE	010001	<input type="checkbox"/> SIN-9	29PPE	011000
<input type="checkbox"/> SIN-9-11	29PPE	011001	<input type="checkbox"/> SIN-10	29PPE	012000
<input type="checkbox"/> SIN-11	29PPE	013000	<input type="checkbox"/> SSN-12	29PPE	014000
<input checked="" type="checkbox"/> PAY-FREQUENCY	29PPE	015000	<input checked="" type="checkbox"/> PAY-FREQUENCY-CODE	29PPE	015001

CREATING A USER QUICK-REFERENCE FIELD MENU, continued

- Continue selecting field names from the second form of field name options. Click Save or press ENTER after you have selected fields from this option form.

Field Menu

Selected Menu: Basic Employee Data  
Segment Type: E Segment Code: E

Main Menu Page: 1  
 Previous Menu  Clear Option

E-SEGMENT	29PPE	003000	SIN-1	29PPE	003001
SIN-1-3	29PPE	003002	SOCIAL-INSURANCE-NBR	29PPE	003003
X SOCIAL-SEC-NBR	29PPE	003004	X SOCIAL-SECURITY-NBR	29PPE	003005
SSN-1-3	29PPE	003006	SIN-2	29PPE	004000
SIN-3	29PPE	005000	SIN-4	29PPE	006000
SSN-4	29PPE	006001	SIN-5	29PPE	007000
SIN-5-7	29PPE	007001	SSN-5-6	29PPE	007002
SIN-6	29PPE	008000	SIN-7	29PPE	009000
SSN-7	29PPE	009001	SIN-8	29PPE	010000
SSN-8-11	29PPE	010001	SIN-9	29PPE	011000
SIN-9-11	29PPE	011001	SIN-10	29PPE	012000
SIN-11	29PPE	013000	SSN-12	29PPE	014000
X PAY-FREQUENCY	29PPE	015000	X PAY-FREQUENCY-CODE	29PPE	015001

Field Menu

Selected Menu: Basic Employee Data  
Segment Type: E Segment Code: E

Main Menu Page: 2  
 Previous Menu  Clear Option

RSX-PAY-FREQUENCY	29PPE	015002	PAYMENT-CODE	29PPE	016000
PAYMENT-TYPE	29PPE	016001	PAYMENT-TYPE-15	29PPE	016002
STATUS	29PPE	017000	STATUS-CODE	29PPE	017001
RSX-SEX	29PPE	019000	SEX	29PPE	019001
X SEX-CODE	29PPE	019002	EEO-RACE	29PPE	020000
LANGUAGE-PREFER	29PPE	020001	LANGUAGE-PREFER-DESC	29PPE	020002
LANGUAGE-PREFERENCE	29PPE	020003	RACE-CODE	29PPE	020004
RSX-EEO-RACE	29PPE	020005	ET-APPLICANT-SOURCE	29PPE	022000
RSX-UNION	29PPE	022001	UNION	29PPE	022002
UNION-CODE	29PPE	022003	ET-AGENCY-ID	29PPE	025000
WORKERS-COMP-CODE	29PPE	027000	X BIRTH-DATE	29PPE	032000
BIRTH-DATE-OLD	29PPE	032001	BIRTH-YEAR	29PPE	032002
BIRTH-MONTH	29PPE	035000	BIRTH-MTH-D	29PPE	035001

- Make your selections on the last form of field name options and click Save or press ENTER.

Field Menu

Selected Menu: Basic Employee Data  
Segment Type: E Segment Code: E

Main Menu Page: 2  
 Previous Menu  Clear Option

COMMISSION-FLAG	29PPE	061000	COMMISSION-FLAG-DESC	29PPE	061001
ET-CURRENT-STATUS	29PPE	061002	FAIR-LABOR-CODE	29PPE	061003
FAIR-LABOR-CODE-DESC	29PPE	061004	PAY-PDS-TO-STOP-PAY	29PPE	062000
PERIOD-OVERRIDE	29PPE	062001	PERIODS-TO-STOP-PAY	29PPE	062002
EMPLOYEE-SECURITY	29PPE	064000	KEY-FIELD-1	29PPE	066000
KEY-FIELDS	29PPE	066001	KEY-FIELD-2	29PPE	067000
KEY-FIELD-3	29PPE	068000	KEY-FIELD-4	29PPE	069000
CLEAR-ANNIV	29PPE	070000			

----Complete----

CREATING A USER QUICK-REFERENCE FIELD MENU, continued

- If you wish to select more field names from other menus, continue by selecting another menu and pressing ENTER.

Field Menu Page: 1

Selected Menu: Employee Fields

Previous Menu  
 Employee Identifier  
 Basic Employee Data  
 Employee Transfer  
 Name and Address  
 Applicant Job Data  
 Employee HED's

Basic Applicant Data  
 Employee Pay Data  
 Applicant Name+Addr  
 W2 Name and Address  
 Pay Allocations  
 Labor/Hist HED Amts

- Continue 'Xing' field names from additional menus until you have all the field names selected for the User Quick Reference Menu. Click Save or press ENTER.

Field Menu Page: 1

Selected Menu: Name and Address  
Segment Type: F

Main Menu  
 Previous Menu  
 Clear Option

<input type="checkbox"/>	F-SEGMENT	30PP	000000	<input type="checkbox"/>	F-SEGMENT-TYPE	30PP	000001
<input type="checkbox"/>	F-SEGMENT-TYPE/CODE	30PP	000002	<input type="checkbox"/>	F-SEGMENT-KEY-DATA	30PP	001000
<input type="checkbox"/>	NAME-CODE	30PP	001001	<input type="checkbox"/>	EMPLOYEE-NAME	30PP	004000
<input type="checkbox"/>	EMPLOYEE-NAME-10	30PP	004001	<input type="checkbox"/>	EMPLOYEE-NAME-15	30PP	004002
<input type="checkbox"/>	EMPLOYEE-NAME-20	30PP	004003	<input type="checkbox"/>	EMPLOYEE-NAME-25	30PP	004004
<input type="checkbox"/>	EMPLOYEE-NAME-27	30PP	004005	<input type="checkbox"/>	EMPLOYEE-NAME-5	30PP	004006
<input type="checkbox"/>	NAME	30PP	004007	<input type="checkbox"/>	NAME-29	30PP	004008
<input type="checkbox"/>	NAME-FIRST-15	30PP	004009	<input type="checkbox"/>	NAME-OF-EMP	30PP	004010
<input checked="" type="checkbox"/>	ADDRESS	30PP	034000	<input type="checkbox"/>	ADDRESS-LINE-ONE	30PP	034001
<input checked="" type="checkbox"/>	ADDRESS-2	30PP	064000	<input type="checkbox"/>	ADDRESS-LINE-TWO	30PP	064001
<input type="checkbox"/>	CITY	30PP	094000	<input type="checkbox"/>	CITY-AND-STATE	30PP	094001
<input type="checkbox"/>	CITY/PROVINCE	30PP	094002	<input checked="" type="checkbox"/>	CITY/STATE	30PP	094003
<input checked="" type="checkbox"/>	STATE	30PP	113000	<input type="checkbox"/>	ZIP-CODE+4	30PP	115000

## ACCESSING A USER QUICK-REFERENCE FIELD MENU

To access your User Quick Reference Menu, select Field Selection Menu in the option field on the Field Selection form in Solution View. You may also use one or more question marks (?) in the Field Name field on the form. Press ENTER.

Solution View Query Field Selection: XTST2    Option: **Field Selection Menu**

Field Name:     Total Option:

From the Field Name Table Main Menu, select User Quick Reference. Press ENTER.

Solution View Query Writer: XTST2    Option:

Field Name Table Main Menu    Page: 1

Clear Selection

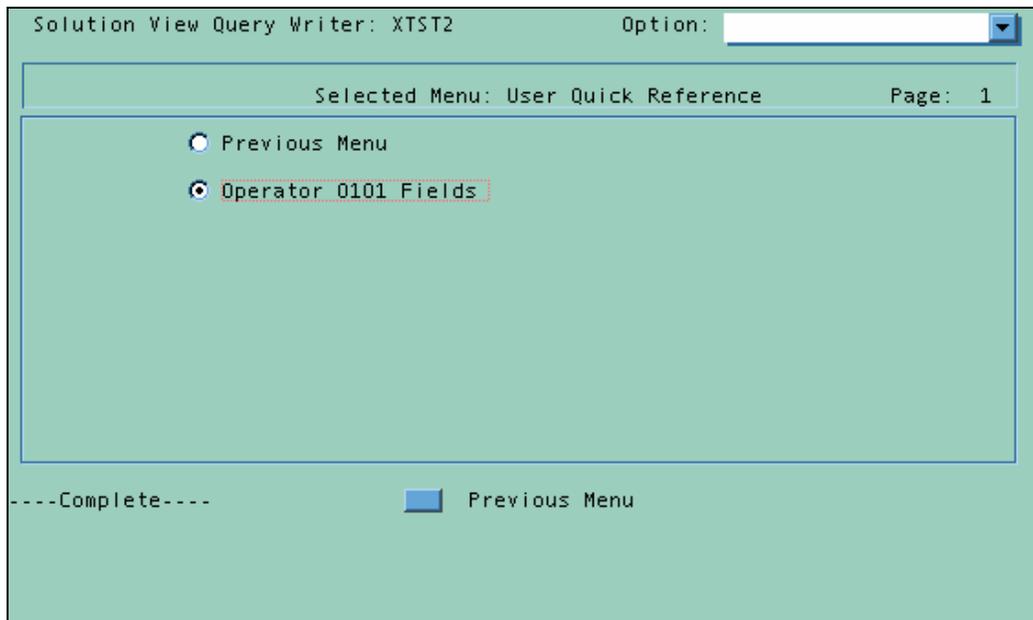
Company Fields       Employee Fields

Table Fields             Time Entry/Adj Flds

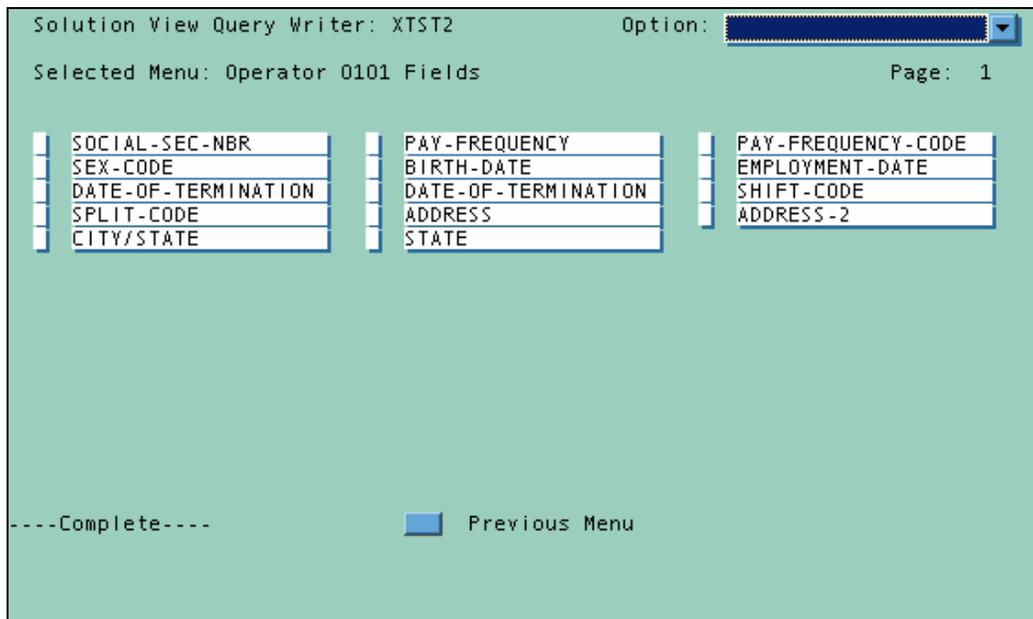
Other Record Fields     Work Fields

**User Quick Reference**     Quick Reference Flds

ACCESSING A USER QUICK-REFERENCE FIELD MENU, continued



Press ENTER. A Menu containing the available User Quick Reference specific menus displays. Type an entry in the Menu Option that corresponds to the option that displays your Operator ID. Press ENTER. This selects the specific menu which you created.



Type X to the left of one or more field names from your User Quick Reference Menu to use in your Solution View program. Press ENTER. The selected fields are now included in your Solution View program.

## DELETING ENTRIES FROM THE USER QUICK REFERENCE FIELD MENU

Access the EDIT Utility, select Field Name Tbl Xref as the Object, and type 99 in the Object Key field.

The screenshot shows a window titled "Edit Utility" with a light green background. It contains two input fields: "Object:" with a dropdown menu showing "Field Name Tbl Xref" and "Object Key:" with a text box containing "99".

Type D in the Action field of the record you wish to delete and click Save or press ENTER.

COMMAND:

PtMcSg/TbDspTie	Field Name	Mc	Cdset	KTDLLL	RDBMS
99QR0101 000000	SOCIAL-SEC-NBR	PP		J00012	Y
99QR0101 000002	PAY-FREQUENCY	PP	PP29	000015	N
99QR0101 000003	PAY-FREQUENCY-CODE	PP	PP29	00001	Y
99QR0101 000004	SEX-CODE	PP	PP41	00001	Y
D 99QR0101 000005	BIRTH-DATE	PP		82006	Y
99QR0101 000006	EMPLOYMENT-DATE	PP		82006	Y
99QR0101 000007	DATE-OF-TERMINATION	PP		82006	N
99QR0101 000015	DATE-OF-TERMINATION	PP		82006	N
99QR0101 000016	SHIFT-CODE	PP	PP43	00001	Y
99QR0101 000017	SPLIT-CODE	PP	PP44	00001	Y
D 99QR0101 000018	ADDRESS	PP		00030	N
99QR0101 000019	ADDRESS-2	PP		00030	N
D 99QR0101 000020	CITY/STATE	PP		00025	N
99QR0101 000021	STATE	PP		00002	N

## CHANGING OPERATOR TITLES FOR QUICK REFERENCE FIELD MENU

Edit Utility

Object:

Object Key:

COMMAND:

Group	Title	L	RDBMS Table Name
1699QR0101	Operator 0101 Fields	T	
2021PPAA	Company Name	T	COMPANY
2021PPAB	Company Address	T	COMPANY_ADDRESS
2021PPAC	Company City/State	T	COMPANY_CITY_STATE
2021PPAD	Company Ctrl Levels	T	COMPANY_CTRL_LEVEL
2021PPAE	Company Run Options	T	COMPANY_RUN_OPTS
2136ATOF	Appl Contact Data	T	APPLICANT
2136ATOG	Appl Reference Data	T	APPLCNT_REFERENCE
2136ATOH	Appl Reference Addr	T	APPLCNT_REF_ADDR
2136ATOI	Appl Job Applied For	T	JOB_APPLIED_FOR
2136ATOJ	Appl Preferred Work	T	WORK_PREFERENCES
2136ATZ5	Appl Interview Data	T	APPL_INTERVIEW
2136ATZ6	Appl Job History	T	PRIOR_EMPLOYMENT
2136ATZ9	Appl New Hire Data	T	APPL_PRE_TRANSFER
2136BA0A	Beneficiary Name	T	BENEFICIARY
2136BA0B	Beneficiary Address	T	BENEFICIARY_ADDR
2136BA0C	Beneficiary City, ST	T	BENEFICIARY_CITY
2136BA0D	Covered Dependents	T	COVERED_DEPENDENTS
2136BA0E	Dependents to COBRA	T	

COMMAND:

Group	Title	L	RDBMS Table Name
C 1699QR0101	MY OPERATOR FIELDS	T	
2021PPAA	Company Name	T	COMPANY
2021PPAB	Company Address	T	COMPANY_ADDRESS
2021PPAC	Company City/State	T	COMPANY_CITY_STATE
2021PPAD	Company Ctrl Levels	T	COMPANY_CTRL_LEVEL
2021PPAE	Company Run Options	T	COMPANY_RUN_OPTS
2136ATOF	Appl Contact Data	T	APPLICANT
2136ATOG	Appl Reference Data	T	APPLCNT_REFERENCE
2136ATOH	Appl Reference Addr	T	APPLCNT_REF_ADDR
2136ATOI	Appl Job Applied For	T	JOB_APPLIED_FOR
2136ATOJ	Appl Preferred Work	T	WORK_PREFERENCES
2136ATZ5	Appl Interview Data	T	APPL_INTERVIEW
2136ATZ6	Appl Job History	T	PRIOR_EMPLOYMENT
2136ATZ9	Appl New Hire Data	T	APPL_PRE_TRANSFER
2136BA0A	Beneficiary Name	T	BENEFICIARY
2136BA0B	Beneficiary Address	T	BENEFICIARY_ADDR
2136BA0C	Beneficiary City, ST	T	BENEFICIARY_CITY
2136BA0D	Covered Dependents	T	COVERED_DEPENDENTS
2136BA0E	Dependents to COBRA	T	

## APPENDIX C: WORKSHEETS

---

### TABLE OF CONTENTS

QUERY WRITER .....	494
REPORT WRITER.....	495
EXTRACT WRITER.....	496
FORM WRITER.....	497
NEW FIELDS DEFINITION WRITER.....	498

QUERY WRITER

QUERY WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X		<b>Module Specific?</b> Y / N	
<b>Program Title:</b>		<b>Security?</b> Y / N	
<b>As of Date:</b>		<b>Total Employee Count?</b> Y / N	
<b>Employee Selection Criteria?</b> Y / N		<b>Display Sequence:</b> <input type="checkbox"/> Employee Number <input type="checkbox"/> Other	
<b>Sequence</b>	<b>Field Name</b>	<b>Segment Occurrence</b>	<b>Print</b>
<b>If Employee Selection Criteria:</b>		<b>Compare Value:</b>	<b>Actions: (C, P, N, E)</b>
<b>Field Name:</b>	(not) E G L F		<input type="checkbox"/>
<b>Field Name:</b>	(not) E G L F		<input type="checkbox"/>
<b>Field Name:</b>	(not) E G L F		<input type="checkbox"/>
<b>Field Name:</b>	(not) E G L F		<input type="checkbox"/>



**EXTRACT WRITER**

EXTRACT WRITER

Date: \_\_\_\_\_

Program Name: X _____ X T		Module Specific? Y / N								
Program Title:		Security? Y / N								
As of Date:		Extract Length 80 / 150								
Employee Selection Criteria? Y / N										
Sequence	Field Name	Segment Occurrence	Output Option			Calculation Type			Result Length	Result Decimals
			No Edit/Edit/No Output			Numeric/Date/Time-Span				
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
			O	P	B	N	D	T		
Calculation Field Name:		Formula: (+ * - /)								
		=								
		=								
		=								
If Employee Selection Criteria:		Compare Value:				Actions: (C, P, N, E)				
Field Name:		(not) E G L F				<input type="checkbox"/>				
Field Name:		(not) E G L F				<input type="checkbox"/>				
Field Name:		(not) E G L F				<input type="checkbox"/>				



**NEW FIELDS DEFINITION WRITER**

NEW FIELDS DEFINITION WRITER

Date: \_\_\_\_\_

<b>Program Name:</b> X__ __ S C R					<b>Module Specific?</b> Y / N		
<b>Program Title:</b>					<b>Security?</b> Y / N		
<b>Date (circle one):</b> Employee / Company					<b>Segment Code Override:</b>		
<b>Help Description:</b>							
<b>Form Prompts:</b>							
Sequence	New Field Name	Length of Field	Field Type	If Numeric, Number of Decimals	Is this a Key Field (K)?	Is this a Required Non-Key Field (G)?	Option List Name

FIELD TYPE Choices: (blank) Alphanumeric, (N0, N2, N4, N6, N\$) Numeric, (DT) Date, (CD) Codeset Definition

## **GLOSSARY**

---

## A

- Action field** In the Solution View Action form, a required field which contains a value which represents the action to be performed. Action codes allow you to Add, Change, Delete, Copy, Inquire, List User Programs, or List All Programs.
- Action Form** The common starting point from which each of the Solution View facilities is accessed. Once a facility is accessed through the Action form, all the forms for that facility are chained together automatically in an order determined by your entries. The program is created only when you have completely entered all the required forms (some forms are optional).
- Additional Key field** Field in the Command Entry dialog. Fifteen-character entry field. Entries in this field may be a continuation of the KEY field or a special value required by a program.
- Alphanumeric** Type of field that can include any letter of the alphabet and any number, as well as punctuation marks.
- ASCII** One of two character sets used by computers in which the sequencing order is blank, A through Z, then 0 through 9.

## C

- Command Line** Contains five display-only fields and six entry fields that let you specify what form and programs you want to execute.
- Conditional Logic** Parameters in a program that test values in a record to include or exclude it in the output.
- Control Break** An instruction given in a program that prints a total or starts a new page when a specified value changes.
- C12RPT program** See Optional Report Schedule Control Form.
- Cyborg Scripting Language** Cyborg's fourth-generation programming language. Cyborg Scripting Language shares many features with COBOL. This type of code is generated by the Writer facilities.

**D**

- DATE field** A field in the body of a form that allows you to enter a date in specified formats (YYMMDD or MM-DD-YY), or century formats (CCYYMMDD or MM-DD-CCYY).
- Delete This Entry** A program that deletes one occurrence in a multiple-occurrence segment.
- Documentation** On-line and printed text which describes a system and its features, and provides instructions for its use. In The Solution Series, documentation is available on-line and in printed manuals for each module.

**E**

- EBCDIC** One of two character sets used by computers in which the sequencing order is blank, 0 through 9, then A through Z.
- Employee Database** One of the two main files in The Solution Series. It is composed of variable-length records that contain data about your organization, employees, or assets, along with report generator records. It is the file that is being updated when information is typed into data entry forms on-line. Previously called FILE02.
- Extract Writer** A Solution View tool that is used to create programs to generate a data file that may be input to another system.

**F**

- Field** An area on a form in which entries can be made or viewed.
- Field Name Table Menu** Allows you to select field names directly from a multiple level listing by selecting an option in the Option field, or by using the question mark (?) feature in the Field Name Entry field.
- FILE01** See System Control Repository.
- FILE02** See Employee Database.
- F-XREF program** A required process that initially builds the Field Name Table Reference Menu and makes it available to the Solution View program.
- F-MENU Program** A program that allows you to view the attributes of the data fields on file in a user-friendly, menu-driven format. Also allows you to create your own Quick Reference Menu of up to 45 fields, accessed when using Solution View.

## H

- HELP form** A form that gives a brief description of the purpose of a Solution Series form, report or program. It also provides options to view documentation, access English Language tools or return to the last menu. For on-line forms, it provides prompts to help you execute the form.
- History Record** A snapshot of the employee payment document which is available in standard and check stub format on-line.

## I

- Initiate Scheduled Reports** An on-line program that enables you to submit batch job, causes the system to run a report, and lets you send the output to a print file or to the Employee Database for on-line viewing.

## J

- JPRTxxxx** A predefined job stream that executes the PRINT process, where xxxx is the OPERATOR-ID for the user who initiated the batch job.

## K

- KEY field** **In the Command Entry dialog:** Ten-character entry field. Entries in this field work in conjunction with the PROGRAM field to identify the specific data you want the program to access. The kind of information you enter in this field may vary from program to program.
- On Application forms:** A field that the system uses to locate the required record, or one segment within a multiple occurrence segment. The key field or fields must be filled for the system to record a form.

## L

- Labor Record** A record generated by the system for each employee earning generated by a pay document or adjustment.
- List By Form Program** A utility program that lists all the fields on an application form. From it, the user can choose to view documentation on any field.

**M**

- Menu** A list of choices. In The Solution Series, menus lead to documentation or to special HELP forms.
- MENU program** A utility program that enables you to select and access documentation.
- MODULE** An optional field on the Action form that contains the module to be associated with the program, for example, Human Resources, Payroll, Benefits Administration. It is associated with option list SC01.
- Multiple-Occurrence Segment** A segment that may contain more than one instance of a particular kind of data for an organization or employee on the Employee Database, for example, L-Segment, which contains Human Resource information for an employee. Also called a “stacked” segment.

**N**

- New Fields Definition Writer** A Solution View tool that enables you to create your own company or employee entry forms using user-defined fields.
- Numeric field** A field that can contain only numbers.

**O**

- OPERATOR-ID field** Field on the Command Entry dialog. This four-position display-only field is based on the entry you made in the User Code field in the Login dialog. The Operator ID identifies you on all audit trail records for all forms on which you make an entry.
- OPTION Field** A field that resides in the upper left corner of most forms in Solution View, and allows you to perform independent tasks from the form you are currently using.
- Optional Report Schedule Control Form** An optional feature of the Report Scheduler that specified which reports are valid for each organization on your Employee Database. This form is set up by organization.
- Option List** A list of acceptable values that can be entered in an option list field. If entries other than those on the list are made into a field, they are rejected. The list of entries can be accessed by using the Question Mark facility.

## P

- PC Selection Writer** A Solution View tool that creates a program that supplies additional employee selection criteria to the PS-SEG information used to gather company/employee data. (This facility is discussed separately in the Extra for Experts appendix.)
- PROGRAM field** Fifth field in the Command Line. Six-character, entry field. Entries in this field are typically the names of forms or programs. Such entries direct the system to display a form or execute a program.

## Q

- QUERY program** An end-user tool that enables you to execute reports written with the Query Writer facility.
- Query Writer** A Solution View tool that is used to create programs that are run and display output on-line through the QUERY program.

## R

- Record** A component of the System Control Repository which contains multiple fields and is 80 characters in length.
- A group of segments in the Employee Database that contains all the information for an organization or an employee.
- RELOAD Program** The program which compiles a program. In Solution View, it is one of the options that can be selected from the Option field. It is invoked automatically when the last of a Solution View form series has been entered.
- Report Writer** A Solution View tool that is used to create programs that are run in batch and produce hardcopy reports.
- RTPRNT program** The program that formats and prints the report detail to hardcopy.
- RUNC12 program** See Schedule C1-2s for Report Run.
- RUNREP program** See Schedule Reports.

---

**S**

<b>Schedule C1-2s For Report Run</b>	An optional feature of the Report Scheduler which allows you to define which Control 1-2 data will be included in a Report Schedule.
<b>Schedule Reports</b>	A program used to schedule and specify needed parameters for reports.
<b>Form Writer</b>	A Solution View tool that enables you to create your own employee level form using existing segments which contain delivered or user (previously defined) fields.
<b>SEARCH Program</b>	See Search Field Name Table for Field.
<b>Search by Full Name</b>	A program that displays documentation for the specific field entered on the form (or in the KEY field on the Command Line).
<b>Search Field Name Table for Field</b>	A utility program that lists all the fields beginning with the same characters specified by the user. From the list, the user can choose to view documentation on a particular field.
<b>Security Code</b>	An optional field on the Action form that contains the security code which determines who can access and run the Solution View program.
<b>Segment</b>	A component of the Employee Database which is a logical subdivision of an organization or employee record, contains multiple fields and is variable in length, for example, A-Segment, which contains company name and address. A segment can be single-occurrence or multiple-occurrence (stacked). Single-occurrence segments contain data that occurs only once, such as an employee social security number or birth date. Multiple-occurrence segments contain more than one instance of a particular kind of data, such as employee rate changes.
<b>Single-Occurrence Segment</b>	A segment that contains data which occurs only once for the organization or for the employee, for example, E-Segment, which contains employee basic information.
<b>Solution View</b>	A user-friendly program writer, an on-line utility that walks you through the creation of new programs without the direct use of Cyborg Scripting Language (Cyborg's programming language).
<b>Sort Key</b>	One or more fields used by a program to determine the order in which the employees will appear on a report.

**SUBMIT Program** See Initiate Scheduled Reports.

**System Control Repository** One of the two main files in The Solution Series system. It contains programs, option lists, information on fields, and any other information needed to make the system run. Previously called FILE01.

## T

**Table** A group of related records on the System Control Repository that validate activity codes or carries the attributes for a value on the employee record, for example, job code or salary grade. Table information is accessed by the employee record and displays information, but does not add to the data on the employee record.

## U

**User Quick Reference Menu** An optional customized Field Name Reference Menu that contains up to 45 field names specified by operator ID or Module Code.

## V

**VIEW** See View Held Report.

**View Held Report** A program that can be invoked to review report output on-line. You can navigate through the formatted pages which are displayed in VIEW.

## W

**WORD program** See Search Field Name Table for Field.

**WRITER Program** The entry point from which you can access the six facilities that are used to create end-user programs.

## Z

**ZDELETE program** See Delete This Entry.

**ZR Records** Temporary records written to the Employee database when you view a report or query on-line.

# Hewitt

---

## Using Report Generator

Document Issue 1.3

Document Issue Status: Fourth Release  
Document Issue Level: 1.3  
Document Issue Date: October 2004  
Software Version: 3.2 and 5.1

### **Copyright Notice**

Copyright© 2004 Hewitt Associates LLC.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series®  
are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear,  
but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

# Contents

Part 1	1
Introduction	1
Chapter 1	3
<b>About This Manual</b> .....	<b>3</b>
Welcome .....	3
Who should use this manual? .....	3
Prerequisite skills .....	3
How to use this manual.....	3
Conventions used in this manual .....	4
Suggestions and feedback .....	4
Chapter 2	5
<b>Introduction to Report Generator</b> .....	<b>5</b>
Payroll Overview Diagram .....	6
Report Generator - where it fits in .....	7
Sequential Master File (P20) Diagram.....	8
Sequential Master File - its story .....	9
System initialization generators in the Sequential Master File .....	9
Operating report generators in the Sequential Master File .....	9
Application data in the Sequential Master File .....	9
System Flow Diagram- the road from P20 to outputs .....	12
P4CALC.....	13
P4CALC and P5PRNT .....	13
Field Number Table .....	18
Sample Field Number Table .....	19

## Using Report Generator

---

Generator and the COBOL Programs .....	20
The Generator and P4CALC .....	20
Generator content .....	21
Sort transactions .....	21
Format transactions.....	27
Extract Record .....	41
Extract Record Sort .....	41
Extract Record .....	43
Efficiency Considerations.....	47
Subroutines .....	47
Combined extract reporting .....	47
Generator instructions.....	47
Other considerations .....	48
Practice .....	49

## Part 2 51

---

### Using Report Generator 51

---

### Chapter 3 53

---

<b>Generator Maintenance.....</b>	<b>53</b>
Generator maintenance .....	54
Loading and selecting generators .....	54
Adding, replacing, and deleting report renenerators .....	55
Report Selection .....	57
Passing Control.....	61

### Chapter 4 63

---

<b>Writing Logic .....</b>	<b>63</b>
Introduction.....	64
Paragraph names.....	64
Practice .....	69

---

<b>Chapter 5</b>	<b>71</b>
<hr/>	
<b>Counters, Arithmetic, and Conditional Branching .....</b>	<b>71</b>
Introduction.....	72
Counters.....	72
Practice.....	78
<b>Chapter 6</b>	<b>79</b>
<hr/>	
<b>Method Codes .....</b>	<b>79</b>
Method Codes general rules.....	80
Writing a Method Code .....	80
Installing a Method Code.....	80
Operating a Method Code.....	81
User exit points.....	82
Accessing History and Labor .....	85
History processing .....	86
Labor processing.....	86
Multi-master processing .....	87
Interface File Output .....	88
Recycle file .....	88
<b>Chapter 7</b>	<b>89</b>
<hr/>	
<b>Printing .....</b>	<b>89</b>
Printing.....	90
Master File print .....	90
Generating multiple payment documents .....	90
Multiple forms - example .....	92
<b>Chapter 8</b>	<b>95</b>
<hr/>	
<b>Debugging .....</b>	<b>95</b>
Introduction to Debugging Report Generators.....	96
Setting up the test.....	96
Running the test .....	96
Analyzing the results .....	96
Report Generator Debugging Checklist.....	98
Report Generator Debugging Aid.....	102
H2 transaction, column 13, extract debugging record format.....	103

---

**Using Report Generator**

---

<b>Part 3</b>	<b>105</b>
<hr/>	
<b>Appendices</b>	<b>105</b>
<hr/>	
<b>Appendix A</b>	<b>107</b>
<hr/>	
<b>Generator Instructions .....</b>	<b>107</b>
Generator Instructions.....	108
Alphabetic List of Instructions .....	108
Unconditional branching .....	112
Compare instructions.....	113
Edit instructions.....	116
Counter manipulation instructions.....	120
Extract record manipulation instructions.....	122
Index manipulation instructions .....	128
Miscellaneous instructions .....	131
Additional Instructions .....	135
<b>Appendix B</b>	<b>137</b>
<hr/>	
<b>Sample Report Generator .....</b>	<b>137</b>
Sample Report Generator.....	138
Introduction to Sample Report Generator.....	138
<b>Appendix C</b>	<b>145</b>
<hr/>	
<b>Sample Field Table.....</b>	<b>145</b>
Sample Field Number Table .....	146
<b>Glossary of Terms</b>	<b>163</b>
<hr/>	
<b>Index</b>	<b>201</b>
<hr/>	

PART 1

# Introduction

---

## In This Section

About This Manual.....	3
Introduction to Report Generator.....	5



## CHAPTER 1

# About This Manual

---

### Welcome

This manual has been designed to assist you in the use of Report Generator. This manual is a reference document. It is also used in classroom training.

### Who should use this manual?

This manual is designed to be used by a number of different users. The following users will find it most useful:

- COBOL programmers  
Use Report Generator (RG) to customize the supplied payroll reports without modifying any of the system core programs and to create transactions for the Recycle File (P05OUT)file to effect mass changes.

### Prerequisite skills

Users of this manual should possess a variety of technical skills, depending on the roles they will play. At a minimum, all users should have:

- Basic understanding of COBOL

### How to use this manual

This manual has been designed as a reference manual as well as a training manual. It has been written to facilitate self-study before and after classroom training.

#### Table of contents

This manual has been carefully designed for ease of use. All our manuals are written to be task oriented to help you complete your business tasks using our software.

The table of contents lists all the tasks and their respective chapters.

#### Glossary of terms

A Glossary of Terms section is provided to explain terms used in the documentation.

### Index

An index is provided to help you locate specific information.

This document was designed to reduce your need for an index. You should find the table of contents sufficient.

### Conventions used in this manual

The underlying page layout and design of this manual are meant to be as intuitive as possible for you. Our intent is to make it easy for you to navigate through the manual and concentrate on learning and doing.

### Cross-references

Wherever appropriate, we provide cross-references to help you find additional information or further discussion of a specific topic.



*Refer to a cross-reference to find more detail or more discussion on a given topic.*

### Notes

Whenever there is important information you should be aware of, we provide a note.

*Note: You will find tips or quick techniques covered in notes. How to get additional help*

If you cannot find the answers to your questions in this manual, contact Customer Support, who will be able to answer specific questions and give you general advice on training.

Please visit our web site at [www.hewitt.com/ecyborg](http://www.hewitt.com/ecyborg) for the latest schedule of available courses and course descriptions.

### Suggestions and feedback

We value your feedback on our performance support materials. Please forward any comments on this manual to Customer Support.

## CHAPTER 2

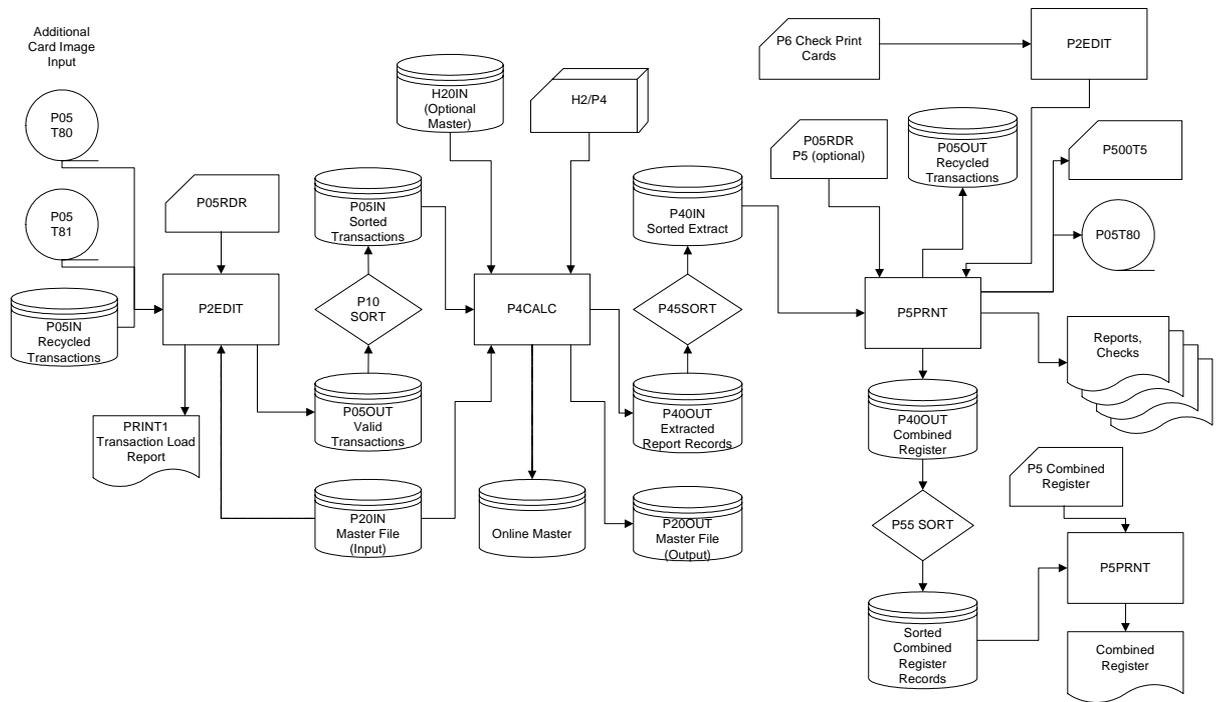
# Introduction to Report Generator

---

## In This Chapter

Payroll Overview Diagram .....	6
Report Generator - where it fits in .....	7
Sequential Master File (P20) Diagram.....	8
Sequential Master File - its story .....	9
System Flow Diagram- the road from P20 to outputs .....	12
P4CALC .....	13
Field Number Table .....	18
Generator and the COBOL Programs .....	20
Extract Record .....	41
Efficiency Considerations .....	47
Practice .....	49

# Payroll Overview Diagram

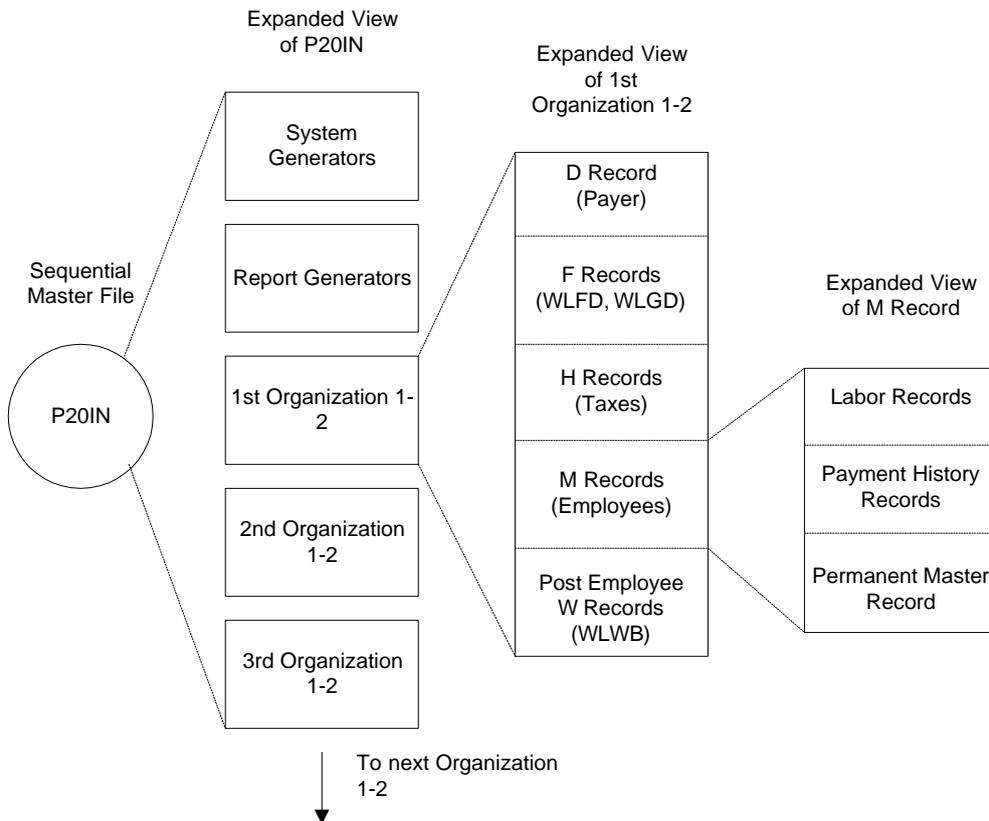


## Report Generator - where it fits in

Report Generator is an integral and versatile part of Payroll Administration. Its abilities allow for easy access and easy use of data, as well as a reduction of errors.

- Report Generator is capable of producing magnetic tape output, as well as printed reports.
- It can access current payroll data, history information, labor distribution, and any other defined user fields.
- Because Report Generator is an interpreter, writing and modifying reports may be accomplished without recompiling any of the system programs.
- Virtually all of the supplied payroll reports are written in Report Generator language. Thus, they may be customized without modifying any of the system core programs.
- The output, the Recycle File (P05OUT), is supported by Report Generator. It is used as an input file the next time the system is run. Also, Report Generator can be written so that it creates transactions for this file, thereby effecting mass changes.
- Changing the Sequential Master File (P20) using Report Generator reduces the chance of error because the recycled transactions must pass through the P2EDIT program.
- All changes made to the system will appear on the Payroll Audit Trail (RPT0101), providing a high degree of control.

# Sequential Master File (P20) Diagram



## Sequential Master File - its story

The P20IN file is the Sequential Master File. The P20IN file contains all payroll data except reconciliation records that will be 'read into' the operation.

In other words, it contains report processing information, company options and parameters, tax specification records and related information for each company, and the payroll and human resource information for each employee.

The P4CALC program creates this file as P20OUT during each payroll run. This P20 is then renamed to be the P20IN for the next payroll run.

Following each payroll run it is the new (updated) Sequential Master File.

The CBSVB program may create the Sequential Master File as FILE12 and may access it as FILE11 and FILE13 for further processing.

The P4CALC program always reads this file as P20IN. P20IN's logical record length is 256 bytes, the first four of which are a record counter, beginning with 1.

The data in P20IN is organized into record groups. Each record group spans one or more logical records and comprises the record group length, the company ID, a record type identifier, and the data.

All the record groups for each company are grouped together.

### System initialization generators in the Sequential Master File

The first records on the Sequential Master File are report generator object components, starting with system or table generators.

These generators are not reports. They are used by the P2EDIT and P4CALC programs to initialize working storage areas.

### Operating report generators in the Sequential Master File

Following the system generators are the available report generators. The beginning of the Payroll Audit Trail report (the Loaded/Not Loaded portion) shows a list of the generators currently on your Sequential Master File.

### Application data in the Sequential Master File

The report generator records are followed by the application data organized sequentially by company. Within each company, the data is separated into the five record types described in the following table.

Record Type	Data Description
D	Payer
F	W records
H	Tax specification records
M	Employee records
W	Post-employee records

Each record type is organized into logical units called record groups.

Each unit of information is composed of one or more 256-character logical records that contain the complete set of information.

Examples of such units are the complete company header or an employee's Permanent Master Record.

Each record group begins with the same three pieces of information following the record number, as listed below.

Description	Length
Group length	3
Company	6
Record Type	1

Record Type D, often referred to as the Company Header Record, contains the high-level information for the company. The company name and address, the earning and deduction parameters, and the report request schedule are all found in this record.

- Type F records contain W data (WLFD and WLGD). Type W records contain post-employee W records with keys such as WLWB. These records are stored following all of the employee data in a company.

There is a maximum of three WL records in each P20IN F record.

The tax specification record information is stored in Type H records. There is one record group for each taxing authority. Some examples of taxing authorities are federal, FICA-OASDI, FICA-HI, state tax, or local tax.

Employee information is stored in Type M records. Three different types of employee record groups exist: Labor Records, Payment History Records, and Permanent Master Records.

During a payroll run, a multiple master is created for each payment or adjustment. The maintenance step of the payroll run, converts the multiple master to a Payment History record group.

Labor Records reflect the basic unit of labor costing. They are created at the rate of at least one for each time entry, automatically paid earning, or earning adjustment.

If an employee has more than one occurrence of location or charge-to information, the system may create more than one Labor Record for each time entry.

The P4CALC program creates Payment History Records for each adjustment or payment that it processes or creates. These records include no to-date information; in other words, only current activity is shown. Only the earning, deduction, or tax authority records affected by the payment or adjustment are included in the Payment History Record.

Permanent Master Records are the permanent records of employee earnings, taxes, and deductions. This is the only record group that contains month-to-date, quarter-to-date, and year-to-date accumulations. An employee's Permanent Master Record also contains all human resource information that has been entered for that employee.

An employee has multiple master records if multiple payments or adjustments have been processed in a payroll cycle. These multiple master records are converted into Payment History Records during Step 5 of the payroll run.

Those record groups that take up more than one physical file record have the length, company, and data type in the first record in the group only, as shown below. The P4CALC program uses the length to control the number of reads necessary for a complete group.

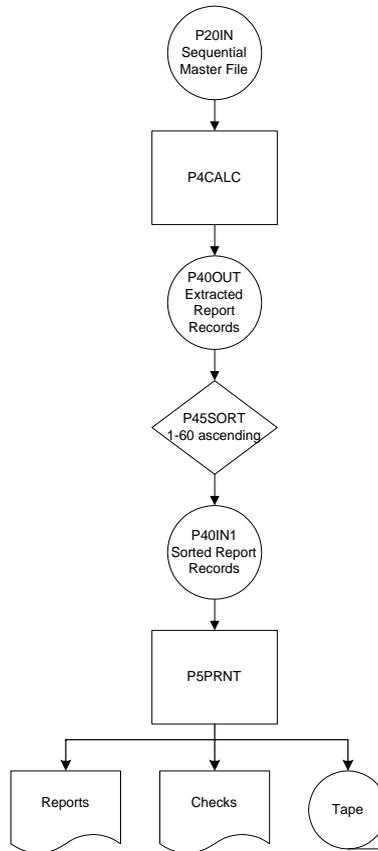
The record layout below shows an example of a record group that requires more than one file record.

1	2	3	4	5	6	7	8													
1	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0
nnnn	mmm	999999M1234	99EE	123-45-6789	11	M01629	CODE3390715	221A15												
nnnn	CONTINUATION RECORD																			
nnnn	3RD RECORD																			
nnnn	4TH RECORD																			

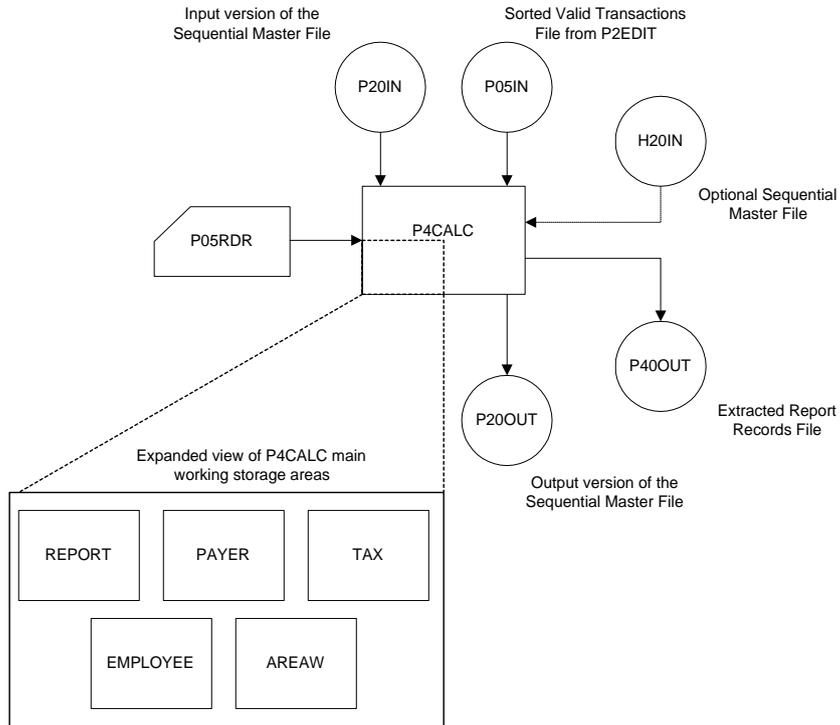
nnnn = record number

mmm = record length

## System Flow Diagram- the road from P20 to outputs



# P4CALC



## P4CALC and P5PRNT

The P4CALC program performs five major functions:

- It reads the Sequential Master File, P20IN.
- It applies any maintenance transactions from the P05IN file.
- It calculates pay for all employees whose frequencies are being paid.
- It extracts information for printed reports and other forms of output to the P40OUT file.
- It writes the updated Sequential Master File, P20OUT, which includes updated earning, deduction, and tax to-date accumulations in the employee records.

P4CALC also performs these minor functions:

- Creates Payment History Records
- Creates Labor Records
- Archives Payment History and Labor Records
- Can be used with the P5PRNT program as a report-only system

During a normal payroll run, the P4CALC program reads information from the Sequential Master File, updates this information with transactions from the P2EDIT program, calculates pay, extracts data for reporting purposes, and writes the information to the new Sequential Master File.

P4CALC begins by reading the P05RDR file. The P05RDR file may contain information that controls the nature of the run. For example, you can tell P4CALC not to calculate pay or not to process the transaction input file.

The P4CALC program conducts a sequential process that merges P05IN file transactions with the P20IN file information. During this process, P4CALC adds, deletes, and modifies information as necessary.

At the same time, P4CALC creates P20OUT, the new Sequential Master File. P4CALC also writes to P40OUT, the Extracted Report Records File.

The P45SORT function then sorts the P40OUT file, in ascending sequence on the first 60 characters, for input to the P5PRNT program.

### Processing in the Sequential Master File

During a P4CALC execution, the program stores information from the Sequential Master File in one of the five major areas shown in the chart below.

Area	Information Stored
REPORT	Report generator logic
PAYER	Company information
TAX	Tax specification information
EMPLOYEE	Employee information
AREAW	W (miscellaneous) information

Depending on your specific needs for maintaining information, you may have to expand these areas.

The first items that P4CALC processes in the Sequential Master File are the report generators.

The Payroll Audit Trail report lists the name of every report generator that the P4CALC program finds, and it shows whether or not each is loaded.

Loaded, in this context, means that a report is placed into the report area (a Working Storage area) of the program (P4CALC) and is available for processing. Any of the organizations on the Sequential Master File may select the generation for processing organization data on the Sequential Master File.

Being loaded also implies that eight control records for each report generator are written to the P40OUT file. These records contain format and total break information for the print program.

Following the report generator data, P4CALC processes one or more complete companies.

Each company includes company information, miscellaneous (W) data, tax specification information, Payment History and Labor Records, and employee Permanent Master Records. Each record type is placed, in turn, into the appropriate working storage area and updated if input transactions are found in the P05IN file.

The first information P4CALC reads in a company is the company header record group. This action is also termed a company header pass.

At this point the system checks the report scheduler information from the Report Requests form (DD-SCR) to determine whether or not any active report generators should be processed for this record group.

This procedure continues through the W and tax specification records, with the report generator schedule tested for each entity.

Eventually, the procedure reaches the employee information. All records for each employee are found together on the Sequential Master File, starting with that employee's oldest labor information.

Following the oldest Labor Record(s) are the Payment History Records from that oldest payment or adjustment, the Labor Records from the next payment, and so on.

Each record group, whether Labor or Payment History, causes a test of the report generator schedule for potential reporting.

The last record group for each employee is the Permanent Master Record. This final record group is the only one that contains the employee's earning, deduction, and tax record information, as well as human resource data.

If there are to-date adjustments in the P05IN file, they are processed now. Labor records may be created if earnings are being adjusted. Payment History and Labor Records are created for each adjustment batch. Again, the report generator schedule is tested for labor and adjustment report requests.

The next step the P4CALC program performs is to determine whether or not the employee is to be paid.

If so, P4CALC processes time documents and automatic earnings, taxes, and deductions. The report generator schedule is exercised for each Payment History and Labor Record created.

P4CALC then writes the Permanent Master Record group to the P20OUT file and processes the next employee's information.

After all employee records in a company have been processed, there may be post-employee W data.

Following this, the P4CALC program processes the next complete company, and so on, until the end of the P20IN file is reached.

Once the P4CALC program execution is ended, the P45SORT utility sorts the Extracted Report Records File, P40OUT.

The P5PRNT program then processes the sorted version of the Extracted Report Records File, now called P40IN1.

The P5PRNT program formats the sorted Extracted Report Records File, P40IN1, for external presentation. This presentation may be either on paper or magnetic medium.

In addition, P5PRNT creates the Recycle File, P05OUT, which the system accesses on its next run.

### **Sequential Master File processing with P4CALC and P5PRNT**

During a P4CALC execution, the program stores information from the Sequential Master File in one of the five major areas shown in the chart below.

<b>Area</b>	<b>Information Stored</b>
REPORT	Report generator logic
PAYER	Company information
TAX	Tax specification information
EMPLOYEE	Employee information
AREAW	W (miscellaneous) information

Depending on your specific needs for maintaining information, you may have to expand these areas.

The first items that P4CALC processes in the Sequential Master File are the report generators.

The Payroll Audit Trail report lists the name of every report generator that the P4CALC program finds, and it shows whether or not each is loaded.

Loaded, in this context, means that a report is placed into the report area (a Working Storage area) of the program (P4CALC) and is available for processing. Any of the organizations on the Sequential Master File may select the generation for processing organization data on the Sequential Master File.

Being loaded also implies that eight control records for each report generator are written to the P40OUT file. These records contain format and total break information for the print program.

Following the report generator data, P4CALC processes one or more complete companies.

Each company includes company information, miscellaneous (W) data, tax specification information, Payment History and Labor Records, and employee Permanent Master Records. Each record type is placed, in turn, into the appropriate working storage area and updated if input transactions are found in the P05IN file.

The first information P4CALC reads in a company is the company header record group. This action is also termed a company header pass.

At this point the system checks the report scheduler information from the Report Requests form (DD-SCR) to determine whether or not any active report generators should be processed for this record group.

This procedure continues through the W and tax specification records, with the report generator schedule tested for each entity.

Eventually, the procedure reaches the employee information. All records for each employee are found together on the Sequential Master File, starting with that employee's oldest labor information.

Following the oldest Labor Record(s) are the Payment History Records from that oldest payment or adjustment, the Labor Records from the next payment, and so on.

Each record group, whether Labor or Payment History, causes a test of the report generator schedule for potential reporting.

The last record group for each employee is the Permanent Master Record. This final record group is the only one that contains the employee's earning, deduction, and tax record information, as well as human resource data.

If there are to-date adjustments in the P05IN file, they are processed now. Labor records may be created if earnings are being adjusted. Payment History and Labor Records are created for each adjustment batch. Again, the report generator schedule is tested for labor and adjustment report requests.

The next step the P4CALC program performs is to determine whether or not the employee is to be paid.

If so, P4CALC processes time documents and automatic earnings, taxes, and deductions. The report generator schedule is exercised for each Payment History and Labor Record created.

P4CALC then writes the Permanent Master Record group to the P20OUT file and processes the next employee's information.

After all employee records in a company have been processed, there may be post-employee W data.

Following this, the P4CALC program processes the next complete company, and so on, until the end of the P20IN file is reached.

Once the P4CALC program execution is ended, the P45SORT utility sorts the Extracted Report Records File, P40OUT.

The P5PRNT program then processes the sorted version of the Extracted Report Records File, now called P40IN1.

The P5PRNT program formats the sorted Extracted Report Records File, P40IN1, for external presentation. This presentation may be either on paper or magnetic medium.

In addition, P5PRNT creates the Recycle File, P05OUT, which the system accesses on its next run.

## Field Number Table

Field names are given corresponding field numbers by Report Generator. Field names are commonly used, but it is important to know the field numbers as well.

The following list contains explanations of all predefined field numbers. The field number contains the following information for each field:

*Note: If any user defined fields have been added to the Field Number Table, a new Field Number Table must be printed. This may be accomplished by placing a 'P' in column 22 of the first batch transaction. The new Field Number Table will be printed as a component of the Transaction Load Report.*

<b>Field Number</b>	A three-digit number, assigned to each field.
<b>Segment</b>	One or more pieces for each type of record on the Master File.
<b>Transaction Code</b>	It is located in columns 1 and 2 of the transaction. This location is normally used to enter the field. See Appendix A for a listing of transaction codes and their meaning.
<b>Description</b>	The name of the field (without dashes as is required in the online system).
<b>Type of Field</b>	<p>The kind or make up of a field name.</p> <p><b>Numeric:</b> Only the digits 0–9 and spaces</p> <p><b>Alphanumeric:</b> Any number, letter, or special character. (Entering an 'A' in the first position will space fill the field.)</p> <p><b>Date:</b> YYMMDD, this allows for only numbers that equal years, months, and days. For example, July 14, 1999 equals 990714. (Entering an 'I' in the first position will zero fill the field.)</p> <p><b>Computational:</b> Is similar to Numeric, the only variation is the internal representation of the field.</p> <p><b>LAST FIRST:</b> A name field, which can only be entered as 'last name, first and middle names or initials'. For example, John J. Smith III should be entered as 'Smith III, John J.'.</p>
<b>Input Length</b>	The size of the field, as a result of entering the field using the transaction specified in the Transaction Code column.
<b>KEY MAINT</b>	For user-defined fields, a value of 'K' indicates that the field is entered in a 'stacked' transaction on the Master File. (A stacked transaction means there is more than one transaction present.)

*Note: Changing the Key Maint field from an '8' to a '9' will not trigger a Master File Print; that is triggered by changing the field with an 'X' transaction or the appropriate transaction.*

**Sample Field Number Table**

FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
001			CONTROL 1	ALPHA-NUMERIC	2	8
002			CONTROL 2	ALPHA-NUMERIC	4	8
003			EMPLOYEE NUMBER	ALPHA-NUMERIC	10	8
004	E		RECORD GROUP	ALPHA-NUMERIC	2	8
005	A	AA	CONTROL 1 NAME	ALPHA-NUMERIC	30	8
006	A	AA	CONTROL 2 NAME	ALPHA-NUMERIC	30	8
007	A	AB	CONTROL ADDR 1	ALPHA-NUMERIC	30	8
008	A	AB	CONTROL ADDR 2	ALPHA-NUMERIC	30	8
009	A	AC	CONTROL CITY ST	ALPHA-NUMERIC	25	8
010	A	AC	CONTROL ZIP	NUMERIC	5	8
011	A	AD	CONTROL 1 TITLE	ALPHA-NUMERIC	12	8
012	A	AD	CONTROL 2 TITLE	ALPHA-NUMERIC	12	8
013	A	AD	CONTROL 3 TITLE	ALPHA-NUMERIC	12	8
014	A	AD	CONTROL 4 TITLE	ALPHA-NUMERIC	12	8
015	A	AD	CONTROL 5 TITLE	ALPHA-NUMERIC	12	8
016	A	AD	CONTROL 6 TITLE	ALPHA-NUMERIC	12	8
017			NOT YET IN USE			
018			NOT YET IN USE			
019			NOT YET IN USE			
020			NOT YET IN USE			
021			NOT YET IN USE			
022			NOT YET IN USE			
023			NOT YET IN USE			
024			NOT YET IN USE			
025			NOT YET IN USE			
026			NOT YET IN USE			
027	A	AE	CLEAR TO DATE	NUMERIC	1	8
028	A	AE	REPORT GENERATE	NUMERIC	1	8
029	A	AE	PURGE TERMINATE	NUMERIC	1	8
030	A	AE	RUN DATE	DATE YYMMDD	6	8
031	A	AE	RUN TYPE	NUMERIC	1	8
032	A	AE	VERSION NUMBER	NUMERIC	2	8
033	A	AE	PRINT UPDATE	ALPHA-NUMERIC	1	8
034	A	AE	REPORT SELECTS	ALPHA-NUMERIC	12	8
035	A	AE	USER DATE	DATE YYMMDD	6	8
036	A	AE	USER FIELD	ALPHA-NUMERIC	4	8
037	A	AE	RESERVED	ALPHA-NUMERIC	3	8
038	A		HDR PERIOD 1		6	8
039	A		HDR PERIOD 2		6	8
040			NOT YET IN USE			
041			NOT YET IN USE			
042			NOT YET IN USE			
043	C		TRANS CODE	ALPHA-NUMERIC	2	8
044	E		ADJUSTMNT BATCH	ALPHA-NUMERIC	1	8
045	L		L CHANGE	ALPHA-NUMERIC	1	
046			UNKNOWN FIELD	ALPHA-NUMERIC	15	8
047			PLUG CODE	NUMERIC	1	8
048	E		ACTION CODE	ALPHA-NUMERIC	1	8

## Generator and the COBOL Programs

A generator is a program written using the Report Generator language. The generator gives instructions to the P4CALC and P5PRNT programs.

Below are the functions of the codes that comprise the generator.

### P4CALC

R0	Determines what data this generator may access
S7	Builds the sort key portion of the extract record
R7	Builds the data portion of the extract record

### P5PRNT2

S1	Determines control breaks
R1	Determines control breaks
R2	Supplies heading data
R3/R4	Reformats the extract
R5/R6	Record data to output layout

A generator must be entered into the system (through program P2EDIT) behind a batch transaction with '999999' in the Organization 1-2 field.

## The Generator and P4CALC

Once a generator is selected to process data for an organization, control is passed to it whenever a record is encountered that contains the type of data requested.

When control is passed to a generator it means that the Report Generator portion of P4CALC begins interpreting the instructions entered on the R7 transactions.

If more than one kind of data is requested, for example, if month-to-date and year-to-date data is requested, control will be passed to the generator twice for each employee's permanent master record.

The extract file is sorted by P45SORT and the sorted file becomes input to P5PRNT.

Because of the sort, all of the extract records created by a generator are sorted down behind the S1, R0, R1, R2, R3, R4, R5, and R6 information previously written out by P4CALC.

P5PRNT finishes the job by producing the desired output file.

When writing a generator, it is important to keep in mind the three-step process that takes place when the generator is run:

1. Create and write extract records (P4CALC).
2. Sort the extract records (P45SORT).
3. Format and produce the output file (P5PRNT).

## Generator content

A generator consists of two sections: sorting specifications (referred to as the 'sort') and formatting specifications (referred to as the 'format').

A sort is made up of at least one, but no more than nine S1 transactions, followed by one or more S7 transactions. The S7 transactions contain instructions that will be executed to build the sort key.

The S1 transactions specify:

1. When totaling breaks are to occur
2. When page breaks are to occur
3. The length of the sort key

A format is made up of one R0 transaction followed by one or more R7 transactions.

Between these two transaction types there may also be an R1 transaction, one or more R2 transactions, an R3 transaction, an R4 transaction, an R5 transaction, and an R6 transaction.

*Note:* *The R7 transactions, together with the S7 transactions, are roughly equivalent to a COBOL program's procedure division.*

## Sort transactions

### **S1 sort transaction (used by P5PRNT)**

The S1 transaction determines control breaks.

It decides which fields will cause a total break and how many spaces will be associated with the totals. The highest-level total is defined first. Each additional S1 transaction defines an additional break or total, progressing from major to minor.

Each sort must have at least one, but no more than nine S1 transactions.

S1 transactions are used to specify fields in the sort key which, when they change, will trigger a total break.

Each sort must have at least one but not more than nine S1 transactions.

- If no total breaks are to occur, use an S1 transaction with the Sort Code in columns 3–4 and 151 in columns 8–10 (a semi-arbitrary convention). The remainder of the transaction beyond column 10 should be left blank, except for columns 15–16 which will contain the Sort Key Length.
- If multiple S1 transactions are used because multiple total breaks are desired, they should be ordered so that the first S1 transaction specifies the highest level total break and the last S1 transaction specifies the lowest level total break.

*Note:* *The S1 transactions do not have to account for the entire sort key, only those portions that are to trigger total breaks.*

When the field specified by the first S1 transaction changes, the system will reset the page counter and force a page break after total line processing (if any) is performed.

To produce a report containing detail information for employees having a common Pay Level (Control) 1 but different Pay Level (Control) 2s, set up a fake Pay Level (Control) 2.

These are examples of fake Pay Level (Control) 2s:

SIXY010001  
S7XY020LIT011  
S7XY030FLD001  
S7XY040LIT04EAST  
S7XY050FLD003

The literal must be a valid Pay Level (Control) 3 associated with the Control 1. The standard heading lines will print the File Version number and Period End Dates for this particular Organization (Control) 1 and 2.

### **Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, S1.

### **Cols 3–4 (alphanumeric) Sort Code (198)**

Assign a letter to the first position (column 3). Then assign any number, letter, or character to the second position.

### **Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the sort.

This field is continued in columns 77–80 of the transaction.

Each transaction must have a Sequence Field that is equal to or greater than the previous transaction.

### **Cols 8–10 (numeric) Field Number (190)**

Enter the Field Number of a field in the sort key that is to trigger a total break, or leave it blank and make an entry in the Field Length field in columns 13–14.

Do not make entries in both places.

### **Cols 11–12 (numeric) Key Position (200)**

This field is used to identify the location of the control break field within the sort key field.

This entry is required only if this break field does not immediately follow the field named in the preceding S1 transaction. If that is the case, then enter the total number of positions (including the Forms Code) occupied by fields in the sort key that precede this field.

### **Cols 13–14 (numeric) Field Length (194)**

Enter the size of a field in the sort key that is to trigger a total break, or leave blank and make an entry in the Field Number field in columns 8–10.

Do not make entries in both places.

### **Cols 15–16 (numeric) Sort Key Length (199)**

Enter the total length of the sort key on the last S1 transaction.

If left blank, the system will attempt to compute this number by adding up the lengths of the fields moved by the S7 transactions.

If any branching is done on the S7 transactions, the system will not be able to compute the correct value. In this case, the user should enter a value here.

**Cols 17–31 (alphanumeric) Total Line Description (201)**

When a total break occurs, the system will print a 15-character description of the field that caused the total break on the left-hand side of the total line.

If an entry is made in this field, it will be used as the total line description.

If this field is left blank, the Field Number field (columns 8–10) is checked.

If Field Number, too, is blank, the Total Line Description is left blank.

If a Field Number was entered, the total line description is taken from the AD transaction for fields 001, 002, and 094–097 (Controls) or from the Field Number Table for all other fields.

**Col 32 (alphanumeric) Stagger Totals (202)**

b or N = Do not stagger totals

Y = Stagger totals on 2 lines to avoid truncating when the figures may exceed the print space on one line.

1–6 = Do not stagger totals but print this number of digits on the high order (the digit in the left most position of a number) in addition to the 12 positions that are normally printed.

Total amounts are stored in 18 digit counters.

Normally, only the low-order 12 digits are printed because this is usually sufficient for most totals. If you expect a total to be larger than this, then use the third option to indicate the number of additional high-order digits needed for the total.

**Col 33 (numeric) Page Advance Before/After (203)**

b or 0 = No advance

1 = Advance before printing

2 = Advance after printing

3 = Advance before and after

4 = Take a page total

The page total option may only be specified on a S1 transaction. This S1 transaction must specify that 1 position must immediately follow the field specified by the transaction. The value of the 1 position field must remain constant.

### **Col 34 (alphabetic) Zero Page Counter (204)**

b or N = Do not reset page counter

Y = Reset the page counter to 1 after this total break, and force a page break for the last S1 transaction

For the first S1 transaction, the system always behaves as though a Y were entered in this field.

### **Col 35 (alphanumeric) Line Advance Before (205)**

b = 1 line

0 = 2 lines

- = 3 lines

2-9 = Skip down to the line number specified in the R0 transaction (see columns 48-63 of the R0 transaction).

This option controls the skipping that occurs before a total line is printed.

If, however, the T option is used (see column 28 of the R1 transaction) this field controls the editing of the total figures.

The editing that is performed is as follows:

b = PIC S9(8)V99

0 = PIC 9(8)V99

- = Normal R5/R6 editing

### **Col 36 (alphanumeric) Line Advance After (206)**

b = 1 line

0 = 2 lines

- = 3 lines

This option controls the skipping that appears after a total break occurs. This option is operative even if the printing of the total line has been suppressed using column 37 of this S1 transaction.

### **Col 37 (numeric) Highest Total (207)**

b or 0 = Print subtotal

1 = Print grand total

2 = Do not print subtotal

3 = Do not print grand total

This field specifies the type of total break (subtotal or grand total) and whether or not the total line is to be printed.

When a subtotal is processed, the subtotal figures (whether they are printed or not) are rolled up into the next higher set of accumulators.

When a grand total is processed, the figures are not rolled up. Thus, although all total breaks may be designated as subtotals, it is more efficient to designate the highest one as a grand total.

**S7 sort transaction (used by P4CALC)**

The S7 transaction defines the sequence of the sort.

Each sort must have at least one S7 transaction. S7 transactions contain the instructions that will build the sort key portion of an extract record.

**Cols 1–2 (alphanumeric) Transaction code**

This is a constant, S7.

**Cols 3–4 (alphanumeric) Sort Code (198)**

Enter the same code as used in columns 3–4 of the S1 transaction(s).

**Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the sort.

This field is continued in columns 77–80 of the transaction.

Duplicate numbers may appear in columns 5–7 as long as the numbers in columns 77–80 have ascending values.

Each transaction must have a Sequence Field that is equal to or greater than the previous transaction.

**Cols 8–10 (alphanumeric) Routine Number (197)**

Enter an instruction mnemonic.

Refer to the heading R7/S7 Coding Considerations.

If '\*' is entered here, columns 11–76 are ignored (that is, this is a comment transaction).

**Cols 11–42 (alphanumeric) Operand Field**

Enter the instructions operand (if one is required) left justified. Comments may be entered in this field to the right of an operand.

**S7 coding considerations**

The first position of a sort key must always contain a Forms Code. Program P5PRNT uses the Forms Code to determine which of its output files is to be written. The following table contains a complete list of valid Forms Codes.

Form Codes	P5PRNT File (FD) Name	Comments
0	PRINT1	Printer reserved for sort 01
1	PRINT1	Printer
2	PRINT2	Printer
3	PRINT3	Printer
4	PRINT4	Printer

Form Codes	P5PRNT File (FD) Name	Comments
5	P500T5	80 character output file
6	P05T80	80 character output file
7	H50OUT	Print image (133 characters with carriage control) file
8	P05OUT	Recycle file
9	P04OUT	Reserved for sort 22
P	P50CDD	Canadian Direct Deposit Tape
Q	P50CT4	Canadian T4/TP4 Tape
R	P50W2T	W2 tape
S	P50ACH	ACH tape
T	PRINTT	Printer—checks
U	PRINTU	Printer—checks
V	PRINTV	Printer—deposit statements
W	PRINTW	Printer—checks
X	P51W2T	NY State Quarterly Tape
Z	P00OUT	Report will be printed during the second pass of P5PRNT using the PRINT1 file
\$	P50SBB	Service Bureau Statistics Tape

The user may add additional output files to P5PRNT, assigning each one a unique Forms Code beginning with A, continuing through K.

Except for Forms Codes 8 and \$, positions 2–5 of a sort key must contain a Report Code.

A Report Code is made up of a two-position Sort Code, followed by a two-position Format Code. This holds true unless the Organization (Control) 1 and 2 names are suppressed from the report headings using options on the R1 transaction.

Positions 6–11 of a sort key must always contain the Organization 1-2 value (fields 001 and 002).

The simplest and most efficient method of satisfying these requirements is to use the SFC (176) instruction. It will place the Forms Code (entered in the Operand Field), Report Code, and Organization 1-2 values into the first 11 positions of the extract record.

The remainder of the sort key may contain whatever the user desires. Any of the R7 transaction instructions may be used to this end. Most commonly, however, the FLD (017) instruction will be used to move fields from the company header record and/or an employee's record to the sort key.

For Forms Code 8, positions 2 through 119 of the extract record should contain a complete and exact image of the record to be written to the recycle file (P5OUT).

The format of the recycle file follows:

Positions	Description of Contents
1-1	Forms Code 8
2-7	Organization 1-2
8	J (employee transaction) I (company transaction)
9-18	Columns 3-12 of the transaction
19-46	Spaces
47-48	Columns 1-2 of the transaction
49-116	Columns 13-80 of the transaction
117-119	Filler

For Forms Code Z, program P5PRNT will completely format the report and write it to the P40OUT file.

Each record will consist of a Forms Code (now a '1') followed by the Organization 1-2 value, the Report Code, a generated sequence number, and a 133 position print image.

Notice that the relative positions of the Report Code and Organization 1-2 values have been reversed.

Thus, if several reports use a Forms Code of 'Z', they will be printed during the second execution of P5PRNT (instead of the first), and all of the reports for a given company will be grouped together.

For Forms Code 1 and 0, if a 'Z' is entered in column 23 of the H2 transaction, program P4CALC will automatically change the Forms Code to a Z just before each extract record is written out.

For Forms Code \$, positions 2-81 of the extract record must contain an image of the record to be written to P50SBB.

Under no circumstances should a sort key ever be constructed that contains low values in positions 6 and 7 for any Forms Codes except 8 and \$.

## **Format transactions**

### **R0 format transaction (used by P4CALC)**

The R0 transaction defines what type of Sequential Master File data the generator may access. The format name (columns 24-47) will print on the audit trail.

Report Select (column 20) determines when a generator will be loaded.

The R0 transaction is used to specify which types of records may be processed by this Report Generator. It also is used to supply a descriptive name for the Report Generator that will appear on the Payroll Audit Trail whenever the generator is selected.

Finally, it is used to supply the physical line numbers to be skipped down to whenever a carriage control character in the range of 2 through 9 is encountered.

Every format must have one and only one R0 transaction.

### **Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, R0.

### **Cols 3–4 (alphanumeric) Format Code (165)**

Assign a letter to the first position (column 3) and then any number, letter, or character to the second position.

A number in the first position is reserved (not available to customers) for internal use. It is used for formats.

### **Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77-80 of the transaction.

Duplicate numbers may appear in 5-7 as long as 77-80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

### **Record Selection**

Columns 8–18 of the R0 transaction specify which types of records may be processed by the Report Generator.

For each Organization 1-2, the corresponding columns on the D transaction determine when and which types of records will actually be processed.

Entries may only be made in columns 8–18 of a D transaction for which entries have been made in the corresponding columns of the R0 transaction.

Thus, the R0 transaction determines which types of record may be processed, and the D transaction specifies when and which of these are actually processed.

### **Col 8 (numeric) Plus Is Valid**

1 = Plus adjustments may be requested on the D transaction.

### **Col 9 (numeric) Minus Is Valid**

2 = Minus adjustments may be requested on the D transaction.

### **Col 10 (numeric) Manual Is Valid**

3 = Manual adjustments may be requested on the D transaction.

### **Col 11 (numeric) Current Is Valid**

4 = Current figures may be requested on the D transaction.

### **Col 12 (numeric) Month Is Valid**

5 = Month-to-date figures may be requested on the D transaction.

### **Col 13 (numeric) Quarter Is Valid**

6 = Quarter-to-date figures may be requested on the D transaction.

### **Col 14 (numeric) Year Is Valid**

7 = Year-to-date figures may be requested on the D transaction.

**Col 15 (alphabetic) Company Header Is Valid**

C = Header (Organization 1-2) information may be requested on the D transaction.

**Col 16 (alphabetic) Taxes Are Valid**

T = Tax body information (T1-T3 transactions) may be requested on the D transaction.

**Col 17 (alphabetic) Labor Is Valid**

L = Labor records may be requested on the D transaction.

**Col 18 (alphabetic) Other Is Valid**

O = 'Other' records may be requested on the D transaction.

**Col 19 (alphabetic) Data Type**

b = Payroll data

L = Labor data

H = History data

A = All data types

This field, in conjunction with columns 8–18, determines exactly what types of data may be processed by the Report Generator.

The following table illustrates how various types of data may be processed.

<b>Data Type</b>	<b>Cols. 8–18</b>	<b>Col. 19</b>
+ Adj. History record	1 in col. 8	H or A
- Adj. History record	2 in col. 9	H or A
Manual adj. History record	3 in col. 10	H or A
Computer payment history record	4 in col. 11	H or A
All types of labor record	L in col. 17	L or A
Current + adjustments	1 in col. 8	b or A
Current - adjustments	2 in col. 9	b or A
Current manual adjustments	3 in col. 10	b or A
Current computer payments	4 in col. 11	b or A
Permanent master record	4 in col. 11	b or A
Permanent master record	5 in col.12	b or A
Permanent master record	6 in col.13	b or A
Permanent master record	7 in col. 14	b or A
Company header record	C in col. 15	b, L, H, or A
Tax record(s)	T in col.16	b, L, H, or A
'Other' records	0 in col.18	b, L, H, or A

There are several facets of this table that require additional explanation.

- When an 'A' is used in column 19, the Record Type field (004) of the master record must be examined to determine which record is being processed. This field will contain one of the following values:

Field 004	Record Type
+L	+ Adj. labor record
-L	- Adj. labor record
ML	Manual adj. labor record
nL	Computer payment labor record
Xb	Budget labor record
+H	+ Adj. history record
-H	- Adj. history record
MH	Manual adj. history record
Nh	Computer payment history record
+b	Current + adjustment
-b	Current – adjustment
Mb	Current manual adjustment
mm	Current computer payment
99	Permanent master record

where 'n' is in the range 0 through 9

'mm' is in the range 01 through 98

'x' is '+' or '-'

- Column 16 results in a pass through the Report Generator for each tax body on the master file that has a 'Y' in column 80 of its T1 transaction.
- With 'b' or '~' in column 19, column 11 causes a pass through the Report Generator with the employee's permanent master record, whether or not she or he is paid. If the employee is paid, one additional pass is made for each computer-produced payment, except for the last one (that is, no payment = 1 pass; 3 payments = 3 passes). The current amounts for the last payment are present in the employee's permanent master record.
- With 'b' or 'A' in column 19, columns 11–14 each cause a pass through the Report Generator with the very same permanent master record. The only difference is the initial setting of the H and J To-Date indexes. Regardless of the pass, the Report Generator can always manipulate these indexes and access current, month-to-date, quarter-to-date, and year-to-date figures.
- Column 15 should not be used unless a special pass through the Report Generator with only the company header record available is required. The company header record data is always available to a Report Generator when employee or 'other' record types are being processed.

- Whenever a record is made available to a Report Generator, the value entered in the column causing the pass is present in field 152. For example, when a manual adjustment record is passed to a Report Generator, field 152 will contain a '3'.

**Col 20 (alphanumeric) Run Select**

0 = Load every run

1 = Do not copy to output master

2 = Execute immediately (reserved for internal use only)

3 = Previously executed (reserved for internal use only)

any other = Must match column 16 of H2 to load.

This field is used to determine whether or not the Report Generator will be loaded for this run. If this field is left blank and no H2 transaction is used, (or column 16 of the H2 transaction is blank), then this generator will be loaded.

**Col 23 (alphanumeric) Generator Type**

This field identifies the type of generator being loaded for this run.

b = Report Generator

**Cols 24-47 (alphanumeric) Format Name**

Enter the name of the Report Generator. It will appear on the Payroll Audit Trail in the Selected/Not Selected report whenever the Report Generator is selected.

**Cols 48-49; 50-51; 52-53; 54-55; 56-57; 58-59; 60-61; 62-63 (numeric) Line Number**

Enter the line number to be skipped down to when a carriage control in the range 2-9 is encountered. Columns 48 and 49 are for a carriage control character of 2, 50 and 51 for a 3, and so forth.

These fields eliminate the need for special carriage tapes (vertical format loops) except when the form has a non-standard number of lines.

**Cols 64-65 (alphanumeric) Sort Name**

This field may be used to enter the name of a sort that is already present on the Master File.

If this field is filled in transactions S1 through S7 need not be entered with this format.

**R1 format transaction (used in P5PRNT)**

The R1 transaction is used to select heading lines options and to specify which counter line record types are to be printed. A format may have one and only one R1 transaction.

The R1 transaction is used to suppress standard headings.

It defines special spacing between headings.

It will also define the number of lines per page, if different than the standard number. The R1 transaction will specify which counter lines are to be printed.

**Cols 1-2 (alphanumeric) Transaction Code**

This is a constant, R1.

### **Cols 3–4 (alphanumeric) Format Code (165)**

Enter the same code as used in columns 3–4 of the R0 transaction.

### **Cols 5–7 (numeric) and 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77-80 of the transaction.

Duplicate numbers may appear in 5-7 as long as 77-80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

### **Col 8 (alphabetic) Print Header 1 (167)**

Y = Print header line 1 (default)

N = Do not print any headers—columns 9–11, 14–18 and 20–24 will be ignored.

### **Col 9 (alphabetic) Header 2 (168)**

Y = Print header line 2 (default)

N = Do not print

### **Col 10 (alphabetic) Header 3 (169)**

Y = Print header line 3 (default)

N = Do not print

### **Col 11 (alphanumeric) Print Header 4 (170)**

Y = Print header line 4 (default)

N = Do not print

9 = Print the word 'UNIT' and the unit number after 'UNIT' on the first detail line (for 941A report)

Tape, card, or disk outputs should not have header lines generated unless they are print image files.

### **Cols 12–13 (numeric) Maximum Lines per page (171)**

01–99 = The desired value. The default is defined in the master copy of the system in the COBOL program P5PRNT and may be changed, if desired, provided you have the capability of recompiling that program.

### **Col 14 (alphanumeric) Header 1 Spacing (172)**

The spacing before printing header line 1.

1 = Top of form (default)

# = Single space

0 = Double space

- = Triple space

**Col 15 (alphanumeric) Header 2 Spacing (173)**

The spacing before printing header line 2.

- 1 = Top of form
- # = Single space (default)
- 0 = Double space
- = Triple space

**Col 16 (alphanumeric) Header 3 Spacing (174)**

The spacing before printing header line 3.

- 1 = Top of form
- # = Single space
- 0 = Double space (default)
- = Triple space

**Col 17 (alphanumeric) Header 4 Spacing (175)**

The spacing before printing header line 4.

- # = Single space (default)
- 0 = Double space
- = Triple space

**Col 18 (alphanumeric) After Headers Spacing (176)**

The spacing after printing header line 4.

- # = Single space
- 0 = Double space (default)
- = Triple space
- 2-9 = Skip down to the line number specified in columns 48-63 of the R0 transaction.

**Col 19 (alphabetic) Generator File (177)**

- N = Payroll file payment format (use AG, AH pay stub messages)
- R = Recycle file
- Y = Payroll file format (default)
- C = Data in this format is used in the Combined Register

The 'N' option causes many of the fields in the R1 through R6 transactions to be treated differently.

**Col 20 (alphabetic) Print Control 1 and 2 (178)**

Y = Print Organization 1 and 2 (default)

N = Do not print

S = Same as Y, but also print the first 4 characters of the AD transaction description of Pay Levels 3–6 and 'EMPLOYEE' on the 3rd heading line.

The Pay Level 1 code and description (from the AA transaction) will appear on header line 1 in print positions 1–48.

The Pay Level 2 code and description (from the AA transaction) will appear on header line 2 in print positions 1–48.

This field also controls whether or not 'REPT' and the REPORT CODE are printed on header lines 1 and 2 in print positions 84–87.

**Col 21 (alphabetic) Print File Version Number (179)**

Y = Print File Version Number (default)

N = Do not print

The File Version Number prints on header line 1 in print positions 106–122.

**Col 22 (alphabetic) Print Page Number (180)**

Y = Print Page Number (default)

N = Do not print

The page number will appear on header line 1 in print position 123–132.

**Col 23 (numeric) Print Time and Date (181)**

0 = Print date and time (default)

1 = Print date only

2 = Suppress date and time

The run date appears on header line 2 in print positions 120–132. The run time appears on header line 2 in print positions 106–118.

**Col 24 (numeric) Print Period Ending Dates (182)**

0 = Print 1st and 2nd lowest frequencies (default)

1 = Print lowest frequency

2 = Print 2nd lowest frequency

3 = Suppress both dates

The Period Date from the AJ transaction with the lowest frequency is printed on header line 1 in print positions 97–104. The next lowest frequency (being paid) Period Date is printed in positions 97–104.

**Col 27 (alphabetic) Print Descriptive Lines (185)**

Y = Print descriptive lines defined by R3 and R4 transactions (default)

N = Do not print.

A descriptive line is produced from an extract record that has a record type other than '1' and 1 through 7. It may not contain figures that are to be totaled. It is formatted under control of the R3/R4 transaction.

**Col 28 (alphabetic) Print Counter Lines (186)**

Y = Print detail counter lines defined by R5 and R6 transactions (default)

N = Do not print detail lines but accumulate totals

T = Do not print detail lines, but print descriptive information from the last extract record on the total line.

A counter line is produced from an extract record that has a record type of 1 through 7. It may contain descriptive information and/or figures to be totaled. The figures to be totaled must follow the descriptive information in the extract record and be in counter (computational) form. A counter line is formatted under control of the R5/R6 transaction.

If the T option is used, see column 35 of the S1 transaction and the Field Length on the R5/R6 transaction.

**Cols 29, 35, 41, 47, 53, 59, 65 (alphabetic) Print Counter Lines (187)**

Y = Print this counter line

N = Do not print (default)

T = Print this counter line detail, but do not print total line

For column 41 only S = Subtract line 2 from line 1 giving line 3

(record type 3)                      A = Add lines 1 and 2 giving line 3

For column 47 only D = Divide line 2 by line 1 giving line 4

(record type 4)

These two options may be used for budget-to-actual comparisons showing amount difference and percentage variance.

Any records directed to line 3 or line 4 when the S, A, or D option is being used for that line will be ignored.

Column 29 indicates the action to be taken for extract records with a record type of 1, column 35 for a record type of 2, and so forth.

### **Cols 30–34; 36–40; 42–46; 48–52; 54–58; 60–64; 66–70 (alphanumeric) Counter Line Descriptions (188)**

Enter the description to print on this counter line. It may be left blank.

The first move on the R5/R6 transaction that has blanks in the Field Length will move this 5-character field to the output detail or total line.

Columns 30–34 will be moved for extract records with a record type of 1, columns 36–40 for a record type of 2, and so forth.

### **R2 format transaction (used in P5PRNT)**

The R2 transaction is used to determine the line and position of heading information.

The R2 transaction is used to supply constant information for any or all of the four heading lines. A format may have any number of R2 transactions.

### **Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, R2.

### **Cols 3–4 (alphanumeric) Format Code (165)**

Enter the same code as used in columns 3–4 of the R0 transaction.

### **Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77–80 of the transaction.

Duplicate numbers may appear in 5–7 as long as 77–80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

### **Col 8 (numeric) Header Number (192)**

1 = Header line 1

2 = Header line 2

3 = Header line 3

4 = Header line 4

Enter the line number of the heading line into that in which constant information is to be inserted.

### **Cols 9–11 (numeric) Header Print Position (193)**

001–132 = The print position in which the heading should start.

You may enter multiple R2 transactions for the same header line.

Remember not to overlay options (date, time, and so forth) for header lines 1 and 2 that were specified on the R1 transaction.

### **Cols 12–71 (alphanumeric) Heading**

Enter the information as it is to appear. (Only non-blank characters are moved. Blanks will not overlay data already in the header from R1 transaction options.)

**R3/R4 format transactions (used by P5PRNT)**

The R3/R4 transactions are used for non-computational moves to the extract record. A non-computational record is record type 0 and A–Z.

The R3 and R4 transactions each allow 13 moves. The R4 transaction is a continuation of the R3 transaction.

A format may have only one R3 and one R4 transaction.

The R3/R4 transactions are used to move data from extract records with a record type other than '.' (period) or 1 through 7 to the output record.

**Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, R3—or R4, if a continuation transaction.

**Cols 3–4 (alphanumeric) Format Code (165)**

Enter the same code as used in columns 3–4 of the R0 transaction.

**Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77–80 of the transaction.

Duplicate numbers may appear in 5–7 as long as 77–80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

**Cols 8–9; 13–14; 18–19; 23–24; 28–29; 33–34; 38–39; 43–44; 48–49; 53–54; 58–59; 63–64; 68–69 (numeric) Field Length (194)**

This represents the length of the field to be moved from the extract record to the output record. Valid lengths are 01–99.

**Cols 10–12; 15–17; 20–22; 25–27; 30–32; 35–37; 40–42; 45–47; 50–52; 55–57; 60–62; 65–67; 70–72 (numeric) Print Position (195)**

This is the starting print position where the field is to be moved.

For non-print data output, the transaction column is 1 more than the print position. For example, to move to transaction column 5, specify print position 4.

*Note:* For special carriage control on printed output, any of the following may be moved to print position 000.

*blank* = Single space (default)  
*0* = Double space  
*-* = Triple space  
*1–9* = Channel 1–9

*Note:* This note is applicable only to the last slot (13th move, columns 70–72) of the R4 transaction only. If the 12th move (columns 63–67) is blank, then the print position of the 13th move can be used to give the print position for the system supplied page number and the Field Length is used as follows:

*bb* = Print page number on header line 1  
*02* = Print page number on header line 2

*03 = Print page number on header line 3*

*04 = Print page number on header line 4*

*01 = Print page number on header line 1 and begin numbering the pages at page number 2 (941A report option).*

*05–09 = Begin page number with this number*

To create multiple print lines from a single report record (mailing labels, address lists, time sheets, and so forth), make a series of moves on the R3 or R4 transaction. Then leave the next Field Length and print position blank and continue with the next set of moves.

The blanks, followed by more moves, will cause the print program to begin the next moves on a new line. This option may be used anywhere on the R3 or R4 transaction except for the last two slots of the R4 (see the option above). Data may be moved to print position zero to control the spacing of the next line.

The moves are performed in order (working from left to right across the R3 transaction and continuing across the R4 transaction). Data is moved from the extract record beginning with the position following the record type and continuing across the extract record.

### **R5/R6 format transactions (used by P5PRNT)**

The R5/R6 transactions are used to move computational record data from extract records with a record type of 1 through 7 to the output record. A computational record is a record type of 1–7.

There is room for 13 moves on the R5 transaction and 13 more on the R6 transaction. A format may have one and only one R5 transaction and one and only one R6 transaction.

Descriptive data may also be moved on the R5/R6 transactions. Counters and computational fields may be moved only after all descriptive data has been moved.

### **Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, R5 or R6 if a continuation transaction.

### **Cols 3–4 (alphanumeric) Format Code (165)**

Enter the same code as used in columns 3–4 of the R0 transaction.

### **Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77-80 of the transaction.

Duplicate numbers may appear in 5-7 as long as 77-80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

### **Cols 8– 9; 13–14; 18–19; 23–24; 28–29; 33–34; 38–39; 43–44; 48–49; 53–54; 58–59; 63–64; 68–69 (numeric) Field Length (194)**

The contents of the Field Length field specify the editing for the counter. See the table that follows.

**Cols 10–12; 15–17; 20–22; 25–27; 30–32; 35–37; 40–42; 45–47; 50–52; 55–57; 60–62; 65–67; 70–72 (numeric) Print Position (195)**

This is the starting position of the print line to which either the field or the counter is to be moved.

For non-print data output, the print position represents a displacement in the output record. For example, the print position 10 would be 11 characters into the record.

Descriptive information may be moved using the R5 and R6 transactions just as it was moved on the R3 and R4 transactions.

All counter fields must follow any descriptive fields in the extract record. Once all the descriptive information (name, department code, job code, and so forth) has been moved, the counters may be moved.

To signify the beginning of the counter moves, leave the Field Length blank and specify the print position for printing the Counter Line Description (188) from the R1 transaction. (The move must be specified even if the Counter Line Description is blank.)

Field 188 must be moved to a print position greater than 18, plus the size of the field causing the break or else the totals will be printed on the line following the one containing the Total Line Description.

After the moves of the descriptive data, only counters may be moved. The contents of the Field Length field specifies the editing for the counter as follows:

Code	Format	Comments
Blank	-,---,---.99	
01	-,---,---.---	
02	-,---,---.99	Print totals only
03	-----V99	
04	-----V99	Blank when zero
05	-----V99	Print totals only
06	---,---.9999	
07	---,---.9999	Print totals only; divide the next counter by the one following it, placing the result into this counter. Two decimals divided by two decimals gives four decimals.
08	-,---,---.99	Print totals only; move to higher level counter instead of adding to higher-level counter.
09	-,---,---.99	Print totals only; multiply +1 counter (V9999) by +2 counter (V99) giving this counter (V99).
10	-----.99	
11	---,---,---	
12	---,---,---	Print totals only

*Note: If the 'T' option has been specified in column 28 of the R1 transaction and column 35 of the S1 transaction contains a zero or space, then the Field Length number is the length of the counter to be edited to the output file. Lengths 01–12 are valid.*

### **R7 format transaction (used by P4CALC)**

R7 transactions are used to supply the logic portion of a Report Generator. Every Report Generator must have at least one R7 transaction.

#### **Cols 1–2 (alphanumeric) Transaction Code**

This is a constant, R7.

#### **Cols 3–4 (alphanumeric) Format Code (165)**

Enter the same code as used in columns 3–4 of the R0 transaction.

#### **Cols 5–7 (numeric) and cols 77–80 (alphanumeric) Sequence Number (166)**

This field is used to sequence the format. This field is continued in columns 77-80 of the transaction.

Duplicate numbers may appear in 5-7 as long as 77-80 have ascending values. Each transaction must have a Sequence Field that is equal to or greater than the previous one.

#### **Cols 8–10 (alphanumeric) Instruction Mnemonic (197)**

Enter an instruction mnemonic. A description of each of the valid instructions may be found later in this section.

If '\*' (an asterisk followed by two spaces) is entered here, columns 11–76 are ignored (that is, this is a comment transaction).

#### **Cols 11–42 (alphanumeric) Operand Field**

Enter the instruction's operand (if one is required) left justified. Comments may be entered in this field to the right of an operand.

### **R7/S7 coding considerations**

Before describing the instructions that may be coded on the R7 and S7 transactions, a discussion of the basic purpose of these instructions is in order.

The primary purpose of the R7 and S7 transactions is to build an extract record containing the necessary data to sequence and produce the desired output file. This extract record is always 150 bytes (characters) long, and the positions are numbered from 0 to 149.

The sequencing information (sort key) must precede the other data within the extract record.

A one character Record Type always follows immediately after the Sort Key.

## Extract Record



- An extract record consists of 2 major parts, the sort key and report data.
- The extract record is created by S7 and R7 instructions.
- The length of each portion varies by generator.
- Total length is a maximum of 150 characters.

Generally, the S7 transactions contain instructions to build the sort key portion of the extract record. The R7 transactions contain instructions to build the remainder of the record.

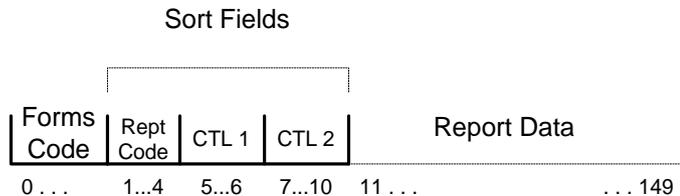
The portion of the extract record containing the sort key is variable in length. For example, different Report Generators may have different sort key lengths.

Examining the sort key more closely shows that there are several fields that are normally present at the beginning of the key.

The first position of the key must be a valid Forms Code. The Forms Code is used by P5PRNT to determine which of its output files is to be used.

Following the Forms Code are normally the Report Code (field 151), Pay Level 1 (field 001), and Pay Level 2 (field 002). You define the remainder of the key.

## Extract Record Sort



- Created by S7 instructions in P4CALC
- Sort length varies by generator. Length is determined by the last S1 transaction and should not exceed 60 characters.
- Forms code denotes the output file:

Forms Code	Description
0	Printer—reserved for Sort01
1	Print file
2	Print file
3	Print file

Forms Code	Description
4	Print file
5	80 character output file
6	80 character output file
7	133 character output file with carriage return
8	Recycle file
9	Reserved for Sort22
P	Canadian Direct Deposit Tape
Q	Canadian T4/TP4 Tape
R	W-2 Tape
S	ACH tape
T	Printer—checks file
U	Printer—checks file
V	Printer—deposit statements
W	Printer—checks file
X	P51W2T—NY State Quarterly Tape
Z	Force report print to second pass of P5PRNT
\$	Service Bureau Statistics File
A–K	Optional—User defined output files

- Sort fields contain data for the report sort.

### Sort key

Examining the other portion of the extract record, the first position must be a valid record type.

The record type is used by P5PRNT to determine whether the R3/R4 transactions or R5/R6 transactions are to be used. A record type in the range of 1 to 7 signifies the R5/R6 transactions.

The record type need not be the same for every extract record for a given Report Generator.

### Data required to produce output file

Another valid record type is a period (.).

A period in the record type signals P5PRNT that this record is to be used to modify one of the four standard heading lines. The record is not written to any of P5PRNT's output files. Also, certain additional pieces of information are required:

1. The number of the print line (1, 2, 3, or 4) to be modified must follow the record type.
2. The beginning print position must follow that.
3. The actual data to be inserted into the heading line must follow that.
4. Two periods must follow the data.

**Period record**

Upon encountering a period record, P5PRNT will move the insert data to the specified heading line beginning at the indicated print position.

The modified heading will appear on each subsequent page of the report or until it is modified again.

Any number of period records may be used to modify the standard headings.

*Note: This method can only be employed when it is necessary to have variable data in the report headings. Constant data is more easily inserted into heading lines using R2 transactions.*

A record type of a period may also be used to modify an S1 transaction on the fly.

To use this option, follow the period with the letter 'S', a number in the range 3–9 indicating which S1 transaction is being changed, columns 11–14 of the S1 transaction and columns 17–38 of the S1 transaction.

The first two S1 transactions may not be altered.

The Field Length (columns 13–14) must be in the range of 01 though 30.

The Key Position, if used, must be in the range of 10 through 64.

Neither the total number of S1 transactions nor the Total Sort Key Length may be altered.

**Extract Record**

	Record Type	Carriage Control	Descriptive Data	Computational Fields
0...				... 149

- Created by the R7 instructions in P4CALC
- Record Type

Type	Heading record
0, A–Z	Record contains only descriptive data formatted by the R3/R4 transactions
1–7	Record contains descriptive data and/or computational fields to be totaled by P5PRNT formatted by the R5/R6 transactions

- Carriage Control—Present only for printed output

1	TOP OF PAGE
blank	SINGLE SPACE
0	DOUBLE SPACE
-	TRIPLE SPACE

- **Descriptive Data**  
Alphanumeric data, which will not be totaled during this generator (for example, name, address, state exemptions)
- **Computational Fields**  
Contain numeric data, stored in binary format, to be totaled by P5PRNT. These must be the last fields in the extract record. The most often used are fields 117–124, 149, 150, and counters. Check the field number table for additional computational fields.

### Extract record work area

The 150-character extract record is also used by P4CALC as a work area.

The RST instruction resets the output position pointer to the left by the number of positions contained in the current field size.

Numerous report generator instructions automatically reset the output position pointer in the extract record.

Review all of the instructions in the report generator section of your manual and mark the alphabetic index of instructions so that you have a ready reference of all generator instructions that perform an automatic reset.

Some instructions automatically reset the output position pointer as follows:

Position	Mnemonic	Instruction
0	WRT	Write extract record
0	SPC	Blank out line
0	BSK	Build sort key
1st Position After Sort Key	DSK	Dummy sort key

### Extract record work area manipulation

#### Sample Generators

CMLxx	The value of (xx) is placed in the current field size. The extract record is unchanged. xx = 01–30
B01xx	The value (xx) is added to the output position pointer and replaces the current field size. The data in the extract record is unchanged. xx = 01–30
CHRxx	Both the output position pointer and the current field size are reduced by the value (xx). xx = 01–30
CHLxx	The value (xx) is chopped off the last field moved to the extract record by shifting the rest of the field to the left. Both the output position pointer and the current field size are reduced by this value. xx = 01–30

**Logic instructions**

Logic instructions on S7 or R7 transactions describe what P4CALC interprets to build the extract record.

Below follows a list of important things to note:

- No input file definition is required; the instructions work on the Sequential Master File.
- No read instruction is required; the control is automatically passed to the generator based on generator criteria.
- Over 100 instructions are available, but only a few are necessary to do a simple listing.
- The output position pointer determines the position in the extract record of the left-most position of the receiving field. Its value is in the range of 0 to 149.
- The current field size normally contains the length of the last field moved to the extract record.

**Instruction Set Sample Generators**

FLDxxx	Move a Sequential Master File field to the extract record, where xxx is the field number.
LITxyyy	Move a literal to the extract record, where xx is the literal length (not to exceed 53 characters) and yyy is the literal value.
AX1	Set all indexes to their first occurrence.
BSK	Perform the S7 instructions to build the sort key portion for the extract record.
DSK	Dummy sort sets the output position pointer to the total sort key length.
SFCx	Move Forms Code, Report Code, Pay Level 1 and Pay Level 2 to the extract record. x = Forms Code
WRT	Write the extract record to the P40OUT file
XIT	Exit from this generator
ZAC	Zero counters 02–34

**Generator instructions - Unconditional branching**

There are ten unconditional branch instructions, five to branch up and five to branch down to each of the five paragraph names.

The branch instructions are written in the operation field; the operand field is not used.

<b>Instruction</b>	<b>Action</b>
BU1 (079)	Branch up to paragraph name 251
BU2 (081)	Branch up to paragraph name 252
BU3 (053)	Branch up to paragraph name 253
BU4 (028)	Branch up to paragraph name 254
BU5 (026)	Branch up to paragraph name 255
BD1 (078)	Branch down to paragraph name 251
BD2 (080)	Branch down to paragraph name 252
BD3 (052)	Branch down to paragraph name 253
BD4 (027)	Branch down to paragraph name 254
BD5 (025)	Branch down to paragraph name 255

In each case, control is transferred to the first specified paragraph name encountered, either up or down, depending on the instruction, from the branch instruction.

An error message will be printed on the Payroll Audit Trail when the report is installed if a branch down instruction can not be resolved (the specified paragraph name does not exist later in the generator).

Each generator is assumed to begin with all five paragraph names; a branch up will never be unresolved.

## Efficiency Considerations

### Subroutines

- Ensure you are using the latest software version for your machine.
- Non-IBM users should have their system personnel review the subroutines for possible improvement.

### Combined extract reporting

- Generators must be consecutively numbered.
- Each must have the same sort key length.
- Application must be one output record.
- Generator with logic should always be active.
- Data not needed on subsequent reports can be removed by the R3/R6 transactions.

### Generator instructions

Existing generators should be modified to take advantage of improved instructions that may not have been available when they were originally coded.

#### **Inefficient generator instructions**

##### **CHL—Chop Left; possible alternatives are:**

- Displacement—FLD094L03D01. This instruction would result in a move of positions 2–4 of Field 094.
- Overlaying—Leave the data on the extract record and overlay with subsequent R3/R6 moves.

##### **CHR—Chop Right; possible alternatives are:**

- Length modifier—FLD094L03. This instruction would result in a move of the first 3 positions of Field 094.
- Overlaying with R3/R6 moves.

##### **RST—Reset Output Position Pointer by the value in the Current Field Size**

Do not be too concerned with the location of the Output Position Pointer if a Build Sort Key (BSK) instruction is coming soon in the generator logic, unless the generator is looping, or if the selection criteria contain many variables.

##### **SPC—Space Out Record**

Normally not necessary to space out an entire extract record. This instruction causes the system to start spacing at position 149 and spacing back one position at a time to 000. Alternative to SPC would be to move blocks of literals.

##### **ZAC—Zero All Counters**

Zeros counters 02–34. If no counters are used in the generator, it is not necessary to zero counters, or if using counters 02–09 only, the following Z29 instruction is more efficient.

### **Efficient generator instructions**

#### **Z29—Zero Counters 02–09**

If the application does not require a large number of counters, use counters 02–09 and this instruction instead of ZAC.

#### **SNC—Set to Next Counter**

More efficient than SET instruction

#### **SLC—Set to Previous Counter**

More efficient than SET instruction

#### **SFC—Sort Forms Code**

More efficient than the individual S7 moves for Forms Code, Report Code, and Organization 1-2.

#### **Length Modifier**

Move as many positions as possible with one instruction. FLD094L16 is much more efficient than the individual moves for Fields 094–097.

#### **MEM—Memory; STO—Store; RCL—Recall**

These instructions should be used when:

- the same data is to be in multiple Extract Records.
- data needs to be passed from the R7 to S7 logic.
- data needs to be passed from one generator to another.

#### **B02xx—Bump the Output Position Pointer xx (01–30) positions**

Data previously moved to the extract record does not need to be moved again when a reset condition would have caused it to be lost.

#### **DSK—Dummy Sort Key**

Sets the Output Position Pointer to the first position in the extract record after the sort key length. Saves rebuilding the Sort Key with BSK.

### **Other considerations**

- Exit from a generator as quickly as possible when selection criteria determine that a record is not to be processed further by this generator.
- Generate as few extract records as possible. If the report has multiple lines for an employee, try to put multiple output lines in one extract record.
- Keep generators as short as possible; each instruction takes time.

## Practice

Write a generator (T1T1) that will produce one line of output per employee and list the following:

Field Name	Field No.
1. Social Security Number	070
2. Employee Name	087
3. Union Code	076
4. Employment Status	073
5. Birth Date	078
6. Job Category	085
7. Employment Date	080

The sort requirement is:

Social Security Number	070
------------------------	-----

The output positions are:

Field Name	Position
1. Social Security	15
2. Employee Name	29
3. Union Code	61
4. Employment Status	68
5. Birth Date MM/DD/YY	72
6. Job Category	82
7. Employment Date MM/DD/YY	88

While writing T1T1, draw out the Extract Record.



PART 2

## Using Report Generator

---

### In This Section

Generator Maintenance.....	53
Writing Logic .....	63
Counters, Arithmetic, and Conditional Branching.....	71
Method Codes.....	79
Printing .....	89
Debugging .....	95



CHAPTER 3

# Generator Maintenance

---

## In This Chapter

Generator maintenance .....54

## Generator maintenance

- The generator must pass all edits for maintenance to be performed.
- Generators must be in a batch with an Organization 1-2 value of 999999.
- Maintain your source documents. Generators are kept on the Sequential Master File in object form only; they are not system retrievable.
- A generator must be re-entered in its entirety to update an existing version.

### Payroll Audit Trail sample showing loaded generators

R00101		O 0	LOADED	PAYROLL AUDIT TRAIL	03/27/98 11:48:31
R00103		CT O	LOADED	CONTROL HEADERS	03/27/98 11:48:31
R00202		7	LOADED	MASTER FILE PRINT	03/27/98 11:48:31
R02222			LOADED	COMBINED REGISTER	03/27/98 11:48:31
R02B2B		7C	LOADED	FLAG ACTIVE J'S	03/24/98 18:50:54
R02H2H 1234		C	LOADED	HED'S-COMBINED REGISTER	03/27/98 11:48:31
R02T2T 1234		C	LOADED	TAXES-COMBINED REGISTER	03/27/98 11:48:31
R05959 1234		O	LOADED	BOND BALANCE REGISTER	09/07/98 20:15:08
R05G51		0	LOADED	FIPS POST. CODES v115	03/27/98 11:48:31
R05G8A			LOADED	ACCUMULATION M/C	09/07/98 16:13:59
R05H5Z		4	NOT LOADED	ON-LINE CSSS ROOT	03/27/98 11:48:31

## Loading and selecting generators

Loading is the process of reading a generator from the Sequential Master File and storing it in the P4CALC working storage area called Requests. Once loaded, a generator may be executed by any company on the Sequential Master File.

If an output Sequential Master File is created, the generator may be saved on it for use during subsequent runs. Generators reside on the Sequential Master File and precede all company and employee data.

One of the first tasks performed by the program P4CALC is to read any newly entered generators from its transaction file and any generators stored on the Sequential Master File.

P4CALC decides which generators will be available for use during the current run by analyzing the 'Run Selects' code in the H2 transaction. The H2 transaction is an optional transaction entered directly into P4CALC.

A generator is available for use if:

- the Run Select code in its R0 transaction is a '0'.
- the Run Select codes in the R0 and H2 transactions are equal.
- the Run Select code in the R0 transaction is blank and no H2 transaction was used or the H2 transaction Run Select code was also blank.

Every generator that is available for use is listed on the Payroll Audit Trail, followed by the word 'Loaded'. Generators that are not available for use are also listed, but followed by the words 'Not Loaded'.

When a generator is loaded, P4CALC copies the S1, R0, R1, R2 R3, R4, R5, and R6 information to the extract file (P40OUT) for use by the program P5PRNT. The R0, S7, and R7 information remains in memory during the entire execution of P4CALC.

Once loaded, a generator may be selected by any or all of the companies (Organizations/Control 1-2's) on the Sequential Master File. To select a generator, a D transaction must be entered (or have been entered during a previous run) behind a batch transaction for the Organization 1-2.

Online users use the Report Requests form (DD-SCR) to create these batch transactions.

A list of all of the D transactions that have been entered for an Organization 1-2 will be printed on the Payroll Audit Trail. Each D transaction will have either 'Selected' or 'Not Selected' printed after it.

Only selected generators may be executed. A generator may be selected by a variety of methods. In each case, column 7 of the D transaction is compared to entries in an AE transaction or a P4 transaction.

A P4 transaction is an optional transaction that may be entered directly into P4CALC and affects all Organization 1-2s during the run. If column 7 of the D transaction matches any of columns 3–80 of the P4 transaction, the report will be selected.

If no P4 transaction is present or columns 3–18 of the P4 transaction are blank, the report will be selected only if column 7 of the D transaction is numeric and less than or equal to column 4 of the AE transaction.

The generator will also be selected during a pay run if column 7 of the D transaction matches any of columns 16–27 of the AE transaction.

Reports to be produced in a normal pay run should be created with a blank in the Run Select code. Reports to be run during a special run should be grouped with a common indicator in the Run Select code.

## **Adding, replacing, and deleting report generators**

### **Adding Report Generators**

To add a Report Generator to the Sequential Master File, enter the sort and format behind a batch transaction with '999999' in the Organization 1-2 field. The sort must precede the Format.

Within the sort, the S1 transactions must precede the S7 transactions. Within the Report Format, the transactions must be ordered as follows:

R0  
R1  
R2  
R3  
R4  
R5  
R6  
R7

If three formats, each using the same sort, are to be added to the Sequential Master File, they should be entered as follows:

SORT

FORMAT #1

FORMAT #2

FORMAT #3

If, on the other hand, one format is to be added to the Sequential Master File with three different sorts, they must be entered as follows:

SORT #1

FORMAT

SORT #2

FORMAT

SORT #3

FORMAT

If the generator being added can use one of the sorts already on file, the name of the sort can be specified on the R0 transaction, and the S1 through S7 transactions can be omitted entirely.

### **Replacing Report Generators**

To replace a Report Generator on the Sequential Master File, enter the sort and format behind a batch transaction with '999999' in the Organization 1-2 field. The transactions must be in the same sequence as specified in the instructions for adding a report generator.

A Report Generator cannot be changed on the Sequential Master File; it can only be replaced. The complete sort and/or format must be entered. If a format is entered without its sort, the Sort Code must be placed in the format's R0 transaction.

If, however, a format is to be changed that was previously paired with three sorts, the sort name must be specified in the R0 transaction of each copy of the generator.

### **Deleting Report Generators**

To delete a Report Generator from the Sequential Master File, enter only an S1 transaction for the sort and only an R0 transaction for the Report format behind a batch transaction with '999999' in the Organization 1-2 field. Do not enter S7, R1, R2, R3, R4, R5, R6, or R7 transactions.

If, however, the sort was previously paired with 3 formats and only one of the formats is to be deleted, the sort should be entered in its entirety with the format, minus the R7 transactions, following it.

## Report Selection

### D transaction

The D transaction is necessary to schedule the selection of a generator for execution. The D transaction must be entered for each company on the Sequential Master File that requires this generator be run. Scheduling the selection of the report generator is established by the select code in column 7 of the D transaction.

Select Code	Definition
0	Maintenance run
1	Pay run
2	End of the month
3	End of the quarter
4	End of the year
ALPHA	Special request only

The Data Select area establishes the type of data that this generator will access, thereby determining when control should be passed to this generator.

### Converting D transactions to online form entries

The D transactions discussed in this document can be entered online using the Report Requests form (DD-SCR).

Following is an example D transaction and the matching entry for the Report Requests form (DD-SCR).

7	8	1	1	2	2	3	3	4	4	5	5	6	6	7
1234567890		5	0	5	0	5	0	5	0	5	0	5	0	5
5	0													
D	aaaabodeefghijklmnnno													

	Position	/ST Form Fields
D	1	--
aaaa	3–6	Report Code
b	7	Report Select
c	8	Adjustments—Plus
d	9	Adjustments—Minus
e	10	Adjustments—Manual
f	11	To-date Amounts—Current
g	12	To-date Amounts—Month
h	13	To-date Amounts—Quarter
i	14	To-date Amounts—Year

	Position	/ST Form Fields
j	15	Data Types—Company Level
k	16	Data Types—Tax Tables
l	17	Data Types—Labor Record
m	18	Data Types—Other Record
nnnn	19–22	User Field
o	23	Extra Copy

### 3.2 Report Requests form (DD-SCR) entry

The screenshot shows the 'Report Requests' dialog box. The title bar reads 'The Solution Series/ST'. The menu bar includes 'File', 'Edit', 'HR', 'Payroll', 'Benefits', 'IA', 'Tools', 'Actions', 'Fonts', and 'Help'. The toolbar contains icons for file operations and help. The dialog is titled 'Report Requests' and contains the following fields and sections:

- Report Code:** 0101
- Report Select:** 0
- User Field:** (empty text box)
- Extra Copy:** (dropdown menu)
- Adjustments:**
  - Plus: Do Not Print (dropdown)
  - Minus: Do Not Print (dropdown)
  - Manual: Do Not Print (dropdown)
- Data Types:**
  - Company Level: Do Not Print (dropdown)
  - Tax Tables: Do Not Print (dropdown)
  - Labor Record: Do Not Print (dropdown)
  - Other Record: Do Not Print (dropdown)
- To-date Amounts:**
  - Current: Do Not Print (dropdown)
  - Month: Do Not Print (dropdown)
  - Quarter: Do Not Print (dropdown)
  - Year: Do Not Print (dropdown)

Buttons for 'OK' and 'Cancel' are located at the bottom of the dialog. The status bar at the bottom of the window displays '999999 DD-SCR'.

**5.x Report Requests form (DD-SCR) entry**

**AE transaction**

The AE transaction is that transaction that determines whether a particular pay frequency is to be paid in a given execution of the system. It may be entered either in card image batch format or by the Payroll Run Process Control (AE-SCR) form.

Generators will be selected for execution if column 7 of the D transaction is one of the following:

- Equal to or less than column 4 (Reporting Type field) of the AE transaction. For example, an entry of '3' in the D transaction selects report generators with codes of 0, 1, and 2, as well as 3.
- Equal to a special select code entered in columns 16–27 of a pay run AE transaction. The Report Select field is ignored in a non-pay/maintenance run.

The AE transactions discussed in this document can be entered online using the Payroll Run Process Control form (AE-SCR). Following is an example AE transaction and the matching entry for the Payroll Run Process Control form (AE-SCR).

	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8																										
1	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0	...	5	...	0													
A	e	a	b	c	d	d	d	d	d	e	f	g	h	h	h	h	h	h	h	h	i	i	i	i	i	j	j	j	j	k	k	k	l	m	m	m	m	n	n	o	p

	Position	Solution Series Form Fields
AE	2	--
a	3	Clear To-date
b	4	Reporting Type
c	5	Purge Rule
d	6-11	Run Date
e	12	Run Type
f	13-14	Version Number
g	15	Print Update
h	16-27	Report Select
i	28-33	User Date
j	34-37	User Field
k	38-40	--
l	41	Frequency
m	42-47	Payment Date
n	48	New Period
o	49	Pay Cycle
p	50	Deduction Cycle

### 3.2 Payroll Run Process Control form (AE-SCR) entry

The screenshot shows the 'Payroll Run Process Control' form. The title bar reads 'The Solution Series/ST'. The menu bar includes 'File', 'Edit', 'HB', 'Payroll', 'Benefits', 'JA', 'Tools', 'Actions', 'Forms', and 'Help'. The toolbar contains various icons for navigation and actions. The form fields are as follows:

- Clear To-date:
- Reporting Type:
- Purge Rule:
- Print Update:
- Version Number:
- Run Type:
- Run Date:
- Report Select:
- User Date:
- User Field:

Below these fields is a section titled 'Define Frequencies to be Paid' with the following options:

- Frequency:  WEEKLY
- New Period:  Yes  No
- Payment Date:
- Pay Cycle:
- Deduction Cycle:

At the bottom of the form are buttons for 'Frequency Selection', 'OK', and 'Cancel'. The status bar at the very bottom shows '999999 AE-SCR'.

### 5.x Payroll Run Process Control form (AE-SCR) entry

### P4 transaction

The P4 transaction is entered into P4CALC directly in the jobstream.

- Codes entered in positions 3–18 will override report generators requested on the AE transaction.
- Codes entered in positions 19–34 will execute report generators in addition to those requested on the AE transaction.

## Passing Control

Multiple selections on the R0 transaction and the D transaction can result in multiple passes through the generator logic.

The R0 transaction determines which data may cause a generator to receive control. But it is the D transaction that determines which data is actually made available to the Report Generator. Therefore, the D transaction has the final control.

Once control is received by a generator, control is passed to the first R7 instruction in the generator.

Report select—field 152—can be interrogated to determine which pass is currently being processed (column 7 in the D transaction). At any given time, the value of field 152 will be equal to the value in the R0 transaction that caused the pass that is occurring at that time.

For example, if the pass now occurring is the pass of 'Other' records caused by there being an 'O' in position 15 of the R0 transaction, then the value of field 152 will be 'O'.

The initial value of To-Date indexes will initially be set at Current if Current pass; if Month-to-Date pass, indexes are initially set at Month-to-Date, and so forth.

It should be noted, however, that indexes may nevertheless be set however desired by the program code.



CHAPTER 4

# Writing Logic

---

## In This Chapter

Introduction .....	64
Practice .....	69

## Introduction

As do most programming languages, the Report Generator allows for logic to determine selection of data to be processed, paths through the program to be taken depending upon comparisons, any so forth. Available paragraph names and branch instructions are described below.

### Paragraph names

Points in the generator logic are similar to COBOL paragraph names. Branching up or down to these points allows the flow of the generator to be controlled.

There are five paragraph names available. Each has two associated branching instructions:

Paragraph Name	Branch Down to	Branch Up to
251	BD1	BU1
252	BD2	BU2
253	BD3	BU3
254	BD4	BU4
255	BD5	BU5

The branch is always to the next occurrence of the paragraph specified.

*Note: Paragraphs 254 and 255 each have an additional use. The system will automatically branch to the next 255 paragraph based on the results of tests performed by certain instructions. The special use of paragraph 254 will be discussed elsewhere in this course.*

### Compares fields/literals

The Compares use the extract record as a work area.

It will compare the next to last field (referred to below as 'first field') in the Extract Record to the last field (referred to below as 'second field') in the Extract Record.

The current field size determines the length of the compare.

A true condition passes control to the next 255 paragraph. A false condition passes control to the next instruction.

The Compares will automatically reset the last field moved to the extract record.

### Sample instructions

Instruction	Description
BE	If first field is equal to second field
BL	If first field is less than second field
BG	If first field is greater than second field
BNL	If first field is not less than second field
BNE	If first field is not equal to second field
BNG	If first field is not greater than second field

**To-date index**

The To-Date index exists exclusively for the H and J segments.

It is initially set by the D transaction in columns 11–14. The initial setting can be changed with the following instructions in the R7 transactions.

H Segment	J Segment
SHC = Current	SJC = Current
SHM = Month-To-Date	SJM = Month-To-Date
SHQ = Qtr-To-Date	SJQ = Qtr-To-Date
SHY = Year-to-Date	SJY = Year-to-Date

Employee Level HEDs																To-Date Fields								Field Numbers	
HED Number	Frequency	Type Code	Method Code	Amt. Or Pct.	Start		Stop		One Time Amount		Amount One	Amount Two	User		Current		Month		Quarter		Year				
					Code	Field	Code	Field	Code	Amt.			Code	Field	Amt.	Hrs.	Amt.	Hrs.	Amt.	Hrs.	Amt.	Hrs.			
130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	149	150	149	150	149	150	149	150	1st		
																							2nd		
																							3rd		
																							4th		
																							5th		
																							6th		
																							7th		
																							8th		
																							9th		
																							10th		
																							11th		
																							12th		
																							13th		
																							14th		
																							15th		
																							16th		
																							17th		
																							18th		
																							19th		
															Cur										
															1	MTD	QTD	YTD							
																2	3	4							
																<b>To-Date Indexes</b>									

Stack Index

### Segment index

Each company/organization record consists of multiple segments of data.

Certain types of data may have several segments (for example, Earnings/Deductions and Taxing Authorities.)

Similarly, each employee record consists of multiple segments, some of which may have multiple occurrences.

These multiple occurring segments are accessed by way of indexing. The Report Generator language provides for that access through several indexing instructions.

Several instructions exist whose purpose is to allow manipulation of segment indexes.

Some of these follow:

AX1	Set the index for all data segments to the first occurrence and all to-date indexes to current.
1STx	Set to the first occurrence of 'x' segment
NXTx	Increment the index to look at the next occurrence of the 'x' segment
PRVx	Decrement the index to look at the previous occurrence of the 'x' segment.

where x is the segment being indexed.

Employee Level	Company Level
E—General data	B—A8s
F—Name, Address	C—Company area
G—Controls/Pay Levels	T—Tax bodies
H—HEDs	W—'Other' records
J—Employee tax records	
L—Personnel/user defined	

**Select instructions**

<b>SELx</b>	<b>Searches the segment type index specified attempting to match a specific occurrence, where x may be:</b>	
	B	Company level (A8s) Earnings/Deductions
	C	Company level (AF, AG, AH, AJ, AK, DG)
	W	Other records (WA, WC, WH, WL)
	T	Company level tax records
	E	Employee basic data
	F	Employee name, address (F1, F2)
	G	Employee control levels
	H	Employee earnings and deductions
	J	Employee tax records
	L	Personnel/user defined records

Search argument and length are obtained from the last field moved to the extract record. The length of the search argument may be varied.

The Index will automatically reset the last field in the extract record. A find condition passes control to the next instruction. No find conditions will pass control to the next 255 paragraph.

*Note:* This is different from a 'true' condition in a branch instruction.

The search begins with the current location, not the first occurrence.

**Special usage of SEL for company/organization level HED and TAX segments**  
**SELB**

Uses the employee HED number as the search argument. For example, if the current employee HED index is 003 and a SELB instruction is issued, then the search will be for Company HED record 003.

This does not alter the Extract Record.

**SELT**

Uses the employee tax body as the search argument. For example, if the current employee tax index is 2IL and a SELT instruction is issued, then the search will be for company Tax record 2IL.

This does not alter the Extract Record.

**Work and save indexes**

Each segment has two indexes associated with it, a work index and a save index.

The work index is used to point to the record currently being accessed by the generator in a given stack of records.

The save index can be used to retain a work index for future reference in the generator.

### **AX1**

This instruction sets both the work and save indexes to the first occurrence for all segments as well as all to-date indexes to Current.

### **EXCx**

Swaps the pointer value in the save index and the pointer value in the work index.

### **Report with totals**

Each extract record written must contain a one character Record Type immediately following the sort key of the record.

The Record Type serves to determine whether the Output Report Record or Output File Record will be formatted by R3/R4 transactions or by R5/R6 transactions.

For a report to have column totals, the R5/R6 formatting must be used.

Record types 1–7 must be formatted by R5/R6 transactions.

P5PRNT will use the R5 and R6 cards to move the extract record data to the print line.

Each record type present must be validated on the R1 card: columns 29, 35, 41, 47, 53, 59, and 65.

Computational fields must be the last items on the Extract Record, though it is not necessary that they follow all descriptive data on the printed report.

### **Record types 1–7**

Record Types 1–7 are interpreted and formatted by P5PRNT based upon the R5/R6 record provided in the Report Generator. As previously discussed, the Record Type is a one-character indicator that must exist as the next character after the Sort Field in the Report Generator Extract Record.

### **R5 moves**

The sort key must be unique for each extract record processed to force the R5 descriptive moves to be performed.

The first blank move length signifies the beginning of the counter field moves. Also, the first blank move length moves the counter description from the R1 transaction (determined by the record type in the extract record) to the print line.

Subsequent moves transfer counter data from the extract record to the print line.

R1 counter description is often overlaid by counter data when only one record type is used.

## Practice

Write a generator (T2T2) that runs once a month and lists the following for hourly (Field 072=1 or 4) employees:

Field Name	Field No.
1. Employee Number	003
2. Social Security Number	070
3. Employee Name (First, Middle, Last)	087
4. Hourly Rate Edit 4 decimals	134
5. Normal Hours Edit 2 decimals	141
6. YTD Regular Earnings	149

Sort Requirements	
1. Department—major	094
2. Employee Number	003

Output	
Field Name	Position
1. Employee Number	04
2. Employee Name	20
3. Social Security Number	55
4. Hourly Rate	70
5. Normal Hours	80
6. YTD Earnings	90

Total the YTD earnings by Department Field 149

Complete the D transaction.

While writing T2T2, draw out the Extract Record.



CHAPTER 5

# Counters, Arithmetic, and Conditional Branching

---

## In This Chapter

Introduction .....	72
Practice .....	78

## Introduction

In this chapter we will explore how the Report Generator allows us to store numeric data, perform arithmetic functions, and transfer control within a program based upon conditional branch instructions.

## Counters

Counters are the equivalent of working storage in a COBOL program. Counters are work fields, with a maximum entry length of 9 positions, that are set aside for use by report generators.

A total of 63 counters are available. They are referenced and numbered 02 through 64.

The counters 02 through 34 are commonly used by generators and are zeroed at the beginning of a generator by issuing the 'ZAC' instruction. The counters 35 through 64 are zeroed automatically at the beginning of each new Organization 1-2. They are not commonly used, but can pass results to another generator.

Be sure to document your usage of these counters.

### Counter instructions

Generators can access one and only one counter at a time. The current counter is stored in a field called the current counter number.

SETxx	Set the current counter number to xx. All subsequent counter instructions will use this counter.
S02	Set to counter number 02.
SNC	Set to the next (+1) counter.
SLC	Set to the previous (-1) counter.
ZAC	Zero counters 02–34. Does not affect 35–64.
Z29	Zero counters 02–09.
ZC	Zero the current counter.
CTR	Move the current counter value to the extract record.
CFC	Moves non-computational field from the extract record to the counter specified by the current counter number, converting it to computational format. The output position pointer is reset.
A1C	Add 1.00 to the current counter value.

**Compare counter fields**

Compares the value in the current counter with the value in the next (+1) counter. The Compare has no effect on the Extract Record.

BCL	IF current COUNTER LESS THAN the next higher counter
BCE	IF current COUNTER EQUAL TO the next higher counter
BCG	IF current COUNTER GREATER THAN the next higher counter

- A true condition passes control to the next 255 paragraph.
- A false condition passes control to the next instruction.

**Arithmetic operations**

Arithmetic operations are add, subtract, multiply, and divide.

**Counter/Extract Record**

ADD	Add the contents of the last field moved to the extract record to the value in the current counter.
SUB	Subtract the contents of the last field moved to the extract record from the value in the current counter.

Both instructions require the field in the extract record to be in computational format.

Both instructions automatically reset the output position pointer by the length in the current field size (which will be 4 for a computational field).

Results of the calculations reside in the current counter, not in the extract record.

**Counter/Counter**

MPY	Multiply the contents of the current counter by the contents of the next (+1) counter. Store the results (rounded to 2 decimals) in the current counter.
DIV	Divide the contents of the current counter by the contents of the next (+1) counter. Store the results (rounded to 4 decimals) in the current counter.
DV2	Divide as above except round result to 2 decimals before storing.

Division procedure using counter contents:

- Counter 09 / Counter 10 = Counter 09
- Current Counter/Next (+1) Counter = Current Counter

Results are stored in the counter and must be moved to the extract record for printing. This does not alter the extract record.

**Add/Subtract**

**Examples**

Instruction	Counter 09	Counter 10
ZAC	0.00	0.00
SET10		V
FLD149 (2180.00)		
ADD	0.00	2180.00
SET09	V	
LIT045000		
CFC	50.00	2180.00
CTR		
SET10		V
SUB	50.00	2130.00

**Multiply/Divide**

**Examples**

Instruction	Counter 09	Counter 10
ZC	0.00	0.00
SET10		V
FLD134 (6.5000)		
CFC	0.00	650.00
SET09	V	
LIT042080		
CFC	20.80	650.00
MPY	13520.00	650.00
SET10		V
LIT041200		
CFC	13520.00	12.00
SET09	V	
DV2	1126.67	12.00

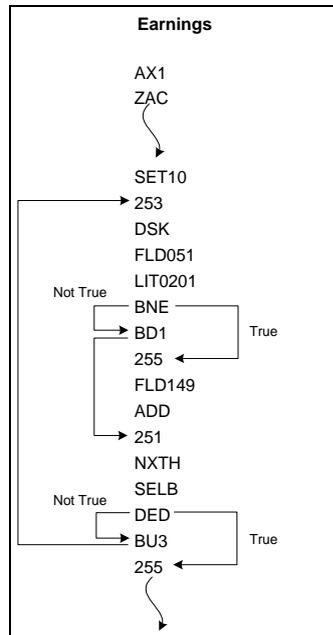
**Index loop examples**

From time to time it is desirable to program an iterative process, performing the same instructions multiple times but using different data, such as the content of the amount fields of all or some subset of the Earnings.

The use of indexing instructions to control data presented (along with branch and compare instructions to determine whether the data is to be included in the process and when the process is finished) makes this possible.

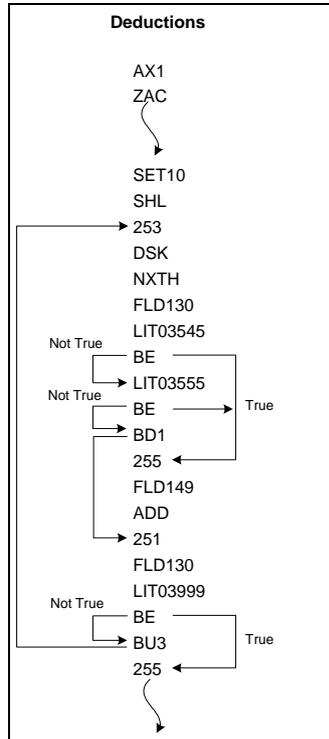
### Earnings

Loop through earnings adding all non-overtime dollars to coun **loop example** ter 10.



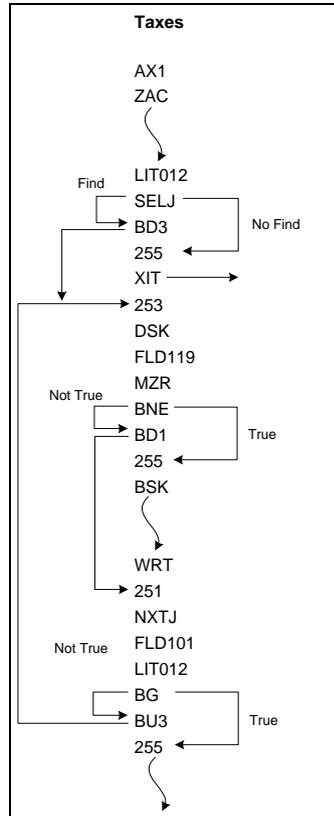
### Deductions loop example

Loop through deductions totaling HED 545 and HED 555.



### Taxes loop example

Set to first state tax body and loop through all states.



## Practice

Write a generator (T3T3) that will produce a state tax report for California. List the following fields on a quarterly basis. Omit employees without the California tax body.

Field Name	Field No.
1. Employee Number	003
2. Employee Name (15 positions)	087
3. Social Security Number	070
4. Legal City & State Address	090
5. Year-to-date Gross Wages	119 (FICA tax body)
6. Quarter-to-date Tax Withheld	120 (California)
7. Year-to-date Tax Withheld	120 (California)

Sort Requirement	
Employee Number	003

Output Positions	
Field Name	Position
1. Employee Number	003
2. Employee Name	015
3. Social Security Number	059
4. Quarter-to-date Tax Withheld	073
5. Year-to-date Tax Withheld	087
6. Year-to-date Gross Wages	102

Total the Gross, Q-T-D state and Y-T-D state.

While writing T3T3, draw out the Extract Record.

## CHAPTER 6

# Method Codes

---

## In This Chapter

Method Codes general rules .....	80
Accessing History and Labor.....	85
Interface File Output.....	88

## Method Codes general rules

This chapter covers the general rules associated with writing, installing, and operating method codes.

### Writing a Method Code

To create a Method Code, the supplied Sort 5G must be used.

The first position of the Format Code (Generator name) must contain an alpha character. (M2, MA, Z5 are proper names; 20, 2M, 57 are not).

The R0 transaction requires only the Format Code in positions 3–4, Format Name in positions 24–47, and 5G in positions 64–65.

Use one R0 and as many R7 transactions as needed. These are the only Generator transactions required.

Processing begins with the first R7 instruction.

At the beginning of processing of a Method Code, the H index points to the active HED. All other index settings are unpredictable and must be controlled and/or initialized by the program.

The Method Code uses the Extract Record Area as a Work Area and the Output Position Pointer is positioned at zero at the beginning of processing.

The results of the Method Code are to be left in one or more Counters as defined later in this course. It is not necessary to use the WRT or WRR instructions.

Messages may be sent to the Payroll Audit Trail using the WAT instruction.

All fields available to a standard generator are available to Method Codes.

### Installing a Method Code

Method Codes must be added to the P20IN file in the same manner as a standard generator.

**Important:** Remember, once installed, you cannot retrieve source coding. You must save your source documentation.

## Operating a Method Code

The Format Code (name of the Method Code) must be entered in the Method-Code-Field (Calculation Method) of the H transaction and the default Method-Code-Field (Calculation Method) of the A8 transaction.

Do not use a D transaction. The Method Code is selected when the Method Code is activated at the employee HED level.

### Sort 5G

The following is a layout of the Sort 5G:

0	1	2	3
1	5	0	5
0	5	0	0
S15G001151			
S75G001XS			
S75G001LIT301			
S75G001LIT04			

### Automatic Earnings/Deductions (HED Method Code)

The name of the Method Code must be entered in the Method Code (Accumulation Method) field of the employee.

The Frequency Code must be set to a code greater than 00 to activate the Method Code. A Frequency Code of 00 will turn off the Method Code for an employee.

The Method Code must place the result of the Amount calculation in Counter 08, which places the content of this counter in the Current Amount field.

The Method Code must place the result of the Hours calculation in Counter 9, which places the content of this counter in the Current Hours field.

When writing a Deduction Method Code, Counter 09 must be zero when you exit, since the Hours field of a deduction is an Arrears amount.

If you moved the H index, P4CALC will restore it to the Active HED after you exit.

Reverse moves cannot be used on the Active HED.

When your Method Code exits, P4CALC will add the contents of CTR 08 to FLD 149 (Current Amount) and CTR 09 to FLD 150 (Current Hours). Current will then roll to MTD, QTD, and YTD.

### Time Entry Method Code

This is valid only for Format 1 Time Entries.

The HED Frequency Code must be 21, and the Method Code must be the name of the Time Entry Method Code.

Position 0 through 25 of the Extract Record Area contains Control 3, 4, 5, 6, and Function from the Time Entry.

Counter 10 will contain the Hours from the Time Entry.

Counter 11 will contain the Rate/Amt from the Time Entry.

The Method Code will process twice for each Time Entry.

On the first pass CTRS 10 and 11 will contain Regular Hours and Rate/Amount from the Time Entry.

On the second pass CTRS 10 and 11 will contain Overtime Hours and Rate/Amount from the Time Entry.

When your Method Code exits, P4CALC will move the contents of Counters 10 and 11 to the Time Entry.

Method Code 9Y is a sample of a user written Time Entry MC that is supplied on CYBMST.

### **Accrual Method Code**

The HED Frequency-Code on the employee H record or Employee Earnings and Deductions form (HH-SCR) must be 22 and the Method-Code (Accumulation Method) must be the name of the Accrual Method Code.

The use of this Method Code requires you to:

1. Add the contents of FLD 141 (Amount One) to a CTR.
2. Add the Accrual amount calculated to that CTR.
3. Reverse Move (RM) the contents of that CTR back to FLD 141 edited as decimal.

The Method Code will process only once per employee, regardless of how many payments are processed.

The Method Code will process whenever an Employee's Frequency is Paid. Thus, control will be passed for Terminated and employees who receive No Pay so the programmer must supply appropriate logic for those circumstances.

## **User exit points**

There are three different points within the system where a user-written report generator routine may receive control:

- HED initialization
- Time entry
- HED processing

### **HED initialization exit point**

This exit point may be used to calculate an accrual.

Enter '22' in the Frequency field of a HED which has an accrual associated with it and place the Format Code of a method code generator in the Method Code field of the HED. Control will be passed to the Method Code whenever the employee's frequency is paid.

Control will only be passed once whether the employee receives several checks or none at all.

The generator will receive control before any pay is calculated. The generator should compute the amount of the accrual, add to it the Amount 1 field and, using the Reverse Move facility, place the result back in the Amount 1 field.

It is the generator's responsibility to check that the employee is not terminated and that the pay run is a normal one.

**Time entry processing exit point**

This exit point may be used to supply or modify a rate or the number of hours in the Regular or Overtime fields of a Format 1 Time Entry.

To use this feature, enter '21' in the Frequency field of the HED and the Format Code of the generator in the Method Code field.

When a Format 1 Time Entry is encountered for these earnings, control will be passed to the generator Method Code.

Counter 10 will contain the contents of the Reg/OT Hours field. Counter 11 will contain the contents of the Reg/OT Rate/Salary field, and the first 26 positions of the extract record will contain Control 3–6 and Function from the time entry.

When the generator exits, the contents of Counter 10 will be used as the number of hours and the contents of Counter 11 will be treated as if it were entered in the Reg/OT Rate/Salary field. This feature will only work for earnings entered on a Format 1 Time Entry.

The time entry method code will be entered twice for each time entry: once with Counters 10 and 11 containing Regular Hours and Rate/Salary, and once with their containing Overtime Hours and Rate/Salary.

If the time entry is to supply information in only the Regular or only the Overtime fields, then the Method Code should begin by comparing the contents of Counter 10 and 11 to zeros and exiting if the condition is true.

**HED processing exit point**

This exit point may be used to compute an Automatic earnings not related to a time entry or a deduction amount. It is commonly referred to as a User Method Code.

The result of the generator calculations is to be placed in Counters 08 and 09. Counter 08 is used to store the Amount calculated by the generator. Counter 09 is used to store the calculated Hours or Units (for earnings only).

The Reverse Move generator instruction may be used within this generator and may be applied to any field with the exception of fields in the current H segment. Any Reverse Move to the current H segment will be nullified.

Upon entry to the generator, the H-Index will point to the HED being processed; the settings of the other indexes are unpredictable.

To exit, use the XIT instruction, not RET.

At all three exit points, the preparation of the generator follows the standard instructions that appear in this manual as modified by the following rules:

1. The Format Code must not be entirely numeric. It may be two letters, a letter and a number or a number and a letter.

For example, it may be XX, X4, or 4X.

The Format code should be placed in the Method Code field on the H card (and/or the Default Method Code field on the A8 card.)

2. Load the format with sort 5G (supplied).
3. The only generator cards required are the R0 and as many R7 cards as needed.
4. Never enter a DD-SCR or Batch D transaction for any Method Code.

## Accessing History and Labor

The R0 transaction, column 19, in conjunction with columns 8–18, determines exactly what types of data may be processed by the report generator.

The following table illustrates how various types of data may be processed:

Data Type	Columns 8–18		Column 19
	Value	Column	Value
+ADJ. HISTORY RECORD	1	8	H, A
-ADJ. HISTORY RECORD	2	9	H, A
MANUAL ADJ. HISTORY RECORD	3	10	H, A
COMPUTER PAYMENT HISTORY RECORD	4	11	H, A
ALL TYPES OF LABOR RECORDS	L	17	L, A
CURRENT + ADJUSTMENTS	1	8	0, A
CURRENT - ADJUSTMENTS	2	9	0, A
CURRENT MANUAL ADJUSTMENTS	3	10	0, A
CURRENT COMPUTER PAYMENTS	4	11	0, A
PERMANENT MASTER RECORD	4	11	0, A
PERMANENT MASTER RECORD - MTD	5	12	0, A
PERMANENT MASTER RECORD - QTD	6	13	0, A
PERMANENT MASTER RECORD - YTD	7	14	0, A
COMPANY HEADER RECORD	C	15	0, L, H, A
TAX RECORDS	T	16	0, L, H, A
'OTHER' RECORD	O (alphabetic)	18	0 (numeric), L, H, A

When an 'A' is used in column 19, the Record Type field (004) of the Sequential Master Record must be examined to determine which record is being processed. This field will contain one of the following values:

*Note:* X = 0–9, YY = 01–98, 0 = Blank Space

Field 004	Record Type
+L	+Adjustment Labor Record
-L	-Adjustment Labor Record
ML	Manual Adjustment Labor Record
XL	Computer Payment Labor Record
+B	+Budget Labor Record
-B	-Budget Labor Record
+H	+Adjustment History Record
-H	-Adjustment History Record
MH	Manual Adjustment History Record
XH	Computer Payment History Record
+0	Current + Adjustment
-0	Current - Adjustment
M0	Current Manual Adjustment
YY	Current Computer Payment
99	Permanent Master Record

### History processing

History processing is created by a computer check or adjustment and it contains some static data.

For all fields present, field numbers are the same as those in the employee Master Record.

History processing contains only the data relevant to this transaction.

Important: You cannot rely on HED 001 or HED 999 being present during history processing.

It is accessed by entering H in column 19 of the R0 transaction.

Columns 8–11 of R0 transaction control the source of the history record.

Multimasters are converted to history records on the pass following a pay run. Therefore, Multimasters data is not available to generators requesting history.

No to-date data—only the gross to net and taxable wages applicable to this transaction—is given.

### Labor processing

Labor records are created for each time entry line item or auto split.

Labor processing contains partial static data and charge-to information.

Control will be passed to the generator once for each labor record.

Field 004 (Record Group) can be used to determine the source of the labor record.

For all fields present, field numbers are the same as used in the employee master (Sequential Master File).

Field 148 (Check Clear Code) can be used to determine if the labor record is current. The value equals 1 for current.

Labor processing contains hours and dollars for this transaction only, no to-date amounts.

It can be accessed by entering 'L' in columns 17 and 19 of the R0 transaction.

### Multi-master processing

A.	01				
		__C__	__M__	__Q__	__Y__
		100.00	100.00	100.00	100.00
B.	02				
		__C__	__M__	__Q__	__Y__
		75.00	175.00	175.00	175.00
C.	03				
		__C__	__M__	__Q__	__Y__
		20.00	195.00	195.00	195.00
D.	99				
		__C__	__M__	__Q__	__Y__
		20.00	195.00	195.00	195.00

- A. The first check is processed, and the system assigns a value of 01 to Field 004 (Record Type).
- B. The second check is processed, and the system assigns a value of 02 to Field 004 (Record Type).
- C. The third check is processed, and the system assigns a value of 03 to Field 004 (Record Type). The system then immediately recognizes that it is the last check and changes 03 to high values.
- D. The high values record is copied onto the employee's permanent master. Please note that in the above example, only three records would be available for interrogation: 01, 02, and 99. The 01 and 02 records would only be available on the pay run that created them.

## Interface File Output

Use SFC6 on the first S7 instruction. This will direct the output to the 80-character output file (P05T80) created by P5PRINT.

The R1 transaction should not generate header lines unless a print image file is being created.

The R7 record type must be zero or an alphabetic character.

No carriage control is allowed.

The R3 transaction must be used to format the extract record.

When using counters, move computational data to the extract record, and then use an edit command (for example, ECF) to change comp to numeric.

Multiple generators may write to the same tape file. A hardware utility should then be used to sort these reports.

Output positions defined on the R3 transaction are relative to position zero, not position 001.

## Recycle file

The Recycle file is a 118 character record file created by P5PRNT to be entered into the next pass of P2EDIT.

R7 transactions are used to build the 118-character recycle record. Do not format with R3/R4 or R5/R6 transactions.

No Sort Key, Record Type, or Carriage Control character is required.

The WRR instruction is used instead of the WRT. This will cause the record content to be shifted to the right by one position and a Forms Code 8 to be entered in position 000.

No batch transaction is required on the Recycle file.

Extract Record format

Position	Content
001–006	Control 1-2
007	J (Employee Record)
008–017	Employee Number
018–045	Move LIT of 28 Blanks
046–047	Transaction Code
048–115	Transaction Data
116–118	Move LIT of 03 Blanks

*Note: It is possible to create a report with the same generator that creates the recycle transactions.*

## CHAPTER 7

# Printing

---

## In This Chapter

Printing .....	90
----------------	----

## Printing

This chapter covers printing the Master File, multiple payment documents, and multiple forms.

### Master File print

The Master File print lists all static fields (both predefined and user defined) and, optionally, the current, month-to date, quarter-to-date, and year-to-date figures for each earnings, deduction, and tax body for every employee whose frequency is being paid that has a value of other than '9999' in field 396.

If column 14 of the D record contains a zero, month-to-date, quarter-to-date, and year-to-date figures for all earnings, deductions, and tax bodies will be omitted from the report. If column 14 of the D record contains a 1, these figures are included.

No modification of this generator is necessary to produce a master file printout for every employee on any run. Simply place '9999' in the User Field of the DD-SCR or D transaction for report 0202.

### Generating multiple payment documents

This procedure describes how to modify the existing payment format (6868) to print the payments on more than one type of stock.

A separate printer file must be designated for each different payment document to be used. Normally, the employee's location or pay type will be used to determine which is to be used.

The required modifications are as follows:

- 1. Assign additional Forms Codes**

Forms Code U and V are already assigned to the standard payment (6868) and deposit (6767) format. Forms Code T and W are available for additional forms.

If more than four payment document forms are required, open, close, and write logic (as well as SELECT and FD entries) must be added to P5PRNT. Forms Codes A through K are currently unassigned and may be used for this purpose.

- 2. Assign additional Report Codes**

A unique four-character Report Code must be chosen for each additional file.

Do not duplicate any existing Report Codes (columns 3–6 of D transactions).

### 3. **Modify sort 68**

The proper Forms Code and Report Code must be assigned based on some field in the employee's record.

For example, if you do not use direct deposit, the entire S7 logic may be replaced with the following:

```
S768001LIT01U
S768002RST
* * * *
```

Insert user logic to supply the Forms Code and Report Code

```
* * * *
S768xxxFLD001L06
S768xxxFLD094L08
S768xxxFLD087L09
S768xxxMUN
```

The user logic should insert U6868 for one of the forms and WCIC1 (for example) for another form, and so forth.

If direct deposit is also going to be used, the user logic must also decide if V6767 should be inserted as the Forms Code and the Report Code.

All of the sort logic before the SFCU instruction is for determining if it is direct deposit (V6767). The logic should be similar to that which is in the original 68 sort.

### 4. **Create dummy sort(s) and format(s)**

Generator 6767 is a good example of a dummy sort and format.

A dummy sort and format must be created for each additional Forms Code/Report Code combination.

The dummies are created by duplicating the following records for each new Report Code.

- Copy the S1 transaction from sort 67.
- Copy the S7 transaction that has the appropriate Forms Code in column 13 from sort 67.
- Copy the R0, R1, R2, R3, R5, and R6 transactions from report 68.
- Copy the R7 transaction from report 67.

The dummy Sort Code must be placed in columns 3 and 4 of each S transaction, and the dummy Format Code must be placed in columns 3 and 4 of each R transaction.

### 5. **Create D records for each dummy report**

Create a D record with the dummy Report Code in columns 3–6 and a '1' in column 7.

### 6. **Update the Master File**

Reload the modified sort 61 records and load each dummy sort, followed by its dummy format, into the system behind a batch record with 999999 in the Organization 1-2 fields. Also load the D records for the dummy Report Codes behind a company BATCH record.

## Multiple forms - example

Assume that one pay document is required for every employee having a Pay Level 3 value of ABCD, and another is required for all other employees. All employees have an Organization 1-2 of 010001.

**1. Assign additional Forms Codes**

Forms Code U will be used for the employees having a Pay Level 3 of other than ABCD, and Forms Code W (an arbitrary selection—T could also be used) for those with a Pay Level 3 value of ABCD.

**2. Assign additional Report Codes**

Report Code CIC1 (another arbitrary selection) will be used with Forms Code W.

**3. Modify sort 68**

The new sort 68 would look as follows:

*Note: The character 'b' in the following listings is being used to indicate a blank space. Do not key it into the transaction.*

```
S168000151bbbb32
S768003FLD094
S768004LIT04ABCD
S768005BE
S768006RST
S768007LIT05U6868
S768008BD4
S768009255
S768010RST
S768011LIT05WC1C1
S768012254
S768013FLDO01L06
S768015FLD094L08
S768017FLD087L09
S7680181STH
S7680191STB
S768020MUN
```

**4. Create dummy sort and format**

The new (dummy) report CIC1 would look as follows:

```
S1C1000151bbbb32
S7C1001LIT01W
R0C1000 remainder same as 68
R1C1001 remainder same as 68
R2C1002 remainder same as 68
R3C1003 remainder same as 68
RSC1004 remainder same as 68
R6C1005 remainder same as 68
R7C1006XIT
```

*Note: Each document could also be formatted differently because each is printed under control of its own R3, R5 and R6 records.*

**5. Create D records for each dummy report.**

The D record for report CIC1 would look as follows:

```
DbC1C11
```

- 6. Update the Master File**
- The input batches to P2EDIT should be as follows:
  - BATCH999999RPTS Y
  - modified sort 68 from step 3
  - dummy report from step 4
  - BATCH01000IDCRD Y
  - D record from step 5 and standard 6868 D record



CHAPTER 8

# Debugging

---

## In This Chapter

Introduction to Debugging Report Generators.....	96
--	----

## Introduction to Debugging Report Generators

A report generator is, for all intents and purposes, a program or, more accurately, a program (P4CALC), a sort (P45SORT), and another program (P5PRNT).

The odds are that a newly written or modified report generator will not produce the desired result on the first try. The programmer should, therefore, conduct a test in such a manner that the maximum amount of information about the test is obtained. The procedure described below will achieve this goal.

### Setting up the test

To minimize confusion, computer time, and paper, it is a good idea to inhibit the execution of all report generators except the one being tested and the Master File printout generator.

This can easily be accomplished by preparing D transactions or DD-SCR screen (Report Request Form) entries for these 2 generators that contain an alphabetic character in column 7 and a P4 transaction with the same character in column 3.

Next, prepare change transactions for several employees near the beginning of the file so that Master File printouts will be triggered for them. (To trigger a Master File printout, it is also necessary to prepare an AE transaction (Payroll Run Process Control form) that causes their frequency to be paid).

Also, prepare an H2 transaction that will prevent an output master. Doing this will not save time, but will allow the Trace generator instruction to be used.

Finally, prepare an R7 transaction with the Trace instruction (TRC) and place it in the generator to be tested before all of the other R7 transactions. Additional trace transactions may also be placed at key points within the R7/S7 logic.

### Running the test

When the test is run, an octal/hex dump of the report file coming out of P4CALC should be obtained. And, of course, if the generator produces tape or disk output, that file should also be dumped.

### Analyzing the results

If the generator did not produce the desired output, or produced no output at all, the place to begin looking is the dump of the report file coming out of P4CALC.

If the generator produced no output at all, try to locate the S1 and R1 through R6 control information near the beginning of the dump. These records will start with the Forms Code specified in the first S7 transaction, followed by the report code, followed by 5 bytes (characters) of low values.

If these records cannot be found, the generator was not loaded into memory by P4CALC. Check the Transaction Load Report and the Payroll Audit Trail for error messages regarding this generator. If none is found, there must be a mismatch between the RUN SELECT code in column 20 of the RO transaction and column 16 of the H2 transaction.

If the control information was found, try to locate a record produced by the trace instruction. It will contain high values in the first position, and the 'address' that was printed next to the TRC instruction on the Payroll Audit Trail in the last 5 positions.

If a record like this cannot be found, control was never passed to the generator. Check the Payroll Audit Trail to see if the generator was selected. The D transaction is probably in error.

If the trace record is found, try to locate an extract record produced by the generator. If there are none, the R7 logic is probably at fault. Either the selection logic is wrong, or the write statement is missing.

If extract records are found, there are a number of conditions that can cause these records to be bypassed by P5PRNT.

Still referring to the dump, verify that the Forms Code and record type are as expected.

Also check to see that the sort key is exactly as long as was specified on the last S1 transaction. Finally, using the Payroll Audit Trail, make sure there is an R3 and/or R5 transaction, and check the options in the R1 transaction (for example, an 'N' in 27 and/or 28 will suppress details, and no 'Y' in 29 will suppress printing of record type 1, and so forth.).

If the generator did produce some output, but it was incorrect, check the extract records in the dump to see if they contain the right data. If they do, the R3/R4 and/or RS/R6 transactions are probably in error. Otherwise, the problem is somewhere in the R7/S7 logic.

Logic problems are more easily solved if trace instructions have been placed at various points. Each trace instruction will cause the extract record to be written out with high values in place of the Forms Code and the Payroll Audit Trail 'address' in the last 5 positions. Thus, it can be used to trace the flow through the generator and, at the same time, provide a snapshot of the extract record.

If necessary, a complete trace can be obtained by placing an 'E' in column 13 of the H2 transaction. In this mode, every instruction executed by every generator will result in a record being written to the extract file in the same format as produced by the TRC instruction.

The exact format is as follows:

<b>Positions</b>	<b>Description</b>
1	High values
2–139	The contents of this portion of the extract record when it was written
140–143	The REPORT CODE of the generator causing the trace record
144	R (R7 instruction being executed) S (S7 instruction being executed)

Positions	Description
145	Reason this record was written: A—Generator is looping B—Output Position Pointer is less than zero C—The Output Position Pointer is greater than 149 D—Current Field Size is less than 1 E—Trace record due to 'E' in column 13 of the H2 transaction T—Trace record due to 'TRC' instruction
146–150	Relative address of R7/S7 instruction (this address is printed on the Payroll Audit Trail for every R7/ S7 instruction).

A 'D' may also be used in column 13 of the H2 transaction. The 'D' will not cause a complete trace of each generator, but will cause an extract record to be written in the format just described for reasons (position 145) 'A', 'B', 'C', 'D', and 'T'.

Any of the legal entries in column 13 of the H2 transaction ('D', 'E', or 'O') will prevent the creation of an output Master File.

If, after following the debugging procedure, you are unable to locate the problem(s), send us:

1. An Audit Trail listing of the generator
2. An octal/hex dump of the unsorted extract file
3. The Audit Trail
4. The resultant report or an octal/hex dump of the P5PRNT output file (if any)
5. A brief note explaining the problem

## Report Generator Debugging Checklist

The report generator debugging checklist has been developed to help when a report generator is not functioning as intended.

Initially, most generator problems are not as complex as they may appear. Determine what in the report generator is not working properly. With that information, read the debugging checklist to locate the condition you have found in your generator.

Second, analyze your generator to determine if one of the common causes listed exists in your generator. If it does, follow the correction instructions, reenter your generator and rerun. Continue this procedure until the report generator is running as intended.

If you have exhausted all possibilities, you will want to advance to the testing/trace phases that are outlined in the section immediately following the debugging checklist. Follow the instructions for each phase required. You will find the H2 transaction, column 13 helpful when analyzing the transactions from P40OUT.

The fourth section is the efficiency considerations, which are intended to be just suggestive. These may help your generator run more efficiently. The comments are by no means intended to discourage you from using some instructions or encourage you to use others, but merely to give you some insight to some conditions you might not otherwise consider.

Condition	Cause	Correction Instruction
1. Generator was not loaded.	A. Run Select Code (col 20 of R0 transaction) does not agree with H2 transaction, col 16.	A. Correct H2 transaction, or Run Select Code on the R0.
	B. Generator failed one of the edits in P2EDIT or P4CALC.	B. Correct errors and re-enter.
2. Generator was not selected.	A. Report Select Code (col 7) of the D transaction is not: Equal to or less than col 4, AE transaction. Equal to col 16–27, AE transaction Equal to col 3–34, P4 transaction	A. Enter the Report Select Code on the AE or P4 transaction.
	B. The generator was requested on the AE transaction (col 16–27) but it was not a Pay Run.	B. Select the generator with a P4 transaction or change the AE transaction to a Pay Run.
	3. Generator was selected; however, no output came from P5PRNT.	A. The Data Selects Code (col 8–16) of the D transaction is greater than col 4 of the AE transaction. Even though the generator is selected, no data is passed.  B. If the IFF instruction was used in the R7 transaction, verify that the appropriate pay frequency was specified in the AE transaction, and that it was a Pay pass, col 4 and 12 are a 1. (Paydate will be present for that AJ on the Control Headers report.)  C. A write instruction—WRT, is missing in the R7 transaction.  D. The sort key length specified in the last S1 transaction is incorrect. Since the record type is the first field after the sort key, the program is looking in the wrong position for the record type.  E. No records passed all selection criteria.  F. The forms code specified in first S7 transaction is not to a printer.  G. Indexes not initialized properly. An AX1 instruction is missing which sets all indexes to the first occurrence. Therefore, tests on various fields in any selection routines are not predictable. The setting of the indexes is unknown.

Condition	Cause	Correction Instruction
4. The generator was selected, but no output from P5PRNT except column headings.	<p>A. The R1 transaction has an "N" in col 27 or 28 (do not print details).</p> <p>B. There are no R3, R4, R5, or R6 transactions to move data from the extract record to the print line.</p> <p>C. A 'Y' was not put into cols 29, 35, 41, 53, 59 or 65 of the R1 transaction for Record Types 1–7 created by this generator.</p> <p>D. An R3/R4 transaction was used to specify the moves for Record Types 1–7.</p> <p>E. A R5/R6 transaction was used to specify moves for a descriptive type detail record–Record Type 0, A–Z.</p>	<p>A. Change col 27 or 28 to a 'Y' or blank.</p> <p>B. Verify that there are R3–R6 transactions with field lengths and print positions specified.</p> <p>C. Put a 'Y' in the appropriate column of the R1 transaction for each Record Type 1–7 detail line you wish to print.</p> <p>D. Change R3/R4 to R5/R6 transactions, or Record Type to 0, A–Z, whichever is applicable.</p> <p>E. Change R5/R6 to R3/R4 transactions, or Record Type to 1–7, whichever is applicable.</p>
5. The detail is printed but the totals and page breaks are not correct.	A. The S1 transactions are not in sync with the S7 transactions.	A. Each S1 transaction must conform in sequence to the S7 transactions, down to the lowest level of control desired. Verify they are in sync. Insert S1 transactions if required.
6. The detail is printed, but not edited or spaced properly.	A. The fields moved to the extract record by the R7 instructions do not correspond to the move instructions specified on the R3/R4 or R3/R6 transaction.	A. Change either the R3/R6 transaction or the R7 transaction so that data on the extract record is accurately reflected by the moves in the R3/R6 transactions.
7. Data is missing from the detail line.	<p>A. The sum total of all fields moved to the extract record exceeds 150 characters causing latter fields not to be moved to output and overlaying code.</p> <p>B. A move specified in an R3/R6 transaction may have overlaid a previously moved field on the print line.</p> <p>C. A move may be missing in an R3/R6 transaction for data on the extract record.</p> <p>D. If the Record Type is 1–7, the move of the counter description may not have been taken into consideration. See #10 below.</p>	<p>A. Modify R7 instructions to produce extract records less than 150 characters. Some machines move 5 position packed, instead of 4 position binary for computational data, check your machine specifications.</p> <p>B. Correct the R3/R6 print position field that is in error.</p> <p>C. Correct the R3/R6 print position field that is in error.</p> <p>D. Correct the R5/R6 fields that are in error.</p>

Condition	Cause	Correction Instruction
8. Random line skipping or run-away printing of paper between detail lines.	<p>A. An erroneous carriage control character was created in the extract record. Valid carriage control characters are:  blank = single space  0 = double space  - = triple space  1 = top of page</p> <p>B. Your machine does not recognize the above carriage control figures as defined.</p> <p>C. Sort key length specified is not proper and a data item in the file is being used as carriage control.</p> <p>D. A carriage control character move was specified in R3/R6 transactions but was not moved to extract record by a R7 transaction.</p> <p>E. Carriage control character not moved to print position 000 by R3/R6 transaction.</p>	<p>A. Correct the R7 instruction to move the proper carriage control character to extract record.</p> <p>B. Check your machine specifications and make appropriate modifications to your generator.</p> <p>C. Correct sort key length in the S1 transaction.</p> <p>D. Insert a R7 instruction in format to move a carriage control character into proper position of the extract record after record type.</p> <p>E. Change R3/R6 transaction to specify moving of carriage control character to print position 000.</p>
9. Abnormally large numbers appear in a dollars or hours print column.	<p>A. A descriptive field was moved as a counter type binary field on the R5/R6 transaction. This is usually the result of extraneous descriptive fields being left on the extract record rather than being reset by a RST instruction in the R7 transactions.</p> <p>B. Counters used were not set to zeros.</p>	<p>A. Identify the extraneous descriptive field on the extract record and either insert a RST instruction in the R7 instructions to reset the field or move descriptive field to the print line.</p> <p>B. Issue a ZAC, Z29, or ZC instruction where appropriate.</p>
10. Dollars or hours field missing on the report.	<p>A. Could be the result of failing to use the first counter type field move in the R5/R6 transactions to the appropriate five position counter description defined in columns 30–34; 36–40; 42–46; 48–52; 54–58; 60–64; 66–70 of the R1 transaction. Therefore, all counter type fields will be shifted by one field move and the last counter field will not be moved from the extract record at all.</p>	
11. Only one detail descriptive line prints when there are multiple print lines per employee.	<p>A. The sort key is exactly the same for all extract records, and descriptive data from R5/R6 moves will only print the first one if the sort key does not change on each extract record.</p>	<p>A. If it is for different records, such as labor, add Movie Unique Number (MUN) instruction to the sort key. If multiple printing from the same record, such as from the Master File, add one to a company counter (36–64) used as part of the sort key.</p>

## Report Generator Debugging Aid

### Testing/trace phase

#### Initial testing

1. Suppress the Output Master File (H2 transaction, column 13 = 0).
2. Suppress the execution of all report generators not related to the test by controlling report selection with a P4 transaction override and a special character in Report Select: D transaction, column 7.
3. Analyze the results using the Report Generator Debugging Checklist.

#### Second phase

1. Suppress all report generator execution except the one being tested. See step 2 above.
2. Insert a Trace (TRC) instruction at various points in the generator logic in both the R7 and S7 transactions.
3. Change column 13 of the H2 transaction to 'D'. This will suppress the Output Master File and result in output records being written to the Extracted Reports File (P400~) describing conditions found in the generator. See the H2 transaction explanation.
4. Obtain an octal/hex dump of the unsorted Extracted Reports File (P400UT) from P4CALC and a ditto dump of any created files, tape or disc.
5. Examine the dump, tracing the flow (Extract Records with a 'T' in position 145) and researching conditions detected by the H2 'D' option.

#### Third phase

1. Suppress all report generator execution except the one being tested. See Initial testing, step 2 above.
2. Change column 13 of the H2 transaction to 'E'. This will suppress the Output Master File and result in an output record being written to the Extracted Reports File (P400UT) for each R7 and S7 instruction executed. See the H2 transaction explanation.
3. Obtain an octal/hex dump of the unsorted Extracted Reports File (P400UT) from P4CALC and a ditto dump of any created files, tape or disk.
4. Analyze the dump of the Extracted Reports File, tracing the flow of the generator from beginning to end.

**Fourth phase**

Send the generator to your Account Manager after phases 1, 2, and 3 have been completed. We will give you as much help as possible in debugging a generator; however, we cannot guarantee your generator problem will be resolved within a given time frame.

Be sure to include the following items:

- The octal/hex dump from Step IV-C above.
- Payroll Audit Trail listing of the generator.
- Payroll Audit Trail Report showing input and Selected/Not Selected from this run.
- A note describing the problem and the purpose of the generator.
- Control Headers Report from this run.
- Master File listing of several employees involved.

**H2 transaction, column 13, extract debugging record format****D Option**

D Option in Column 13 creates records as follows for each problem Extract Record/Instruction only:

Position	Description
001	High Values
002–143	Contents of this portion of the Extract Record when it is written
144	R - an R7 instruction is being executed S - an S7 instruction is being executed
145	Reason this record is written: A Generator is looping B Output position pointer is greater than 149 C Output position pointer is greater than 149 D Current field size is less than 01 T Trace record due to the TRC instruction in R7 or S7 logic
146–150	Relative address of the R7 or S7 instruction

**E Option**

E option in column 13 creates records as above but for every extract record/instruction with an additional reason code used:

Position	Description
145	E - Trace record due to 'E' in column 13



PART 3

# Appendices

---

## In This Section

Generator Instructions .....	107
Sample Report Generator.....	137
Sample Field Table .....	145



APPENDIX A

## Generator Instructions

---

### In This Appendix

Generator Instructions .....108

## Generator Instructions

### Alphabetic List of Instructions

Mnemonic	Number	Description
ADA	054	Add days to date
ADD	032	Add to counter
AMO	086	Add months to date
AP	221	Accounts Payable amount
AX1	024	Set all indexes to 1
A1C	035	Add 1.00 to counter
BCE	039	If counter equal to
BCG	040	If counter greater than
BCL	038	If counter less than
BD1	078	Branch down to 251
BD2	080	Branch down to 252
BD3	052	Branch down to 253
BD4	027	Branch down to 254
BD5	025	Branch down to 255
BE	010	Branch if equal
BF1	222	Browse FILE01
BG	011	Branch if greater
BL	009	Branch if less than
BLZ	099	Blank leading zeros
BNE	013	Branch if not equal
BNG	014	Branch if not greater
BNL	012	Branch is not less than
BOP	055	Bump output position
BO1	174	Bump output position 1 or more
BO2	175	Bump output position 2
BSC	065	If same ORGANIZATION 1-2
BSE	066	If same Employee
BSK	016	Build sort key
BU1	079	Branch up to 251
BU2	081	Branch up to 252
BU3	053	Branch up to 253
BU4	028	Branch up to 254

<b>Mnemonic</b>	<b>Number</b>	<b>Description</b>
BU5	026	Branch up to 255
CAD	048	Condense address
CAL	197	Call generator
CCF	042	Edit S9(8)V99
CFC	056	Convert to counter
CHL	062	Chop left
CHR	063	Chop right
CML	177	Change move length
CNZ	076	If HED current not zeros
CTG	234	Complement to Gregorian
CTR	034	Move counter
DED	070	If deduction
DIV	036	Divide counter
DSK	090	Dummy sort key
DTE	232	Edit century date
DV2	173	Divide-2 decimal result
ECF	096	Edit 2(7)VZZ or S9(7)V99
EDN	235	Edit -----,99
EDT	060	Edit ----,---,99
ELZ	236	Edit 9(7)v99
EP	097	Edit packed S9(9)V99
ERN	067	If earnings
EX1	224	Exchange FILE01
EXC	051	Set X to save index
ES	047	Edit \$\$,\$\$\$,\$\$9V99
FC	073	Find category
FLD	017	Move
FML	045	Reverse name
FST	084	Set to first state
GRP	057	Move group
GTC	233	Gregorian to Complement
HUP	140	Set H to-date up 1
IFF	085	Select if frequency
JUP	145	Set J to-date up1
LIT	018	Move literal
MDY	008	Edit date to MM/DD/YY
MEM	186	Reserve memory

Mnemonic	Number	Description
MPC	181	Multiply percent
MPY	037	Multiply counter
MSD	094	Move system date
MST	095	Move system time
MTO	188	Move to other
MTW	201	Move to work
MUN	064	Move the unique number
MVW	195	Move work
MZR	029	Move zeros
NCT	069	If not combo/tips
NET	058	Set to net pay
NMD	071	If not memo deduction
NME	068	If not memo earnings
NNP	072	If not net pay
NPA	216	Net pay available
NSE	074	If not shift earnings
NTE	075	If not tips earnings
NXT	019	Set Index up 1
OVT	218	Overtime pay amount
PEX	194	Perform exit
PF4	193	Perform
PRV	020	Set Index down 1
RAD	049	Restore address
RCL	185	Recall
REG	217	Regular pay amount
REP	059	Replace
RET	198	Return
RF1	223	Read FILE01
RM	192	Reverse move
RST	030	Reset last move
SBL	083	Set B to last earnings
SDD	087	Subtract date from date
SEL	022	Set Index to Immediate
SET	401	Set to counter XX
SFC	176	Sort FORMS CODE
SHC	141	Set H to-date to current
SHL	082	Set B, H to last earnings

<b>Mnemonic</b>	<b>Number</b>	<b>Description</b>
SHM	142	Set H to-date to month
SHQ	143	Set H to-date to quarter
SHY	144	Set H to-date to year
SJC	146	Set J to-date to current
SJM	147	Set J to-date to month
SJQ	148	Set J to-date to quarter
SJY	149	Set J to-date to year
SLC	092	Set to lower counter
SNC	091	Set to next counter
SPC	046	Blank out line
SPL	219	Special pay amount
STO	184	Store
SUB	033	Subtract
S02	093	Set to
TNZ	077	If tax not zero
TRC	098	Trace
VAC	220	Vacation pay amount
WAT	205	Write audit trail
WCR	006	Write payment record
WRR	088	Write recycled record
WRT	005	Write extract record
XIT	007	Exit report
XS	015	Exit sort
XTD	023	Set X to-date to n
YMD	216	Edit date to YYMMDD
ZAC	050	Zero counters 02-34
ZC	031	Zero counter
ZSC	061	Edit Z(7)VZZ
ZS2	043	Edit 2 decimals
ZS4	044	Edit 4 decimals
Z29	182	Zero counters 02-09
1ST	021	Set X to 1
251	251	Paragraph name
252	252	Paragraph name
253	253	Paragraph name
254	254	Paragraph name
255	255	Paragraph name

## Unconditional branching

There are 10 unconditional branch instructions, 5 to branch up and 5 to branch down to each of the 5 paragraph names.

The branch instructions are written in the operation field; the operand field is not used.

Instruction	Result
BU1 (079)	Branch up to paragraph name 251
BU2 (081)	Branch up to paragraph name 252
BU3 (053)	Branch up to paragraph name 253
BU4 (028)	Branch up to paragraph name 254
BU5 (026)	Branch up to paragraph name 255
BD1 (078)	Branch down to paragraph name 251
BD2 (080)	Branch down to paragraph name 252
BD3 (052)	Branch down to paragraph name 253
BD4 (027)	Branch down to paragraph name 254
BD5 (025)	Branch down to paragraph name 255

In each case, control is transferred to the first specified paragraph name encountered, either up or down, depending on the instruction, from the branch instruction.

An error message will be printed on the Payroll Audit Trail when the report is loaded if a branch down instruction can not be resolved (for example, the specified paragraph name does not exist later in the generator).

Each generator is assumed to begin with all 5 paragraph names; a branch up will never be unresolved.

### **PF4 (193) PERFORM paragraph 254 through PEX**

This instruction branches down to the next 254 paragraph and performs the next series of instructions until a PEX instruction is encountered.

The instructions located between paragraph 254 and PEX may not contain another PF4 instruction.

In other words, nested performs are not allowed. If one is found, the generator will be disabled, and an error message will print on the Payroll Audit Trail with the relative address of the illegal perform.

### **PEX (194) Exit from a Perform**

This instruction is used in conjunction with PF4 as an exit from a performed routine.

If the generator encounters this instruction when no perform is in effect, the instruction is ignored.

## Compare instructions

There are three general types of compare instructions. One type compares two fields that are in the extract record, another type compares a counter to the next (higher numbered) counter, and the third compares a field in an employee's record or the company header record to a value, a group or range of values, or a field in the master record on the Master File.

In each case, control is transferred down to the next paragraph named 255 if the specified condition is satisfied.

The compare instructions are written in the operation field; the operand field is not used.

### Extract Record Compares

Instruction	Comparison
BL (009)	If less than
BE (010)	If equal to
BG (011)	If greater than
BNL (012)	If not less than
BNE (013)	If not equal to
BNG (014)	If not greater than

The Current Field Size is subtracted from the Output Position Pointer.

The field preceding the new Output Position Pointer is compared to the field beginning at the Output Position Pointer.

The Current Field Size determines the number of characters to be compared. If the specified condition is true, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare instruction.

If the Current Field Size is equal to the Output Position Pointer prior to the execution of the compare, the results of the compare are unpredictable.

### Counter Compares

Instruction	Comparison
BCL (038)	If counter is less than
BCE (039)	If counter equal to
BCG (040)	If counter greater than

The counter specified by the Current Counter Number is compared to the next higher numbered counter. If the specified condition is true, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare instruction.

If the Current Counter Number is 64, the results of the compare are unpredictable.

### **Miscellaneous compares**

#### **BSC (065) If next record is the same Organization 1–2**

The Organization 1–2 in the current master record is compared to the Organization 1–2 in the next record on the input Master File (which has already been read in by P4CALC).

If they are equal, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare instruction.

This instruction is effectively an unconditional branch if history or labor records are being processed because the next record will always be another history or labor record or a current master for the same employee.

#### **BSE (066) If next record same employee**

The Record Group (field 004) is compared to '99'.

If it is not equal to '99', control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

The Record Group field will only contain '99' when the last master is being processed for an employee.

This instruction is effectively an unconditional branch if history or labor records were being processed, because the Record Group field will never contain '99' for a history or labor record.

#### **CNZ (076) If current HED not zeros**

The Record Group (field 004) is compared to '99'.

If it is equal to '99', control is transferred down to the next paragraph named 255. Otherwise, the current amount and hours fields (fields 149 and 150) of the HED pointed to by the H index are compared to zeros.

If they are not zeros, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

#### **ERN (067) If HED is earnings**

The HED Category (field 051) of the company HED pointed to by the B index is compared to 51.

If it is less, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

#### **TNZ (077) If tax amount not zero**

The tax body pointed to by the J index, fields 117 through 124 (current, MTD, QTD, or YTD, depending on the setting of the J To-Date index) are compared to zeros.

If they are not all zeros, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

For tax body 101 only, the compare is only made against the fields 117 through 122.

*Note: The following compare instructions refer to company level HED information pointed to by the B index, not the H index. To have the B index and H index both pointing to information for the same HED number use the SELB instruction.*

**NME (068) If earnings not memo**

The HED Category (field 051) of the company HED pointed to by the B index is compared to 14, 16, 17, 18, and 19.

If it is not equal to any of these values, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

**NCT (069) If earnings not combo or tips**

The HED Category (field 051) of the company HED pointed to by the B index is compared to '09', '10', and '11'.

If it is not equal to any of these values, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

**DED (070) If HED is deduction**

The HED Category (field 051) of the company HED pointed to by the B index is compared to '51'.

If it is equal or greater, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

**NMD (071) If deduction not memo**

The HED Category (field 051) of the company HED pointed to by the B index is compared to '54', '55', '56', and '57'.

If it is not equal to any of these values, control is transferred down to the next paragraph named 255; otherwise, processing continues with the instruction following the compare.

**NNP (072) If ORGANIZATION 1-2 HED not net pay**

The HED Number (field OSO) of the company HED pointed to by the B index is compared to '999'.

If it is not equal, control is transferred down to the next paragraph named 255; if the HED is '999', processing continues with the instruction following the compare.

**NSE (074) If HED not shift differential earnings**

The HED Category (field 051) of the company HED pointed to by the B index is compared to '06' and '07'.

If it is not equal to either of these values, control is transferred down to the next paragraph named 255; if the category is '06' or '07', processing continues with the instruction following the compare.

**NTE (075) If HED not tips earnings**

The HED Category (field 051) of the company HED pointed to by the B index is compared to '08' and '09'.

If it is not equal to either of these values, control is transferred down to the next paragraph named 255; if the category is '08' or '09', processing continues with the instruction following the compare.

### Edit instructions

There are three different types of edit instructions for generators:

- Date conversions
- Numeric conversions
- Text edits

#### **Date conversions**

##### **MDY (008) Edit date to MM/DD/YY**

The report generator assumes that the last field moved to the extract record was a 6-position date in the format YYMMDD or Complement Date format, cyymdd.

The date field is converted from YYMMDD or cyymdd to MM/DD/YY and is placed in the extract record overlaying the old date.

The Output Position Pointer is incremented by 2 and the Current Field Size is set to 8.

##### **YMD (216) complement to year month day date conversion**

The report generator assumes that the last field moved to the extract record was a 6-position date in Complement format (cyymdd). The date field is converted to YYMMDD format.

The Field Size and Output Position Pointer are not altered.

##### **DTExy (232) Edit date**

The report generator assumes that the last field moved to the extract record is a date.

The date may be in format CCYYMMDD, YYMMDD or Complement Date format, cyymdd.

The resulting date will be edited according to the following value of x:

x = 1 convert to CCYY/MM/DD

x = 2 convert to MM/DD/CCYY

x = 3 convert to DD/MM/CCYY

The value of the 'y' parameter will appear as the delimiter. The above assumes that the value of 'y' was '/'.

In cases where the input date was of the format CCYYMMDD, the Output Position Pointer is incremented by 2 and the Current Field Size is set to 10.

In cases where the input date was of the format YYMMDD or cyymdd, the Output Position Pointer is incremented by 4 and the Current Field Size is set to 10.

The 'x' parameter is required or the resulting date will contain '\*':

##### **GTC (233) Gregorian to complement date conversion**

The report generator assumes that the last field moved to the extract record was a 6-position date in the format YYMMDD. The date field is converted from YYMMDD to the Complement Date format (cyymdd).

Neither the Output Position Pointer nor the Current Field Size is altered.

**CTG (234) complement to Gregorian date conversion**

The report generator assumes that the last field moved to the extract record was a 6-position Complement Date format (cyymdd). The date field is converted from cyymdd to Gregorian Date format (CCYYMMDD).

The Output Position Pointer is incremented by 2 and the Current Field Size is set to 8.

**Numeric conversions**

**CFC (056) Convert to counter**

The last field moved to the extract record is moved to an internal work area.

The work area is examined, and any non-numeric characters are converted to zeros.

The work area is converted to binary and moved to the counter specified by the Current Counter Number.

The Current Field Size is subtracted from the Output Position Pointer. If a literal is being converted to counter format, the LIT instruction should immediately precede the CFC, as the system will generate more efficient code.

**CCF (042) Convert counter**

The report generator assumes that the last field moved to the extract record is in counter format.

The field is converted from counter format to a 10 digit signed external decimal number and is placed in the extract record overlaying the old field.

The Output Position Pointer is incremented to point to the first position in the extract record following the converted field, and the Current Field Size is set to 10.

**EDT (060) Edit counter to -----,---.99**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture of -----,---.99.

The Current Field Size is set to 12 and the Output Position Pointer is incremented to the first position in the extract record following the edited field.

**ECF (096) Edit counter format field for plus or minus**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture of S9(7)V99 if the field is negative, or Z(7)VZZ if it is not. The V indicates the assumed position of the decimal point.

The Current Field Size is set to 9 and the Output Position Pointer is incremented to the first position in the extract record following the edited field.

### **ZS2 (043) Edit 2 decimals**

The report generator assumes that the last field moved to the extract record is in external decimal format.

The 2 low-order digits are moved to the right one position and a '.' is inserted. The field is examined from left to right and all leading zeros are converted to blanks.

If the number is negative, a '-' will replace the high order digit.

Both the Output Position Pointer and the Current Field Size are incremented by one.

### **ZS4 (044) Edit 4 decimals**

The report generator assumes that the last field moved to the extract record is in external decimal format.

The 4 low-order digits are moved to the right one position and a '.' is inserted.

The field is examined from left to right and all leading zeros are converted to blanks. If the number is negative, a '-' will replace the high order digit. Both the Output Position Pointer and the Current Field Size are incremented by one.

### **ZSC (061) Edit counter to ZZZZZZVZZ**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture of `ZZZZZZVZZ`. The V indicates the assumed position of the decimal point.

The Current Field Size is set to 9, and the Output Position Pointer is incremented to the first position in the extract record following the edited field.

### **E\$ (047) Edit with \$**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture of `$$,$$$,$$9V99` (the V indicates the assumed position of the decimal point).

The Current Field Size is set to 12, and the Output Position Pointer is incremented to the first position in the extract record following the edited field.

### **EDN (235) Edit Counter Format to -----.99**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture `-----99`.

The current field size is set to 12, and the Output Position Pointer is incremented to the first position of the extract record following the edited field.

**ELZ (236) Edit Counter to 9(7)V99**

The report generator assumes that the last field moved to the extract record is in counter format.

It is edited using a picture 9(7) V99. The V indicates the assumed position of the decimal point.

The current field size is set to 9, and the Output Position Pointer is incremented to the first position of the extract record following the edited field.

**EP (097) Edit packed (IBM users only)**

The counter specified by the Current Counter Number is converted to packed using a picture of S9(9)V99 and moved to the extract record beginning at the position specified by the Output Position Pointer. The V indicates the assumed position of the decimal point.

The Current Field Size is set to 6, and the Output Position Pointer is incremented by 6.

**BLZ (099) Blank leading zeros**

The last field moved to the extract record is examined from left to right, and all leading zeros are changed to blanks.

This is applicable to numeric fields only.

**Text edits**

**FML (045) Edit name**

The report generator assumes the last field moved to the extract record is a name in the form of last name, comma, space, first name.

The field is changed to the format first name, space, last name.

If the Current Field Size is not equal to 30, or if no comma is found, or if the comma is not followed by a space, the field is not changed.

**REP (059) Examine replacing**

The report generator assumes that the previous instruction was a LIT02xy and that the instruction before that was an FLDaaa.

Field aaa in the extract record is examined and all occurrences of 'x' are replaced by 'y'.

The Output Position Pointer is reduced by 2 and the Current Field Size is set to the length of field aaa.

Normally, the length of the next to last field moved to the extract record is not available; however, the LIT instruction saves it for use by the REP instruction.

**CHL (062) Chop left**

The number of positions specified in the operand field, which must be in the range of 01–30, is chopped off the last field in the extract record. The result is moved to the extract record by shifting the remainder of the field to the left.

Both the Output Position Pointer and the Current Field Size are reduced by this value.

If the operand field is left blank, a value of 01 is assumed.

### **CHR (063) Chop right**

Both the Output Position Pointer and the Current Field Size are reduced by the value of a 2-digit operand that must be in the range of 01 through 30.

If the operand field is blank a value of 01 is assumed.

## **Counter manipulation instructions**

### **ZC (031) Zero counter**

The counter specified by the Current Counter Number is set to zeros.

### **ADD (032) Add to counter**

The last field, which must be in 'counter format', moved to the extract record is added (a computational add) to the counter specified by the Current Counter Number.

The Current Field Size is subtracted from the Output Position Pointer.

If the last field moved to the extract record was not the length of a counter, the results of this instruction are unpredictable.

### **SUB (033) Subtract from counter**

The last field, which must be in 'counter format', moved to the extract record is subtracted (a computational subtract) from the counter specified by the Current Counter Number.

The Current Field Size is subtracted from the Output Position Pointer.

If the last field moved to the extract record was not the size of a counter, the results of this instruction are unpredictable.

### **CTR (034) Move counter to output**

The counter specified by the Current Counter Number is moved to the extract record beginning at the position specified by the Output Position Pointer

The Current Field Size is set to the length of a counter (varies depending on computer) and then added to the Output Position Pointer.

### **A1C (035) Add 1 to counter**

1.00 is added (a computational add) to the counter specified by the Current Counter Number.

### **DIV (036) Divide counter**

The counter specified by the Current Counter Number is divided by the next higher numbered counter, and the results are stored in the first counter.

Both the dividend and the divisor are assumed to have 2 decimal places. The quotient is rounded to 4 decimal places; thus its picture is S9(5)V9999. The V indicates the assumed position of the decimal point

If the divisor is zero, the quotient is set to zero.

If the Current Counter Number is 64 the results of this instruction are unpredictable.

**DV2 (172) Divide with 2 decimal result**

The counter specified by the Current Counter Number is divided by the next higher numbered counter, and the results are stored in the first counter

Both the dividend and the divisor are assumed to have 2 decimal places. The quotient is rounded to 2 decimal places; thus its picture is S9(7)V99. The V indicates the assumed position of the decimal point.

**MPY (037) Multiply counter**

The counter specified by the Current Counter Number is multiplied by the next higher numbered counter, and the results are stored in the first counter.

Both the multiplier and the multiplicand are assumed to have two decimal places. The product is rounded to 2 decimal places.

If the Current counter number is 64, the result of this instruction is unpredictable.

**SET (401) Set Current Counter Number**

The first 2 positions of the operand field must specify a valid counter number (02-64).

The Current counter number is set to this value.

**ZAC (050) Zero all counters**

The lowest number counters (02-34) are all set to zeros.

To zero a higher numbered counter (35-64), the ZC (031) instruction must be used.

**SNC (091) Set to next higher numbered counter**

The Current Counter Number is incremented by 1.

If the Current Counter Number is incremented past 64, the results of any of the instructions involving counters are unpredictable.

**SLC (092) Set to next lower number counter**

The Current Counter Number is decremented by 1.

If the Current Counter Number is decremented below 20, the results of any of the instructions involving counters are unpredictable.

**S02 (093) Set to counter 2**

The Current Counter Number is set to 02.

**Z29 (182) Zero counters 02 through 09**

Counters 02–09 are set to zeros.

**MPC (181) Multiply percent**

The last field moved to the extract record, which is assumed to be in counter format with a picture of S9V9(6), is multiplied by the counter specified by the Current Counter Number.

The counter is assumed to have 2 decimal places. The product is rounded to 2 decimal places and stored in the counter. The V indicates the assumed position of the decimal point

The Current Field Size is subtracted from the Output position Pointer.

## Extract record manipulation instructions

### **BSK (016) Build sort key**

The Output Position Pointer is set to zero, and the instructions on the S7 transactions are performed.

Control is returned to the instruction following the BSK when either an XS instruction or the last S7 instruction is executed.

The settings of the Output Position Pointer and the Current Field Size are determined by the S7 instructions.

### **FLD (017) Move**

The field specified in the operand field is moved from a Master File record to the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to the length of the data that was moved and then added to the Output Position Pointer.

Optionally, the length of the data moved and/or the displacement into the field at which the move begins may be modified.

A length modification is indicated by an 'L' followed by a 2 digit number in the range of 00 through 60 (00 is replaced by a length equal to the size of a counter).

A displacement modification is indicated by a 'D' followed by a 2-digit number in the range of 01 through 60.

If both a length modifier and a displacement modifier are specified, the length modifier must come first.

Thus, for example, to move the 2nd and 3rd characters of field 094, the following would be coded:

```
FLD094L02D01
```

The length and displacement modifiers are only subject to the restrictions just mentioned; they are not bound by the actual length of the field specified, for example, FLD094L16 will move fields 094, 095, 096, and 097, because they are contiguous).

A Reverse Move which moves data from the extract record to the Master File record is performed by adding an 'R' to the FLD instruction. The 'R' is placed immediately following all parameters coded for this instruction.

For example, using the preceding move, FLD094L02D01R, we can move two characters back into the second and third positions of field 094.

Finally, the Reverse Move instruction resets the Current Position Pointer after execution.

*Note: Extreme caution must be taken with this instruction because it is possible to put data into the Master File that would not be valid on the next run.*

*Note: Many of the fields in a Master File record exist within occurrences. When one of these fields is moved, the setting of that segment's index determines which occurrence the field is moved from. Additionally, fields 117 through 124, 149, and 150 occur 4 times (in a multiple or permanent employee master) within these occurrences. Which of the 4 is moved is determined by the setting of the J To-Date index (for taxes) or the H To-Date index (Earnings/Deductions).*

**LIT (018) Move a literal**

The operand field must contain a 2-digit number in the range of 01–53 followed by a literal of that length.

The literal is moved to the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to the length of the literal and then added to the Output Position Pointer.

**MZR (029) Move zero (counter format)**

Zeros, in counter format (signed, computational), are moved to the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to the length of a counter and then added to the Output Position Pointer.

**RST (030) Reset last move**

The Current Field Size is subtracted from the Output Position Pointer.

Note that another RST instruction will subtract the same number from the Output Position Pointer and not the size of the previous field moved to the extract record.

**SPC (046) Blank extract record**

Spaces are moved to the entire 150-position extract record, and the Output Position Pointer is set to zero.

**ADA (054) Add days to date**

The report generator assumes that the 8 positions in the extract record immediately preceding the Output Position Pointer contain a date in the YYMMDD format or in the cymdd format followed by a 2-digit number.

This number is added to the YYMMDD date (for example, 770630 + 03 = 770703).

The Current Field Size is set to 6 and the Output Position Pointer is decremented by 2.

A maximum of 68 days may be added to a date.

The resulting date will be in the same format as the original.

**BOP (055) Bump output position**

The Current Field Size is added to the Output Position Pointer.

The extract record is unchanged.

### **GRP (057) Move group**

This instruction is used to move a large block of data from the Master File, or it may be used to move several contiguous counters to the extract record.

This instruction sets the Current Field Size to the value contained in the counter specified by the Current Counter Number.

GRP can be used in place of the FLD instruction when it is necessary to move more than 60 characters. For this purpose, the counter used is assumed to have no decimal places.

Therefore, when specifying the number of characters to be moved, do not allow for two decimal places (see the LIT instruction in the example below).

To use the instruction, it is necessary to identify the two variables, (1) how many characters to move, and (2) where to move them from.

In the example, lines 2, 3, and 4 place the number of characters to move in a counter.

Line 1 moves field 087, which establishes the point from which data is to be moved; in this case, movement begins with field 088.

A group of counters can be moved to the extract record in a similar manner by executing a CTR instruction instead of an FLD instruction.

For example:

1. FLD087 Move field 087 (name) to extract record
2. SET03 Set to counter 03
3. LIT0290 Move literal of 90 to extract record
4. CFC Convert literal to binary and store in counter 03
5. GRP Move 90 characters to extract record (fields 088, 089, 090, and 091)

### **MUN (064) Move unique number**

The unique number (4 positions —computational) is moved to the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to 4, and 4 is added to the Output Position Pointer.

Each record on the Master File is assigned a unique number beginning with 1. Thus, it may be used as part of a sort key to insure a unique key when sorting on the employee name field or other fields that may not be unique.

### **AMO (086) Add months to date**

The report generator assumes that the 8 positions in the extract record immediately preceding the Output Position Pointer contain a date in the YYMMDD format or in the Complement Date format, cymdd, followed by a 2-digit number in the range 01 to 87.

The 2-digit number is added to the months portion of the YYMMDD date (for example, 771101 + 03 = 780201).

The Current Field Size is set to 6, and the Output Position Pointer is decremented by 2.

The resulting date will be in the same format as the original.

**SDD (087) Time span between dates**

The report generator assumes that the 12 positions in the extract record immediately preceding the output Position Pointer contain 2 dates in the YYMMDD format or in the Complement Date format, cyymdd.

It further assumes that the Current Field Size is set to 6 and that the larger (newer) date precedes the smaller (older) date.

The smaller date is subtracted from the larger one, and the result replaces the larger one (for example, 780115 - 761101 = 010214).

The answer is expressed in years, months, and days.

The Current Field Size is set to 6, and the Output Position Pointer is decremented by 6.

The resulting date will be in the same format as the original.

If the first date is not greater than the second one, or either or both dates are zero, the result will be zeros.

*Note:* This instruction will produce unpredictable results if the two dates are not in the same century.

**DSK (090) Dummy sort**

The Output Position Pointer is set total sort key length.

**MSD (094) Move system date**

The current date (provided by the operating system) in the format MM/DD/YY is placed in the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to 8, and then added to the Output Position Pointer.

**MST (095) Move system time**

The current time (provided by the operating system) in the format HH/MM/SS is placed in the extract record beginning at the position specified by the Output Position Pointer.

The Current Field Size is set to 8 and then added to the Output Position Pointer.

**BO1 (174) Bump output position by 1 or more**

The value in the operand field, a 2 digit number in the range of 01 through 30, is added to the Output Position Pointer and becomes the Current Field Size.

The extract record is unchanged.

If the operand field is left blank, a value of 01 is assumed.

**BO2 (175) Bump output position by 2**

Two is added to the Output Position Pointer.

The extract record is unchanged.

### **SFC (176) Sort FORMS CODE**

The first position of the operand field and fields 151 (Report Code), 001 (Control 1), and 002 (Control 2) are moved to the extract record beginning at the first position.

The Current Field Size is set to 4 (the length of Control 2), and the Output Position Pointer is set to 11.

### **CML (177) Change move length**

The value in the operand field, a 2-digit number in the range 01 through 30, is placed in the Current Field Size.

The extract record is unchanged.

### **MEM (186) Reserve memory**

An area of memory is reserved equal in size to the value in the operand field.

This value must be in the range of 01 through 30.

Optionally, the initial value of this area may be entered in the operand field immediately following the size operand.

### **STO (184) Store**

The last field(s) moved to the extract record is moved to an area reserved by the last MEM instruction.

This length is subtracted from the Output Position Pointer.

### **RCL (185) Recall**

The data in the area reserved by the last MEM instruction that was executed is moved to the extract record beginning at the position specified by the Output Position Pointer.

The length of the reserved area is moved to the Current Field Size and then added to the Output Position Pointer.

### **MTO (188) Move to 'Other' record**

This instruction updates an 'Other' record (that is, a record with a data type of F, G, W, or X).

To accomplish the update, first move the new data to the extract record. Then, using a single FLD instruction, move the old data from the 'Other' record to the extract record, and execute the MTO instruction.

The combined length of the new and old data will be subtracted from the Output Position Pointer. The 'to' area of the 'Other' record being updated must be beyond column 27.

### **RM (192) Reverse Move**

This instruction will move the data in the extract record to the Master File field immediately following the RM instruction.

For example:

```
RM             REVERSE MOVE
FLD097        MOVE PAY LEVEL 6
```

The Reverse Move instruction resets the Current Position Pointer after execution.

A Reverse Move which moves data from the extract record to the Master File record may also be performed by adding an 'R' to the FLD instruction. The 'R' is placed immediately following all parameters coded for this instruction.

For example, using the instruction, FLD094L02D01R, we can move the last two characters in the extract record back into the second and third positions of field 094.

Finally, the Reverse Move instruction resets the Current Position Pointer after execution.

*Note:* *Extreme caution must be taken with this instruction because it is possible to put data into the Master File that would not be valid on the next run.*

### **MVW (195) Move Data to or from a Work Area**

This is used to move information from the work areas to the extract record or from the extract record to the work area.

The direction of move is controlled by the value in field 853.

### **MTW (201) Move data to a work field**

This instruction moves the current field in the extract record to the work field specified in the operand.

This instruction resets the Current Position Pointer after execution.

For example:

```
LIT040661 MOVE LITERAL
MTW851 MOVE IN LENGTH
```

### Index manipulation instructions

An index is identified by a single alphabetic character that is the same as the segment it is associated with.

The valid indexes are B, C, E, F, G, H, J, L, P, and W.

#### **NXT (019) Set index up 1**

The index specified in the operand field is incremented to point to the next occurrence of the segment.

If the index was already pointing to the last occurrence, it is set to point to an area that contains high values in the first position.

#### **PRV (020) Set index down 1**

The index specified in the operand field is decremented to point to the previous occurrence of the segment.

If the index was pointing to the first occurrence, it is decremented to point to an area before the first occurrence so that a subsequent NXT instruction will result in the index pointing to the first occurrence again.

#### **1ST (021) Set index to 1**

The index specified in the operand field is the first occurrence of the segment.

#### **SEL (022) Set immediate**

The occurrences of the segment specified in the operand field are scanned, beginning with the occurrence pointed to by that segment's index, searching for a match between a search argument and an equal number of characters at the start of an occurrence.

If no match is found, control is transferred down to the next paragraph named 255, and the index is left pointing to the first occurrence encountered that contained a value greater than the search argument.

Specifying a 'B' in the operand field results in the HED Number (field 130) of the H segment occurrence pointed to by the H index being used as the search argument.

Thus, the company HED index (B index) is set to the same HED Number as the employee HED currently pointed to by the H index.

Specifying a 'T' in the operand field results in the Tax Type (field 101) and Tax Number (field 102) of the J segment occurrence pointed to by the J index being used as the search argument.

Thus, the company tax index (T index) is set to the same Tax Type and Tax Number as the employee tax body currently pointed to by the J index.

Specifying any other index (other than B or T) results in the last field moved to the extract record being used as the search argument, and the Current Field Size is subtracted from the Output Position Pointer.

Thus, for example, to set the J index to the first Ohio city tax body, the following may be used:

LIT0340H
SELJ

*Note:* Searching always begins with the occurrence pointed to by the index specified in the operand field. The index is not automatically set to 1 prior to the search. No match will be found if the index was already pointing past the desired occurrence.

**XTD (023) Set to-date**

The H to-date index or J to-date index is set to point at the current, month-to-date, quarter-to-date, or year-to-date fields.

The operand field must contain an H or J followed by a 1 (current), 2 (month-to-date), 3 (quarter-to-date), or 4 (year-to-date).

Before control is given to a report generator program, the H to-date index and J to-date index are set to agree with the type of data requested in columns 11-14 of the D transaction.

The H to-date index and J to-date index are only used when fields 149-150 and fields 117-124 are referenced.

Also see SHx and SJx instructions, which are preferred.

**AX1 (024) Set all indexes to 1**

All of the indexes (excluding the to-date indexes) are set to point to the first occurrence of their respective segments.

During a company or tax pass, only the B and C indexes are affected.

**EXC (051) Save the index**

The current value of the index specified in the operand field is swapped with a saved value.

There are, in fact, 2 indexes associated with each segment type. An AX1 or IFF instruction sets both of the groups of indexes to 1.

All of the other instructions (except EXC, AX1, and IFF) that implicitly or explicitly reference indexes only reference the current value.

Thus, for example, if the H index is pointing to a deduction and it is necessary to access a field in HED 001 and then set the H index to again point at the deduction, the following could be written:

EXCH	Exchange current and saved H index
1STH	Set H index to 1 (first occurrence)
FL0134	Move regular hourly rate to extract record
EXCH	Exchange current and saved H index again

**NET (058) Set HED to net pay**

The H index is set to point to HED 999.

*Note: Certain types of employee records may not contain an entry for HED 999 (such as a labor record or possibly adjustment record). The HED Number (field 130) should be compared to 999 to verify that the H index is pointing the proper occurrence.*

**FC (073) Set header HED to category**

2 is subtracted from the output Position Pointer

The B index is incremented to point to the next HED. If there are no more company HEDs, control passes to the instruction following the FC.

HED Category (field 051) is compared to the 2 characters last moved to the extract record. If they are not equal this instruction reverts to the preceding step above.

If they are equal, control is transferred down to the next paragraph named 255.

**SHL (082) Set header and employee HED to last earnings**

The B index and the H index are set to point to the last earnings in the B and H segments respectively.

**SBL (083) Set header HED to last earnings**

The B index is set to point to the last earnings in the B segment.

**FST (084) Set employee tax to first state**

The J index is set to point to the first state tax body for this employee.

Because an employee may not have a state tax body set up, it is best to compare the Tax Type (field 101) to a 2 after an FST instruction to verify that the J index is indeed pointing to a state tax body occurrence.

**HUP (140) Set H To-date up 1**

The H to-date index is incremented to point to the next higher-level occurrence of fields 149 and 150.

Thus, for example, if the H to-date index is pointing to the month-to-date fields, this instruction causes it to point to the quarter-to-date fields.

Incrementing the H to-date index past the year-to-date fields will cause unpredictable results.

**SHC (141) Set H To-Date to current**

The H to-date index is set to point to the specified fields.

**SHM (142) Set H To-date to month-to-date**

The H to-date index is set to point to the specified fields.

**SHQ (143) Set H To-date to quarter-to-date**

The H to-date index is set to point to the specified fields.

**SHY (144) Set H To-date to year-to-date**

The H to-date index is set to point to the specified fields.

**JUP (145) Set J To-date up 1**

The J to-date index is incremented to point to the next higher-level occurrence of fields 117-124.

Incrementing the J to-date index past the year-to-date fields will cause unpredictable results.

**SJC (146) Set J To-date to current**

The J to-date index is set to point to the specified fields.

**SJM (147) Set J To-date to month-to-date**

The J to-date index is set to point to the specified fields.

**SJQ (148) Set J To-date to quarter-to-date**

The J to-date index is set to point to the specified fields.

**SJY (149) Set J To-date to year-to-date**

The J to-date index is set to point to the specified fields.

**Miscellaneous instructions**

**WRT (005) Write to extract file**

This instruction results in the extract record being written to the report file (P400UT), and it also causes the Output Position Pointer to be set to zero.

Currently, the WCR and WRT instructions operate identically. To insure proper operation in the future, however, use of the WCR instruction should be limited to report generators that are producing payments.

**WCR (006) Write payments**

This instruction results in the extract record being written to the report file (P400UT), and it also causes the Output Position Pointer to be set to zero.

Currently, the WCR and WRT instructions operate identically. To insure proper operation in the future, however, use of the WCR instruction should be limited to report generators that are producing payments.

**WRR (088) Write to recycled file**

This instruction results in the extract record being written to the report file (P400UT), and it also causes the Output Position Pointer to be set to zero.

The WRR instruction moves the first 149 positions of the extract record to positions 2–150 of the output area, sets position 1 of the output area to an 8, and writes the record to the extract file.

**TRC (098) Trace**

This instruction results in the extract record being written to the report file (P400UT).

The TRC instruction is intended for use as a debugging aid. It sets the Forms Code to high values and inserts an address into the last 5 positions of the extract record prior to writing it.

This instruction is treated as a no-op, however, unless the output master is suppressed by using an H2 transaction with no output master option.

### **XIT (007) Exit report**

Control is returned to P4CALC.

An XIT instruction is automatically inserted by the system after the last instruction (R7 transaction).

### **XS (015) Exit sort**

Control is transferred to the instruction immediately following the last BSK instruction that was executed.

An XS instruction is automatically inserted by the system after the last sort instruction (S7 transaction).

This instruction should not be used on an R7 transaction.

### **CAD (048) Condense address**

The CAD instruction compares fields 088 and 089 (Address Line 1 and 2) to blanks.

If field 088 is blank, field 089 is moved up to it and fields 090 and 091 are moved to 089.

If field 089 is blank, fields 090 and 091 are moved to it.

If CAD is used, it must be followed by RAD, within the generator.

A common use of these instructions would be in a generator that is producing mailing labels. Consider the following set of instructions:

CAD	Condense address
FLD087	Move NAME to extract record
FML	Reverse name
FLD088	Move ADDRESS LINE 1 to extract record
FLD089	Move ADDRESS LINE 2 to extract record
FLD090	Move CITY and STATE/PROVINCE to extract record
FLD091	Move ZIP CODE/POSTAL CODE to extract record
RAD	Restore address

If an employee did not have an Address Line 2, the City, State/Province and ZIP Code/Postal Code would print on the third line and the fourth line would be blank.

If the CAD and RAD were not used, some of the labels would have an embedded blank line.

**RAD (049) Restore address**

The RAD instruction reverses the action of CAD.

A common use of these instructions would be in a generator that is producing mailing labels. Consider the following set of instructions:

CAD	Condense address
FLD087	Move NAME to extract record
FML	Reverse name
FLD088	Move ADDRESS LINE 1 to extract record
FLD089	Move ADDRESS LINE 2 to extract record
FLD090	Move CITY and STATE/PROVINCE to extract record
FLD091	Move ZIP CODE/POSTAL CODE to extract record
RAD	Restore address

If an employee did not have an Address Line 2, the City, State/Province and ZIP Code/Postal Code would print on the third line and the fourth line would be blank.

If the CAD and RAD were not used, some of the labels would have an embedded blank line.

**IFF (085) Select employee if frequency**

Field 314 (Report Frequency) is tested.

If it is blank, an AX1 is executed.

If it is numeric, it is compared to field 071 (employee Frequency).

If they are equal an AX1 is executed; otherwise, an XIT is executed.

If it is equal to an F, the AJ transaction for the employee's frequency is checked to see if his frequency is being paid.

If it is, an AX1 is executed; otherwise, an XIT is executed.

This instruction should only be used when a multiple or permanent master record is being processed. For other record types it will always behave as an XIT instruction.

The results of this instruction are unpredictable if a pay run is not being made.

### **CAL (197) Call another generator**

This instruction allows the user to call another generator and treat it as a sub-routine of the current generator.

This called generator may not contain another CAL instruction.

In other words, nested calls are not allowed.

The generator called must use Sort 5G and must be written expressly as a subroutine.

This instruction may not be used as part of an online generator because Sort 5H already contains a CAL instruction.

The name of the generator to be called must be placed in field 852 prior to the call, as follows:

```
LIT045GLX      MOVE LITERAL
MTW852         MOVE CALL CODES
CAL            CALL GENERATOR
```

The MTW instruction will Reset the LIT04 automatically.

When the called generator returns control to this generator the next instruction executed will be the one immediately following the CAL instruction.

### **RET (198) Return to calling generator**

This instruction is used in place of the XIT instruction in a generator called by another generator. (See CAL.)

It will return control to the calling generator.

If no CAL is in effect, this will be treated the same as XIT.

### **WAT (205) Write audit trail**

This instruction may be used by any report generator to write the first 132 positions of the extract record area to the audit trail.

## Additional Instructions

### **NPA (216) Net Pay Available**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 is other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **REG (217) Regular Pay Amount (all category 05 and 15 earnings)**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 is other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **OVT (218) Overtime Pay Amount (all category 01 earnings)**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 is other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **SPL (219) Special Pay amount (all categories except 01, 05, and 15)**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 is other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **VAC (229) Vacation Pay Amount (all category 02 and 03 earnings)**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 if other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **AP (221) Account Payable amount (all taxability type code 16 earnings)**

This instruction is functional only in Method Code Generators, not Report Generators.

The amounts described above may be accessed directly in Method Code generators only without looping through HEDs to gather this information.

These are 'point in time' instructions. For example, if the NPA is interrogated at HED 501 it will contain a different number than at HED 800 if other deductions were taken between those two deductions.

Execution of any of the above instructions will place the described number in the extract record in counter format.

### **Instructions to access FILE01**

#### **BF1 (222) Browse FILE01. Find and read a record from FILE01**

This instruction allows report generators to access FILE01.

The last field moved to the extract record is used as a key in a Start and Read of FILE01.

If the record is found, it is placed in FLD860.

If the record is not found control is passed to the next 255 paragraph.

The Current field Size (the size of the key) is subtracted from the Output Position Pointer, effecting an automatic reset of the key field.

#### **RF1 (223) Read FILE01. Read the next FILE01 record.**

This instruction allows report generators to access FILE01.

The next record on FILE01 is read and placed into FLD860.

#### **EX1 (224) Exchange FILE01. Swap FILE01 records.**

This instruction allows report generators to access FILE01.

The last FILE01 record read is swapped with a saved record.

A P P E N D I X B

## Sample Report Generator

---

### In This Appendix

Sample Report Generator.....138

# Sample Report Generator

## Introduction to Sample Report Generator

The following example provides a step-by-step sample of how to use the Report Generator.

### Report request transaction

The D transaction or the DD-SCR form (Report Request Form) is the request transaction that must be entered to produce a report for an Organization 1-2. If a report is to be produced for more than one Organization 1-2, a D transaction or DD-SCR form (Report Request Form) must be entered for each Organization 1-2.

### Sort format

The sort format (or sequence) is the routine or logic that physically builds the fields on the front of the extract record so that they will be printed in sequence after the report records are sorted by the P45SORT.

The S1 and S7 transactions are used to define the sort format. Columns 3–4 of each S1 and S7 transaction will contain the 2 position Sort Code that is referred to in columns 3–4 of the D transaction.

### Report format

The report format is used to define the headings and detail lines of the actual printed report or file output.

The RO through R7 transactions are used to define the format and the Format Code is in columns 3–4 of each of the RO–R7 transactions. To request the report, the Format Code is placed in columns 5–6 of the D transaction.

### Process control transaction

The process control (AE transaction) is used to trigger the report requests that have been submitted by Organization 1-2 on the D transactions.

The Report Generate field (02–3) is used to select reports at the end of a pay period, month, quarter, or year.

The Report Selects (field 034) is used to select special reports. Up to 12 different report requests may be entered.

This example will create a turn-around time document each week for reporting the next week's time.

The D transaction below shows that the Sort Code (column 3–4) will be T1, and the Format Code (columns 5-6) will be T1.

The Report Select (column 7) indicates that the report will be produced every payroll cycle.

Column 14 (Report Year) indicates that the logic or R7 transactions will be performed each time year-to-date fields are found on the database. This is done because, regardless of the number of payments or adjustments for an employee, there is only one set of year-to-date fields for each employee. Therefore, the R7 logic will only be executed once for each employee.





The moves begin with the first character after the record type (or the sort key plus 1) and progress across the record from left to right; every field beyond the record type in the extract record must be accounted for in the R3 and R4 moves.

Move 1	The first character of the record is moved to print position 000. Any time a character is moved to print position zero it will be used as the carriage control. Blank = single space 0 = double space - = triple space 1 = top for form 1-9 = skip down to a line number specified on the R0 transaction
Move 2	The next 30 characters (employee name) will be printed beginning in print position 1.
Move 3	The next 12 characters (social security number) will be printed beginning in print position 36.
Move 4	Because the field length and print position are both blank, the system will skip to a new print line.
Move 5	The next character is moved to print position zero for carriage control on the new print line.
Move 6	The next 4 characters (organization 3—department number) will be printed beginning in print position 18.
Move 7	The next 10 characters (employee number) will be printed beginning in print position 18.
Move 8	Skip to a new print line (see move 4).
Move 9	Carriage control (see move 5).
Move 10	The next 8 characters (period ending date) will be printed beginning in print position 12.

1	1	2	2	3	3	4	4	5	5	6// 7
1...	5...	0...	5...	0...	5...	0...	5...	0...	5...	0//.5.
R3T1002010003000112056      010000400810018      0100008012										

Position	Description
1-2	Transaction type
3-4	Format
5-7	Line number
8-9	Move 1 length
10-12	Move 1 print position
13-14	Move 2 length
15-17	Move 2 print position
18-19	Move 3 length

Position	Description
20–22	Move 3 print position
23–24	Move 4 length
25–27	Move 4 print position
28–29	Move 5 length
30–32	Move 5 print position
33–34	Move 6 length
35–37	Move 6 print position
38–39	Move 7 length
40–42	Move 7 print position
43–44	Move 8 length
45–47	Move 8 print position
48–49	Move 9 length
50–52	Move 9 print position
53–54	Move 10 length
55–57	Move 10 print position

### Report instructions

The report instructions (R7 transactions) provide the logic to select certain employees, make calculations, compare fields, and create the output report record that will be sorted by P45SORT and printed by P5PRNT. The R7 instructions must:

- Build the sort key on the front of the record by executing the sort (the BSK instruction).
- Build the extract record by moving master file fields, counters, or literals.

Write the extract record (the 'WRT' instruction).

Instruction	Action
Instruction 1	This instruction sets all of the indexes to 1 and should always be done before attempting to access any fields in an employee record.
Instruction 2	Field 072 (pay types) is being moved to the extract record.
Instruction 3	A literal of 4 is being moved to the extract record.
Instruction 4	Field 072 and the literal are compared. If field 072 is less, control is passed down to the 255 paragraph at instruction 6.
Instruction 5	If pay type is 4 or more (an automatically paid employee), the report is exited.
Instruction 6	This is a paragraph label. Control will be transferred here if the pay type is less than 4.
Instruction 7	Field 071 (pay frequency) is being moved to the extract record.
Instruction 8	A literal of 1 is being moved to the extract record.

<b>Instruction</b>	<b>Action</b>
Instruction 9	Field 071 and the literal are compared. If they are equal (weekly employee) control is passed down to the 255 paragraph at instruction 11.
Instruction 10	If the pay frequency is not 1 the report is exited.
Instruction 11	This is a paragraph label. Control will be transferred here if the pay frequency is 1.
Instruction 12	This instruction builds the sort key by performing the instructions on the S7 transactions.
Instruction 13	A literal of 0 is moved to the extract record. The first position following the sort key is the record type. A record type of zero or any alphabetic character indicates that the final formatting of the extract record will be done using the R3/R4 transaction data. A record type of 1-7 signifies that the R5/R6 transaction data will be used.
Instruction 14	A literal of 1 is moved to the extract record, which will be used for carriage control.
Instruction 15	Field 087 (employee name) is being moved to the extract record.
Instruction 16	Field 070 (Social Security Number) is being moved to the extract record.
Instruction 17	A literal of '-' is moved to the extract record. The blanks in move 4 of the R3 transaction indicate that a new print line is to be used. The '-' will be used for carriage control (triple space).
Instruction 18	Field 094 (organization 3) is being moved to the extract record.
Instruction 19	Field 003 (employee number) is being moved to the extract record.
Instruction 20	See Instruction 17.
Instruction 21	Field 138, which contains the period end date, is being moved to the extract record.
Instruction 22	A literal of 14 is being moved to the extract record.
Instruction 23	The literal of 14 is being added to the weekly period end date to form a date that is 14 days later.
Instruction 24	The date is being edited from its internal YYMMDD format to MM/DD/YY.
Instruction 25	The completed extract record is written to the output file.

```

      1      1      2      2      3      3      4      4      5 // 7
1...5...0...5...0...5...0...5...0...5...0...5...0...//.5.
R7T1001AX1
R7T1002FLD072
R7T1003LIT014
R7T1004BL
R7T1005XIT
R7T1006255
R7T1007FLD071
R7T1008LIT011
    
```

```
R7T1009BE
R7T1010XIT
R7T1011255
R7T1012BSK
R7T1013LIT010
R7T1014LIT011
R7T1015FLD087
R7T1016FLD070
R7T1017LIT01-
R7T1018FLD094
R7T1019FLD003
R7T1020LIT01-
R7T1021FLD138
R7T1022LIT0214
R7T1023ADA
R7T1024MDY
R7T1025WRT
```

Position	Description
1-2	Transaction type
3-4	Format
5-7	Line number
8-10	Instruction number
11-34	Contents

APPENDIX C

## Sample Field Table

---

### In This Appendix

Sample Field Number Table .....146

# Sample Field Number Table

FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
001			CONTROL 1	ALPHA-NUMERIC	2	8
002			CONTROL 2	ALPHA-NUMERIC	4	8
003			EMPLOYEE NUMBER	ALPHA-NUMERIC	10	8
004	E		RECORD GROUP	ALPHA-NUMERIC	2	8
005	A	AA	CONTROL 1 NAME	ALPHA-NUMERIC	30	8
006	A	AA	CONTROL 2 NAME	ALPHA-NUMERIC	30	8
007	A	AB	CONTROL ADDR 1	ALPHA-NUMERIC	30	8
008	A	AB	CONTROL ADDR 2	ALPHA-NUMERIC	30	8
009	A	AC	CONTROL CITY ST	ALPHA-NUMERIC	25	8
010	A	AC	CONTROL ZIP	NUMERIC	5	8
011	A	AD	CONTROL 1 TITLE	ALPHA-NUMERIC	12	8
012	A	AD	CONTROL 2 TITLE	ALPHA-NUMERIC	12	8
013	A	AD	CONTROL 3 TITLE	ALPHA-NUMERIC	12	8
014	A	AD	CONTROL 4 TITLE	ALPHA-NUMERIC	12	8
015	A	AD	CONTROL 5 TITLE	ALPHA-NUMERIC	12	8
016	A	AD	CONTROL 6 TITLE	ALPHA-NUMERIC	12	8
017			NOT YET IN USE			
018			NOT YET IN USE			
019			NOT YET IN USE			
020			NOT YET IN USE			
021			NOT YET IN USE			
022			NOT YET IN USE			
023			NOT YET IN USE			
024			NOT YET IN USE			
025			NOT YET IN USE			
026			NOT YET IN USE			
027	A	AE	CLEAR TO DATE	NUMERIC	1	8
028	A	AE	REPORT GENERATE	NUMERIC	1	8
029	A	AE	PURGE TERMINATE	NUMERIC	1	8
030	A	AE	RUN DATE	DATE YYMMDD	6	8
031	A	AE	RUN TYPE	NUMERIC	1	8
032	A	AE	VERSION NUMBER	NUMERIC	2	8
033	A	AE	PRINT UPDATE	ALPHA-NUMERIC	1	8
034	A	AE	REPORT SELECTS	ALPHA-NUMERIC	12	8
035	A	AE	USER DATE	DATE YYMMDD	6	8
036	A	AE	USER FIELD	ALPHA-NUMERIC	4	8
037	A	AE	RESERVED	ALPHA-NUMERIC	3	8
038	A		HDR PERIOD 1		6	8
039	A		HDR PERIOD 2		6	8
040			NOT YET IN USE			
041			NOT YET IN USE			
042			NOT YET IN USE			
043	C		TRANS CODE	ALPHA-NUMERIC	2	8
044	E		ADJUSTMNT BATCH	ALPHA-NUMERIC	1	8
045	L		L CHANGE	ALPHA-NUMERIC	1	
046			UNKNOWN FIELD	ALPHA-NUMERIC	15	8

Field Number Table

FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
--------------	----------	------------------	-------------	---------------	--------------	-----------

047			PLUG CODE	NUMERIC	1	8
048	E		ACTION COD	ALPHA-NUMERIC	1	8
049	E	EA	TO D	ALPHA-NUMERIC	1	8
050	B	A8	HED NUMBER	NUMERIC	3	8
051	B	A8	HED CATEGORY	NUMERIC	2	8
052	B	A8	HED DES	ALPHA-NUMERIC	15	8
053	B	A8	HED VACATION	NUMERIC	1	8
054	B	A8	HED PERM ORDER	NUMERIC	3	8
055	B	A8	HED TEMP ORDER	NUMERIC	3	8
056	B	A8	HED TC 2 HOURS	NUMERIC	1	8
057	B	A8	HED TC 2 MONEY	NUMERIC	1	8
058	B	A8	USE IN OT RATE	NUMERIC	1	8
059	B	A8	FREQ F	ALPHA-NUMERIC	1	8
060	B	A8	DEFAULT FREQ	NUMERIC	2	8
061	B	A8	DEFAULT TYPE	NUMERIC	2	8
062	B	A8	DEFAULT	ALPHA-NUMERIC	2	8
063	B	A8	DEFAULT AMT PCT	NUMERIC	7	8
064	B	A8	HED A	ALPHA-NUMERIC	1	8
065	B	A8	PERIOD TABLE	NUMERIC	1	8
066	B	A8	ADD TOTAL HOURS	NUMERIC	1	8
067	B	A8	HED REGISTER	ALPHA-NUMERIC	1	8
068	B	A8	HED TIMING	ALPHA-NUMERIC	1	8
069	E		E TRANS CODE	ALPHA-NUMERIC	2	8
070	E	E	SOCIAL SEC NBR	ALPHA-NUMERIC	12	8
071	E	E	PAY FREQUENCY	ALPHA-NUMERIC	1	8
072	E	E	PAYMENT CODE	NUMERIC	1	8
073	E	E	STATUS CODE	ALPHA-NUMERIC	2	8
074	E	E	SEX CODE	ALPHA-NUMERIC	1	8
075	E	E	EQUAL OPPORTNTY	ALPHA-NUMERIC	2	8
076	E	E	UNION CODE	ALPHA-NUMERIC	5	8
077	E	E	WORKERS COMP	ALPHA-NUMERIC	5	8
078	E	E	BIRTH DATE	DATE YYMMDD	6	8
079	E	E	EMPLOYMENT CODE	ALPHA-NUMERIC	2	8
080	E	E	EMPLOYMENT DATE	DATE YYMMDD	6	8
081	E	E	TERMINATE CODE	ALPHA-NUMERIC	2	8
082	E	E	TERMINATE DATE	DATE YYMMDD	6	8
083	E	E	NORMAL SHIFT	ALPHA-NUMERIC	1	8
084	E	E	SPLIT METHOD	NUMERIC	1	8
085	E	E	JOB CATEGOR	ALPHA-NUMERIC	4	8
086	F	F1	ADDRESS CODE	NUMERIC	3	8
087	F	F1	ADDRESS NAME	LAST FIRST	30	F
088	F	F1	ADDRE	ALPHA-NUMERIC	30	F
089	F	F2	ADDR	ALPHA-NUMERIC	30	F
090	F	F2	ADDRESS	ALPHA-NUMERIC	25	F
091	F	F2	ADDRESS ZIP	NUMERIC	5	F
092	G	G	SPLIT NUMBER	NUMERIC	2	8
093	G	G	SPLIT PERCENT	NUMERIC	5	8
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
094	G	G		ALPHA-NUMERIC	4	G
095	G	G	CONT	ALPHA-NUMERIC	4	G
096	G	G	CONTROL 5	ALPHA-NUMERIC	4	G
097	G	G	CONTROL 6	ALPHA-NUMERIC	4	G
098	G	G	FUNCTION	ALPHA-NUMERIC	10	8
099	J		FEDERAL TAX NBR	NUMERIC	2	8
100	J		TAX STATE NBR	ALPHA-NUMERIC	2	8
101	J	J	TAX TYPE	NUMERIC	1	8

102	J	J	TAX NUMBER	ALPHA-NUMERIC	6	8
103	J	J	TAX METHOD	ALPHA-NUMERIC	1	8
104	J	J	TAX RESIDENCE	NUMERIC	1	8
105	J	J	TAX DEPENDENTS	NUMERIC	2	8
106	J	J	TAX MARITAL	NUMERIC	1	8
107	J	J	TAX AMOUNT PCT	NUMERIC	7	8
108	J	J	TAX UNEMP INSUR	NUMERIC	1	8
109	J	J	TAX PLEDGE PCT	NUMERIC	5	8
110	J	J	TAX RECIP CODE	ALPHA-NUMERIC	1	8
111	J	J	TAX RECIP STATE	ALPHA-NUMERIC	2	8
112	J	J	TAX TABLE KEY	NUMERIC	1	8
113	J	J	WORK FLAG	NUMERIC	1	8
114	J	J	STATE FLSA FLAG	ALPHA-NUMERIC	1	8
115	J	J	TAX COUNTY CODE	ALPHA-NUMERIC	6	8
116	J	J	TAX CITY CODE	ALPHA-NUMERIC	6	8
117	J	KD	TAX WORK IN	COMPUTATIONAL	9	8
118	J	KD	TAX LIVE IN	COMPUTATIONAL	9	8
119	J	KD	TAX WAGES	COMPUTATIONAL	9	8
120	J	KD	TAX AMOUNT	COMPUTATIONAL	9	8
121	J	KD	TAX UNEMPLY	COMPUTATIONAL	7	8
122	J	KD	TAX FICA/SDI	COMPUTATIONAL	5	8
123	J	KD	TAX PREMIUM	COMPUTATIONAL	6	8
124	J	KD	TAX WEEKS	COMPUTATIONAL	4	8
125			NOT YET IN USE			
126	E	EA	PERIOD DATE	DATE YYMMDD	6	8
127	E	EA	RECON DATE	DATE YYMMDD	6	8
128			DATA-TYPE	ALPHA-NUMERIC	1	8
129	E	E	CLEAR ANNIVRSRY	ALPHA-NUMERIC	1	8
130	H	H	HED NUMBER	NUMERIC	3	8
131	H	H	HED FREQUENCY	NUMERIC	2	8
132	H	H	HED TYPE CODE	NUMERIC	2	8
133	H	H	HED METHOD CODE	ALPHA-NUMERIC	2	8
134	H	H	HED AMOUNT PCT	NUMERIC	7	H
135	H	H	HED START CODE	NUMERIC	1	8
136	H	H	HED START FIELD	NUMERIC	6	8
137	H	H	HED STOP CODE	ALPHA-NUMERIC	1	8
138	H	H	HED STOP FIELD	NUMERIC	6	8
139	H	H	HED ONCE CODE	NUMERIC	1	8
140	H	H	HED ONCE FIELD	NUMERIC	7	8
141	H	H	HED AMOUNT ONE	NUMERIC	7	H
Field Number Table						
FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
142	H	H	HED AMOUNT TWO	NUMERIC	8	H
143	H	H	HED USER CODE	ALPHA-NUMERIC	2	8
144	H	H	HED ACCOUNT NBR	ALPHA-NUMERIC	11	8
145	E	EA	MASTER NUMBER	NUMERIC	4	8
146	E	EA	BANK CODE	ALPHA-NUMERIC	2	8
147	E	EA	RECON NUMBER	NUMERIC	8	8
148	E	EA	RECON CLEAR	ALPHA-NUMERIC	1	8
149	H	KB	HED AMOUNT	COMPUTATIONAL	9	8
150	H	KB	HED HOURS	COMPUTATIONAL	7	8
151	D	D	REPORT CODE	ALPHA-NUMERIC	4	8
152	D	D	REPORT SELECT	ALPHA-NUMERIC	1	8
153	D	D	REPORT PLUS	NUMERIC	1	8
154	D	D	REPORT MINUS	NUMERIC	1	8
155	D	D	REPORT MANUAL	NUMERIC	1	8
156	D	D	REPORT CURRENT	NUMERIC	1	8

157	D	D	REPORT MONTH	NUMERIC	1	8
158	D	D	REPORT QUARTER	NUMERIC	1	8
159	D	D	REPORT YEAR	NUMERIC	1	8
160	D	D	REPORT HEADERS	NUMERIC	1	8
161	D	D	REPORT TAXES	NUMERIC	1	8
162	D	D	REPORT LABOR	NUMERIC	1	8
163	D	D	REPORT OTHER	NUMERIC	1	8
164	D	D	USER D FIELD	ALPHA-NUMERIC	4	8
165	R	R1	FORMAT CODE	ALPHA-NUMERIC	2	8
166	R	R1	SEQUENCE NUMBER	NUMERIC	3	8
167	R	R1	PRINT HEADER 1	ALPHA-NUMERIC	1	8
168	R	R1	PRINT HEADER 2	ALPHA-NUMERIC	1	8
169	R	R1	PRINT HEADER 3	ALPHA-NUMERIC	1	8
170	R	R1	PRINT HEADER 4	ALPHA-NUMERIC	1	8
171	R	R1	LINES PER PAGE	NUMERIC	2	8
172	R	R1	HEADER 1 SPACE	ALPHA-NUMERIC	1	8
173	R	R1	HEADER 2 SPACE	ALPHA-NUMERIC	1	8
174	R	R1	HEADER 3 SPACE	ALPHA-NUMERIC	1	8
175	R	R1	HEADER 4 SPACE	ALPHA-NUMERIC	1	8
176	R	R1	AFTER HDR SPACE	ALPHA-NUMERIC	1	8
177	R	R1	WRITE EXTRACT	ALPHA-NUMERIC	1	8
178	R	R1	PRT CONTROL 1-2	ALPHA-NUMERIC	1	8
179	R	R1	PRINT VERSION	ALPHA-NUMERIC	1	8
180	R	R1	PRINT PAGE	ALPHA-NUMERIC	1	8
181	R	R1	PRINT DATE-TIME	NUMERIC	1	8
182	R	R1	PRINT PERIOD	NUMERIC	1	8
183			NOT YET IN USE			
184			NOT YET IN USE			
185	R	R1	PRINT DETAIL 1	ALPHA-NUMERIC	1	8
186	R	R1	PRINT DETAIL 2	ALPHA-NUMERIC	1	8
187	R	R1	LINE USED	ALPHA-NUMERIC	1	8
188	R	R1	LINE DESC	ALPHA-NUMERIC	5	8
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
189	R	R7	COUNTER NUMBER	NUMERIC	2	8
190			FIELD NUMBER	NUMERIC	3	8
191	R	R5	SET TODATE	NUMERIC	1	8
192	R	R2	HEADER NUMBER	NUMERIC	1	8
193	R	R2	PRINT POSITION	NUMERIC	3	8
194	R	R3	FIELD LENGTH	NUMERIC	2	8
195	R	R3	PRINT POSITION	NUMERIC	3	8
196	R	R7	SEGMENT TYPE	ALPHA-NUMERIC	1	8
197	R	R7	ROUTINE CODE	ALPHA-NUMERIC	3	8
198	D		SORT CODE	ALPHA-NUMERIC	2	8
199	S	S1	SORT KEY LENGTH	NUMERIC	2	8
200	S	S1	KEY POSITION	NUMERIC	2	8
201	S	S1	TOTAL HEADING	ALPHA-NUMERIC	15	8
202	S	S1	STAGGER TOTALS	ALPHA-NUMERIC	1	8
203	S	S1	PAGE ADVANCE	NUMERIC	1	8
204	S	S1	PAGE ZERO AFTER	ALPHA-NUMERIC	1	8
205	S	S1	LINE ADV BEFORE	ALPHA-NUMERIC	1	8
206	S	S1	LINE ADV AFTER	ALPHA-NUMERIC	1	8
207	S	S1	HIGHEST TOTAL	NUMERIC	1	8
208			NOT YET IN USE			
209			NOT YET IN USE			
210			NOT YET IN USE			
211			NOT YET IN USE			

212			BATCH CODE	ALPHA-NUMERIC	4	8
213			TRANS COUNT	NUMERIC	4	8
214			REGULAR HOURS	NUMERIC	6	8
215			REG RATE SALARY	NUMERIC	7	8
216			OVERTIME CODE	NUMERIC	1	8
217			OVERTIME HOURS	NUMERIC	4	8
218			OVT RATE SALARY	NUMERIC	6	8
219			TC HED NUMBER	NUMERIC	3	8
220			PERIOD END MNTH	NUMERIC	2	8
221			PERIOD END DAY	NUMERIC	2	8
222			TC TAX TYPE	ALPHA-NUMERIC	1	8
223			OVERRIDE FREQ	ALPHA-NUMERIC	1	8
224			REGULAR HOURS	NUMERIC	5	8
225			OVERTIME CODE	NUMERIC	1	8
226			OVERTIME HOURS	NUMERIC	4	8
227			HOURS 1	NUMERIC	4	8
228			HOURS 2	NUMERIC	4	8
229			HOURS 3	NUMERIC	4	8
230			HOURS 4	NUMERIC	4	8
231			TIME CARD RATE	NUMERIC	6	8
232			TIME AMOUNT	NUMERIC	6	8
233			NOT YET IN USE			
234			HOURS GROUP	NUMERIC	1	8
235	T		TAX BODY	ALPHA-NUMERIC	7	8
236	T	T1	TAX DESCRIPTION	ALPHA-NUMERIC	15	8
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
237	T	T1	FILING NUMBER	ALPHA-NUMERIC	25	8
238	T	T1	UNEMPLOY NUMBER	ALPHA-NUMERIC	25	8
239	T	T1	W2 FORM NUMBER	ALPHA-NUMERIC	5	8
240	T	T1	USE TAX BODY	ALPHA-NUMERIC	1	8
241	T	T2	DISABILITY RATE	COMPUTATIONAL	7	8
242	T	T2	DEFAULT WTH PCT	COMPUTATIONAL	7	8
243	T	T2	UNEMPLOY RATE	COMPUTATIONAL	7	8
244	T	T2	TAX MISC FIELD	COMPUTATIONAL	9	8
245	T	T2	TAX MISC2 FIELD	NUMERIC	1	8
246			NOT YET IN USE			
247			NOT YET IN USE			
248			NOT YET IN USE			
249			NOT YET IN USE			
250	T	T3	MAX UNEMPL WAGE	COMPUTATIONAL	9	8
251	T	T3	MAX DISAB WAGE	COMPUTATIONAL	9	8
252	T	T3	MAX DISAB TAX	COMPUTATIONAL	7	8
253	T	T3	DISAB RECIP	NUMERIC	1	8
254	T	T3	UNEMPLOY RECIP	NUMERIC	1	8
255	T	T3	RES SNGL TABLE	NUMERIC	1	8
256	T	T3	RES MAR TABLE	NUMERIC	1	8
257	T	T3	RES HEAD TABLE	NUMERIC	1	8
258	T	T3	NON SNGL TABLE	NUMERIC	1	8
259	T	T3	NON MAR TABLE	NUMERIC	1	8
260	T	T3	NON HEAD TABLE	NUMERIC	1	8
261	T	T3	TAX TABLE 7	NUMERIC	1	8
262	T	T3	TAX TABLE 8	NUMERIC	1	8
263	T	T3	TAX TABLE 9	NUMERIC	1	8
264	T	T3	TAX CLASS	NUM NOT JUST	9	8
265	T	T3	TAX FLAGS	ALPHA-NUMERIC	3	8
266	T	T4	TABLE NUMBER	NUMERIC	1	8

267	T	T4	EXTRA EXEMPTION	NUMERIC	1	8
268	T	T4	USE FEDERAL	NUMERIC	1	8
269	T	T4	EXEMPTION TYPE	NUMERIC	1	8
270	T	T4	MAX TAX FREQ	NUMERIC	1	8
271	T	T4	TAX CREDITS	NUMERIC	1	8
272	T	T4	MAX EXEMPT AMT	COMPUTATIONAL	7	8
273	T	T4	MAX TAX WITHELD	COMPUTATIONAL	7	8
274	T	T4	EXEMPTION PCT	COMPUTATIONAL	7	8
275	T	T4	MAX DEPENDENTS	NUMERIC	2	8
276	T	T4	DEPENDENT AMT 1	COMPUTATIONAL	4	8
277	T	T4	DEPENDENT AMT 2	COMPUTATIONAL	4	8
278	T	T4	DEPENDENT AMT 3	COMPUTATIONAL	4	8
279	T	T4	DEPENDENT AMT R	COMPUTATIONAL	4	8
280	T	T4	TAX CREDIT 1ST	COMPUTATIONAL	4	8
281	T	T4	TAX CREDIT 2ND	COMPUTATIONAL	4	8
282	T	T4	TAX CREDIT 3RD	COMPUTATIONAL	4	8
283	T	T4	TAX CREDIT REST	COMPUTATIONAL	4	8
Field Number Table						
FIELD NUMBER	SEGMENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
284	T	T4	MIN EXEMPT AMT	COMPUTATIONAL	4	8
285	T	T4	MIN WAGE TAXED	COMPUTATIONAL	4	8
286			NOT YET IN USE			
287			NOT YET IN USE			
288	T	T5	BRACKET NUMBER	NUMERIC	2	8
289	T	T5	BRACKET WAGES	COMPUTATIONAL	9	8
290	T	T5	BRACKET TAX	COMPUTATIONAL	9	8
291	T	T5	BRACKET PERCENT	COMPUTATIONAL	7	8
292	T	T5	BRACKET EXCLUDE	COMPUTATIONAL	9	8
293			NOT YET IN USE			
294			NOT YET IN USE			
295	E	EA	ROUTING CODE	ALPHA-NUMERIC	9	8
296	L		USER TRANSACTION CODE	ALPHA-NUMERIC	2	8
297			FORMS CODE	ALPHA-NUMERIC	1	
298		P6	LOW RECON	NUMERIC	8	8
299		P6	HIGH RECON	NUMERIC	8	8
300		P6	PRINT TYPE	ALPHA-NUMERIC	1	8
301	C	AF	PAYER BANK CODE	ALPHA-NUMERIC	2	8
302	C	AF	RESERVED	ALPHA-NUMERIC	8	8
303	C	AF	FLSA METHOD	ALPHA-NUMERIC	1	8
304	C	AF	PAYMENT RECON	ALPHA-NUMERIC	1	8
305	C	AF	DEFAULT UNEMPLY	NUMERIC	1	8
306	C	AF	HISTORY CONTENT	NUMERIC	1	8
307	C	AF	PRINT BRACKETS	ALPHA-NUMERIC	1	8
308	C	AF	ON-LINE FILE	ALPHA-NUMERIC	1	8
309	C	AF	OVERTIME SHIFT	NUMERIC	1	8
310	C	AF	COMPNY CATEGORY	ALPHA-NUMERIC	2	8
311	C	AF	PAY RAISE SPLIT	ALPHA-NUMERIC	1	8
312	C	AF	PAY FREQ SET UP	NUMERIC	1	8
313	C	AF	CLEAR ALL FREQ	ALPHA-NUMERIC	1	8
314	C	AF	REPORT FREQ	ALPHA-NUMERIC	1	8
315	C	AF	DROP LABOR	NUMERIC	2	8
316	C	AF	DROP HISTORY	NUMERIC	2	8
317	C	AF	COUNTRY	NUMERIC	1	8
318	C	AF	NO PAY WARNING	ALPHA-NUMERIC	1	8
319	C	AF	CLEAR THEN ADJ	ALPHA-NUMERIC	1	8
320	C	AF	ROUTING NUMBER	ALPHA-NUMERIC	9	8
321	E	EA	HISTORY HOURS	NUMERIC	7	H

322	E	EA	HISTORY RATE	NUMERIC	7	H
323	E	EA	HISTORY SALARY	NUMERIC	8	H
324	G		CONTROLS 3-4-5	NUMERIC	12	8
325	F		NAME - FIRST 15	ALPHA-NUMERIC	15	F
326	B		HED COMPARES	ALPHA-NUMERIC	1	8
327			IDENTIFIER	ALPHA-NUMERIC	7	8
328			FIELD CONTENTS	ALPHA-NUMERIC	30	8
329	E	EB	TRANSFER CODE	ALPHA-NUMERIC	1	8
330	E	EB	TRANSFER 001	ALPHA-NUMERIC	2	8
331	E	EB	TRANSFER 002	ALPHA-NUMERIC	4	8
Field Number Table						
FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
332	E	EB	TRANSFER 003	ALPHA-NUMERIC	10	8
333	J	J	FICA UNIT NBR	NUMERIC	4	8
334	J	J	FED UNIT NBR	NUMERIC	4	8
335	J	J	STATE UNIT NBR	ALPHA-NUMERIC	4	9
336	B	A8	HED CLASS	NUMERIC	1	8
337	B	A8	HED ACCRUAL OPT	ALPHA-NUMERIC	1	8
338	B	A8	HED ARR/DIFF	ALPHA-NUMERIC	1	8
339			RESERVED FOR A8		7	
340			NOT YET IN USE			
341			NOT YET IN USE			
342	B	A8	HED MISC FIELD	ALPHA-NUMERIC	15	8
343	D	D	EXTRA COPY	ALPHA-NUMERIC	1	8
344			NOT YET IN USE			
345			NOT YET IN USE			
346			NOT YET IN USE			
347	W	WA	W CHANGE	ALPHA-NUMERIC	1	8
348	W	WA	W TRANS CODE	ALPHA-NUMERIC	2	8
349	W	WX	RETRO TABLE	ALPHA-NUMERIC	2	9
350	W	WX	RETRO KEY	ALPHA-NUMERIC	10	9
351	W	WX	RETRO SELECT	ALPHA-NUMERIC	2	8
352	W	WX	RETRO RATE	NUMERIC	7	8
353	W	WX	RETRO SALARY	NUMERIC	8	8
354	W	WX	RETRO START	DATE YYMMDD	6	8
355	W	WX	RETRO END DATE	DATE YYMMDD	6	8
356	W	WX	RETRO HED NBR	NUMERIC	3	8
357	W	WX	RETRO CODE	NUMERIC	1	8
358	C	AK	AK IDENTIFIER	ALPHA-NUMERIC	6	K
359	C	AK	AK FIELD 1	ALPHA-NUMERIC	12	K
360	C	AK	AK FIELD 2	ALPHA-NUMERIC	12	K
361	C	AK	AK FIELD 3	ALPHA-NUMERIC	12	K
362	C	AK	AK FIELD 4	ALPHA-NUMERIC	12	K
363	C	AK	AK FIELD 5	ALPHA-NUMERIC	12	K
364	C	AK	AK FIELD 6	ALPHA-NUMERIC	12	K
365			VALIDATE ONLY	ALPHA-NUMERIC	1	8
366			MODIFY CONTROLS	ALPHA-NUMERIC	1	8
367			LOADED FILE	ALPHA-NUMERIC	1	8
368			XTRA TRANSACTNS	ALPHA-NUMERIC	1	8
369			VERSION NUMBER	NUMERIC	2	8
370			RECON RUN	ALPHA-NUMERIC	1	8
371	H		H CHANGE	ALPHA-NUMERIC	1	
372	J		J CHANGE	ALPHA-NUMERIC	1	
373			NEXT TRAN CODE	ALPHA-NUMERIC	2	8
374	E	EA	TO DATE-BATCH	ALPHA-NUMERIC	2	8
375	R	R7	NOT CODE	ALPHA-NUMERIC	1	8
376	R	R7	COMPARE CODE	ALPHA-NUMERIC	1	8

377	R	R7	AND CODE	ALPHA-NUMERIC	3	8
378			CONT 1-2 CHANGE	ALPHA-NUMERIC	1	8
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
379	F	F3	CITY PROVINCE	ALPHA-NUMERIC	24	8
380	F	F3	POSTAL CODE	ALPHA-NUMERIC	6	8
381	R	R7	LITERAL LENGTH	NUMERIC	2	8
382	P	PE	PERIOD TABLE	DATE YYMMDD	6	8
383	P	PE	PERIOD AMOUNT	COMPUTATIONAL	9	8
384	P	PE	PERIOD HOURS	COMPUTATIONAL	9	8
385	P	PE	PERIOD WEEKS	COMPUTATIONAL	9	8
386	B	A8	ABSENCE TYPE	ALPHA-NUMERIC	2	8
387	F	F4	CITY	ALPHA-NUMERIC	20	8
388	F	F4	POSTAL CODE	ALPHA-NUMERIC	10	8
389	C	AG	PAY MSG 1 01-30	ALPHA-NUMERIC	30	8
390	C	AG	PAY MSG 1 31-60	ALPHA-NUMERIC	30	8
391	C	AG	PAY MSG 1 61-67	ALPHA-NUMERIC	7	8
392	C	AH	PAY MSG 2 01-30	ALPHA-NUMERIC	30	8
393	C	AH	PAY MSG 2 31-60	ALPHA-NUMERIC	30	8
394	C	AH	PAY MSG 2 61-67	ALPHA-NUMERIC	7	8
395	E	EA	CHANGE DATE	DATE YYMMDD	6	9
396	E		KEY FIELDS	ALPHA-NUMERIC	4	8
397	E		KEY FIELD 1	ALPHA-NUMERIC	1	8
398	E		KEY FIELD 2	ALPHA-NUMERIC	1	8
399	E		KEY FIELD 3	ALPHA-NUMERIC	1	8
400	E		KEY FIELD 4	ALPHA-NUMERIC	1	8
401			NOT YET IN USE			
402			NOT YET IN USE			
403			NOT YET IN USE			
404			NOT YET IN USE			
405			NOT YET IN USE			
406			NOT YET IN USE			
407			NOT YET IN USE			
408			NOT YET IN USE			
409			NOT YET IN USE			
410			NOT YET IN USE			
411			NOT YET IN USE			
412			NOT YET IN USE			
413			NOT YET IN USE			
414			NOT YET IN USE			
415			NOT YET IN USE			
416			NOT YET IN USE			
417			NOT YET IN USE			
418			NOT YET IN USE			
419			NOT YET IN USE			
420			NOT YET IN USE			
421			NOT YET IN USE			
422			NOT YET IN USE			
423			NOT YET IN USE			
424			NOT YET IN USE			
425			NOT YET IN USE			
426			NOT YET IN USE			
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
427			NOT YET IN USE			

428			NOT YET IN USE			
429			NOT YET IN USE			
430			NOT YET IN USE			
431			NOT YET IN USE			
432			NOT YET IN USE			
433			NOT YET IN USE			
434			NOT YET IN USE			
435			NOT YET IN USE			
436			NOT YET IN USE			
437			NOT YET IN USE			
438			NOT YET IN USE			
439			NOT YET IN USE			
440			NOT YET IN USE			
441			NOT YET IN USE			
442			NOT YET IN USE			
443			NOT YET IN USE			
444			NOT YET IN USE			
445			NOT YET IN USE			
446			NOT YET IN USE			
447			NOT YET IN USE			
448			NOT YET IN USE			
449			NOT YET IN USE			
450			NOT YET IN USE			
451			NOT YET IN USE			
452			NOT YET IN USE			
453			NOT YET IN USE			
454			NOT YET IN USE			
455			NOT YET IN USE			
456			NOT YET IN USE			
457			NOT YET IN USE			
458			NOT YET IN USE			
459			NOT YET IN USE			
460			NOT YET IN USE			
461			NOT YET IN USE			
462			NOT YET IN USE			
463			NOT YET IN USE			
464			NOT YET IN USE			
465			NOT YET IN USE			
466			NOT YET IN USE			
467			NOT YET IN USE			
468			NOT YET IN USE			
469			NOT YET IN USE			
470			NOT YET IN USE			
471			NOT YET IN USE			
472			NOT YET IN USE			
473			NOT YET IN USE			
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
474			NOT YET IN USE			
475			NOT YET IN USE			
476			NOT YET IN USE			
477			NOT YET IN USE			
478			NOT YET IN USE			
479			NOT YET IN USE			
480			NOT YET IN USE			
481			NOT YET IN USE			
482			NOT YET IN USE			

483			NOT YET IN USE				
484			NOT YET IN USE				
485			NOT YET IN USE				
486			NOT YET IN USE				
487			NOT YET IN USE				
488			NOT YET IN USE				
489			NOT YET IN USE				
490			NOT YET IN USE				
491			NOT YET IN USE				
492			NOT YET IN USE				
493			NOT YET IN USE				
494			NOT YET IN USE				
495			NOT YET IN USE				
496			NOT YET IN USE				
497			NOT YET IN USE				
498			NOT YET IN USE				
499			NOT YET IN USE				
500			NOT YET IN USE				
501			NOT YET IN USE				
502			NOT YET IN USE				
503			NOT YET IN USE				
504			NOT YET IN USE				
505			NOT YET IN USE				
506			NOT YET IN USE				
507			NOT YET IN USE				
508			NOT YET IN USE				
509			NOT YET IN USE				
510			NOT YET IN USE				
511			NOT YET IN USE				
512			NOT YET IN USE				
513			NOT YET IN USE				
514			NOT YET IN USE				
515			NOT YET IN USE				
516			NOT YET IN USE				
517			NOT YET IN USE				
518			NOT YET IN USE				
519			NOT YET IN USE				
520			NOT YET IN USE				
521			NOT YET IN USE				
Field Number Table							
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT	
522			NOT YET IN USE				
523			NOT YET IN USE				
524			NOT YET IN USE				
525			NOT YET IN USE				
526			NOT YET IN USE				
527			NOT YET IN USE				
528			NOT YET IN USE				
529			NOT YET IN USE				
530			NOT YET IN USE				
531			NOT YET IN USE				
532			NOT YET IN USE				
533			NOT YET IN USE				
534			NOT YET IN USE				
535			NOT YET IN USE				
536			NOT YET IN USE				
537			NOT YET IN USE				

538			NOT YET IN USE			
539			NOT YET IN USE			
540			NOT YET IN USE			
541			NOT YET IN USE			
542			NOT YET IN USE			
543			NOT YET IN USE			
544			NOT YET IN USE			
545			NOT YET IN USE			
546			NOT YET IN USE			
547			NOT YET IN USE			
548			NOT YET IN USE			
549			NOT YET IN USE			
550			NOT YET IN USE			
551			NOT YET IN USE			
552			NOT YET IN USE			
553			NOT YET IN USE			
554			NOT YET IN USE			
555			NOT YET IN USE			
556			NOT YET IN USE			
557			NOT YET IN USE			
558			NOT YET IN USE			
559			NOT YET IN USE			
560			NOT YET IN USE			
561			NOT YET IN USE			
562			NOT YET IN USE			
563			NOT YET IN USE			
564			NOT YET IN USE			
565			NOT YET IN USE			
566			NOT YET IN USE			
567			NOT YET IN USE			
568			NOT YET IN USE			
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
569			NOT YET IN USE			
570			NOT YET IN USE			
571			NOT YET IN USE			
572			NOT YET IN USE			
573			NOT YET IN USE			
574			NOT YET IN USE			
575			NOT YET IN USE			
576			NOT YET IN USE			
577			NOT YET IN USE			
578			NOT YET IN USE			
579			NOT YET IN USE			
580			NOT YET IN USE			
581			NOT YET IN USE			
582			NOT YET IN USE			
583			NOT YET IN USE			
584			NOT YET IN USE			
585			NOT YET IN USE			
586			NOT YET IN USE			
587			NOT YET IN USE			
588			NOT YET IN USE			
589			NOT YET IN USE			
590			NOT YET IN USE			
591			NOT YET IN USE			
592			NOT YET IN USE			

593			NOT YET IN USE				
594			NOT YET IN USE				
595			NOT YET IN USE				
596			NOT YET IN USE				
597			NOT YET IN USE				
598			NOT YET IN USE				
599			NOT YET IN USE				
600			NOT YET IN USE				
601	L	PA	MARITAL STATUS	NUMERIC	1	8	
602	L	PA	WORK AREA CODE	NUMERIC	3	8	
603	L	PA	WORK TELEPHONE	NUMERIC	7	8	
604	L	PA	WORK EXTENSION	NUMERIC	4	8	
605	L	PA	HOME AREA CODE	NUMERIC	3	8	
606	L	PA	HOME TELEPHONE	NUMERIC	7	8	
607	L	PA	CITIZENSHIP	ALPHA-NUMERIC	1	8	
608	L	PA	DRAFT CODE	ALPHA-NUMERIC	2	8	
609	L	PA	MILITARY CODE	ALPHA-NUMERIC	2	8	
610	L	PA	VETERAN CODE	ALPHA-NUMERIC	2	8	
611	L	PA	BLOOD TYPE	ALPHA-NUMERIC	3	8	
612	L	PA	BLOOD DONOR	ALPHA-NUMERIC	1	8	
613	L	PA	DONOR DATE	DATE YYMMDD	6	8	
614	L	PA	HANDICAP CODE	ALPHA-NUMERIC	2	8	
615	L	PA	RESTRICT WORK	ALPHA-NUMERIC	2	8	
616	L	PA	DISABILITY CODE	ALPHA-NUMERIC	2	8	
Field Number Table							
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT	
617	L	PA	DISABILITY DATE	DATE YYMMDD	6	8	
618	L	PA	ADJ SERVICE DTE	DATE YYMMDD	6	8	
619	L	PA	REVW/RAISE DATE	DATE YYMMDD	6	8	
620			NOT YET IN USE				
621			NOT YET IN USE				
622			NOT YET IN USE				
623			NOT YET IN USE				
624			NOT YET IN USE				
625			NOT YET IN USE				
626	L	PB	PENSION CODE	ALPHA-NUMERIC	1	8	
627	L	PB	RETIREMENT DATE	DATE YYMMDD	6	8	
628	L	PB	COMP PAID INS	NUMERIC	7	8	
629	L	PB	INSURANCE CODE1	ALPHA-NUMERIC	4	8	
630	L	PB	INSURANCE AMT 1	NUMERIC	7	8	
631	L	PB	INSURANCE CODE2	ALPHA-NUMERIC	4	8	
632	L	PB	INSURANCE AMT 2	NUMERIC	7	8	
633	L	PB	INSURANCE CODE3	ALPHA-NUMERIC	4	8	
634	L	PB	INSURANCE AMT 3	NUMERIC	7	8	
635	L	PB	INSURANCE CODE4	ALPHA-NUMERIC	4	8	
636	L	PB	INSURANCE AMT 4	NUMERIC	7	8	
637	L	PB	NOT IN USE	ALPHA-NUMERIC	7	8	
638			NOT YET IN USE				
639			NOT YET IN USE				
640	L	PC	MED COV CODE	ALPHA-NUMERIC	4	8	
641	L	PC	MED ELIG DATE	DATE YYMMDD	6	8	
642	L	PC	SHORT TERM CODE	ALPHA-NUMERIC	4	8	
643	L	PC	SHRT TM ELIG DT	DATE YYMMDD	6	8	
644	L	PC	LONG TERM CODE	ALPHA-NUMERIC	4	8	
645	L	PC	LONG TM ELIG DT	DATE YYMMDD	6	8	
646	L	PC	DENTAL PLAN CD	ALPHA-NUMERIC	4	8	
647	L	PC	DENTAL ELIG DT	DATE YYMMDD	6	8	

648	L	PC	NOT IN USE	ALPHA-NUMERIC	25	8
649			NOT YET IN USE			
650			NOT YET IN USE			
651	L	PH	SALARY DATE	DATE YMMDD	6	K
652	L	PH	REASON CODE	NUMERIC	2	K
653	L	PH	REASON DESCRIP	ALPHA-NUMERIC	10	K
654	L	PH	PRIOR RATE	NUMERIC	7	K
655	L	PH	PRIOR FREQ	NUMERIC	1	K
656	L	PH	PRIOR PAY CODE	NUMERIC	1	K
657	L	PH	NORMAL HOURS	NUMERIC	5	K
658	L	PH	PRIOR SALARY	NUMERIC	8	K
659	L	PH	ANNUAL SALARY	NUMERIC	7	K
660	L	PH	ANNUAL INCREASE	NUMERIC	6	K
661	L	PH	PERCENT CHANGE	NUMERIC	4	K
662	L	PH	MONTHS SINCE	NUMERIC	2	K
663	L	PH	SAL PERFORMANCE	NUMERIC	3	K
Field Number Table						
FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
664	L	PH	JOB CATEGORY	ALPHA-NUMERIC	4	K
665			NOT YET IN USE			
666			NOT YET IN USE			
667			NOT YET IN USE			
668	L	PS	SKILL CODE	ALPHA-NUMERIC	6	K
669	L	PS	SKILL DESC	ALPHA-NUMERIC	30	K
670	L	PS	ORGANIZATION CD	NUMERIC	1	K
671	L	PS	YRS EXPERIENCED	NUMERIC	2	K
672	L	PS	LAST YR SKILL	NUMERIC	2	K
673	L	PS	PROFICIENCY RTE	NUMERIC	3	K
674	L	PS	LAST ACTIVITY	DATE YMMDD	6	K
675			NOT YET IN USE			
676			NOT YET IN USE			
677			NOT YET IN USE			
678			NOT YET IN USE			
679			NOT YET IN USE			
680			NOT YET IN USE			
681	L	PJ	JOB DATE	DATE YMMDD	6	K
682	L	PJ	JOB REASON	NUMERIC	2	K
683	L	PJ	JOB HISTORY CD	ALPHA-NUMERIC	5	K
684	L	PJ	JOB DESCRIPTION	ALPHA-NUMERIC	15	K
685	L	PJ	SALARY GRADE	ALPHA-NUMERIC	6	K
686	L	PJ	JOB ORG CODE	NUMERIC	1	K
687	L	PJ	LOCATION CODE	ALPHA-NUMERIC	4	K
688	L	PJ	LOCATION DESC	ALPHA-NUMERIC	15	K
689	L	PJ	JOB PERFORMANCE	NUMERIC	3	K
690	L	PJ	JOB STATUS CODE	ALPHA-NUMERIC	2	K
691			NOT YET IN USE			
692			NOT YET IN USE			
693			NOT YET IN USE			
694			NOT YET IN USE			
695			NOT YET IN USE			
696	L	PR	INJURY DATE	DATE YMMDD	6	K
697	L	PR	INJURY CODE	ALPHA-NUMERIC	4	K
698	L	PR	INJURY DESC	ALPHA-NUMERIC	25	K
699	L	PR	DISAB CLAIM CD	NUMERIC	1	K
700	L	PR	WKRS COMP CODE	NUMERIC	1	K
701	L	PR	LITIGATION CODE	NUMERIC	1	K
702	L	PR	OSHA CODE	NUMERIC	1	K

703	L	PR	CLAIM NUMBER	ALPHA-NUMERIC	12	K
704	L	PR	DAYS LOST	NUMERIC	3	K
705	L	PR	DAYS RESTRICTED	NUMERIC	3	K
706	L	PR	OPEN CODE	ALPHA-NUMERIC	1	K
707	L	PR	HOME DEPT	ALPHA-NUMERIC	4	K
708			NOT YET IN USE			
709	L	PT	EDUCATION TYPE	ALPHA-NUMERIC	6	K
710	L	PT	EDUCATION NAME	ALPHA-NUMERIC	15	K
711	L	PT	INSTITUTION CD	ALPHA-NUMERIC	4	K
Field Number Table						
FIELD NUMBER	SEGMENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
712	L	PT	INSTITUTION DSC	ALPHA-NUMERIC	20	K
713	L	PT	ED NBR YEARS	NUMERIC	2	K
714	L	PT	ED LAST YEAR	NUMERIC	2	K
715	L	PT	DEGREE CODE	ALPHA-NUMERIC	3	K
716	L	PT	COMPLETION CODE	ALPHA-NUMERIC	1	K
717	L	PT	ED REFUND	ALPHA-NUMERIC	3	K
718	L	PT	ED PERCENT CO	NUMERIC	3	K
719			NOT YET IN USE			
720			NOT YET IN USE			
721			NOT YET IN USE			
722	C	DG	SALARY GRADE	ALPHA-NUMERIC	6	K
723	C	DG	MINIMUM AMOUNT	NUMERIC	6	K
724	C	DG	MAXIMUM AMOUNT	NUMERIC	6	K
725	C	DG	MIDPOINT AMOUNT	NUMERIC	6	K
726			NOT YET IN USE			
727			NOT YET IN USE			
728			NOT YET IN USE			
729			NOT YET IN USE			
730	L	PD	PLAN IDENTIFIER	ALPHA-NUMERIC	6	K
731	L	PD	PLAN ENTRY DATE	DATE YYMMDD	6	K
732	L	PD	PERCENT VESTED	NUMERIC	3	K
733	L	PD	CURR VEST DATE	NUMERIC	4	K
734	L	PD	FULLY VEST DATE	NUMERIC	4	K
735	L	PD	PRIOR YEAR BAL	NUMERIC	9	K
736	L	PD	CURR EMPL CONT	NUMERIC	7	K
737	L	PD	CURR EMPLR CONT	NUMERIC	7	K
738	L	PD	NORMAL BENEFIT	NUMERIC	5	K
739	L	PD	EARLY BENEFIT	NUMERIC	5	K
740	L	PD	DEATH BENEFIT	NUMERIC	5	K
741	L	PD	SURV SPOUSE ID	ALPHA-NUMERIC	1	K
742	L	PD	SURV SPOUSE DTE	NUMERIC	4	K
743			NOT YET IN USE			
744			NOT YET IN USE			
745			NOT YET IN USE			
746	L	QV	HIGH PAID KEY	NUMERIC	2	K
747	L	QV	FILLER	ALPHA-NUMERIC	4	K
748	L	QV	PRI YR EARNINGS	NUMERIC	10	K
749	L	QV	PRI YR DEF EARN	NUMERIC	10	K
750	L	QV	PRI YR 5% OWNER	ALPHA-NUMERIC	1	K
751	L	QV	PRI YR OFFICER	ALPHA-NUMERIC	1	K
752	L	QV	PRI YR TOP 20%	ALPHA-NUMERIC	1	K
753	L	QV	CUR YR EARNINGS	NUMERIC	10	K
754	L	QV	CUR YR DEF EARN	NUMERIC	10	K
755	L	QV	CUR YR 5% OWNER	ALPHA-NUMERIC	1	K
756	L	QV	CUR YR OFFICER	ALPHA-NUMERIC	1	K
757	L	QV	CUR YR TOP 100	ALPHA-NUMERIC	1	K

758	L	QV	CUR YR TOP 20%	ALPHA-NUMERIC	1	K
Field Number	Table					
FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
759	L	QV	OWNER FAMILY	ALPHA-NUMERIC	1	K
760	L	QV	OWN EMP NUMBER	ALPHA-NUMERIC	10	K
761	L	QV	EXCLUD 20% TEST	ALPHA-NUMERIC	1	K
762	L	QV	HIGHLY PAID	ALPHA-NUMERIC	1	K
763	L	QW	HIGH PAID YEAR	NUMERIC	2	K
764	L	QW	FILLER	ALPHA-NUMERIC	4	K
765	L	QW	OWNER C 1-2	ALPHA-NUMERIC	6	K
766			NOT YET IN USE			
767			NOT YET IN USE			
768	W	WS	DATA TYPE-TABLE	ALPHA-NUMERIC	2	9
769	W	WS	STMNT SEQ. NBR	NUMERIC	6	9
770	W	WS	BENEFIT CODE	ALPHA-NUMERIC	6	9
771	W	WS	CARR. CONTROL	ALPHA-NUMERIC	1	8
772	W	WS	TEXT FIELD 1	ALPHA-NUMERIC	30	8
773	W	WS	TEXT FIELD 2	ALPHA-NUMERIC	30	8
774	W	WS	TEXT FIELD 3	ALPHA-NUMERIC	2	8
775	W	WP	DATA TYPE-TABLE	ALPHA-NUMERIC	2	9
776	W	WP	WP KEY	ALPHA-NUMERIC	10	9
777	W	WP	POSITION DESC	ALPHA-NUMERIC	15	8
778	W	WP	POSITION TITLE	ALPHA-NUMERIC	15	8
779	W	WP	E.E.O. CODE	NUMERIC	2	8
780	W	WP	WRKRS COMP CODE	ALPHA-NUMERIC	5	8
781	W	WP	SALARY GRADE	ALPHA-NUMERIC	6	8
782	W	WP	POSITIONS BDGTD	NUMERIC	5	8
783	W	WP	SALARY BUDGETED	NUMERIC	9	8
784	W	WP	RESERVED FOR WP	ALPHA-NUMERIC	7	8
785	W	WQ	DATA TYPE-TABLE	ALPHA-NUMERIC	2	9
786	W	WQ	WQ KEY	ALPHA-NUMERIC	10	9
787	W	WQ	SALARY INCR.	NUMERIC	9	8
788	W	WQ	QUAL. FIELD 1	ALPHA-NUMERIC	30	8
789	W	WQ	QUAL. FIELD 2	ALPHA-NUMERIC	24	8
790	W	WQ	RESERVED FOR WQ	ALPHA-NUMERIC	1	8
791	W	WU	DATA TYPE-TABLE	ALPHA-NUMERIC	2	9
792	W	WU	WU KEY	ALPHA-NUMERIC	10	9
793	W	WU	WU USER FLD 1	ALPHA-NUMERIC	8	8
794	W	WU	WU USER FLD 2	ALPHA-NUMERIC	8	8
795	W	WU	WU USER FLD 3	ALPHA-NUMERIC	8	8
796	W	WU	WU USER FLD 4	ALPHA-NUMERIC	8	8
797	W	WU	WU USER FLD 5	ALPHA-NUMERIC	8	8
798	W	WU	WU USER FLD 6	ALPHA-NUMERIC	8	8
799	W	WU	WU USER FLD 7	ALPHA-NUMERIC	8	8
800	W	WU	WU USER FLD 8	ALPHA-NUMERIC	8	8
801	W	WA	DATA TYPE-TABLE	ALPHA-NUMERIC	2	9
802	W	WA	KEY DEFINE	ALPHA-NUMERIC	19	9
803	W	WA	SUB-TABLE CODE	ALPHA-NUMERIC	1	9
804	W	WA	DESCRIPTIVE FLD	ALPHA-NUMERIC	20	8
805	W	WC	EDIT FIELD	NUMERIC	5	8
806	W	WL	WL TABLES	ALPHA-NUMERIC	2	9
Field Number	Table					
FIELD NUMBER	SEG-MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
807	W	WL	WL KEY	ALPHA-NUMERIC	23	9
808	W	WL	WL FIELD 1	ALPHA-NUMERIC	26	8

809	W	WL	WL FIELD 2	ALPHA-NUMERIC	26	8
810	E	E	USER E FIELD	ALPHA-NUMERIC	1	8
811	E	E	FAIR CODE	ALPHA-NUMERIC	1	8
812	E	E	PERIOD OVERRIDE	NUMERIC	2	8
813	J	J	CDN TAX CREDIT	NUMERIC	5	8
814			NOT YET IN USE			
815			NOT YET IN USE			
816			NOT YET IN USE			
817			NOT YET IN USE			
818			NOT YET IN USE			
819			NOT YET IN USE			
820	C	AL	PAY THROUGH ID	NUMERIC	1	K
821	C	AL	BLANK FIELD	ALPHA-NUMERIC	5	K
822	C	AL	PAY THRU FIELD1	ALPHA-NUMERIC	30	K
823	C	AL	PAY THRU FIELD2	ALPHA-NUMERIC	30	K
824			NOT YET IN USE			
825			NOT YET IN USE			
826	C	AF	PAYSLIP FLAG 01	ALPHA-NUMERIC	1	8
827	C	AF	PAYSLIP FLAG 02	ALPHA-NUMERIC	1	8
828	C	AF	PAYSLIP FLAG 03	ALPHA-NUMERIC	1	8
829	C	AF	PAYSLIP FLAG 04	ALPHA-NUMERIC	1	8
830	C	AF	PAYSLIP FLAG 05	ALPHA-NUMERIC	1	8
831	C	AF	PAYSLIP FLAG 06	ALPHA-NUMERIC	1	8
832	C	AF	PAYSLIP FLAG 07	ALPHA-NUMERIC	1	8
833	C	AF	PAYSLIP FLAG 08	NUMERIC	1	8
834	C	AF	PAYSLIP FLAG 09	ALPHA-NUMERIC	1	8
835	C	AF	PAYSLIP FLAG 10	ALPHA-NUMERIC	1	8
836	C	AF	PAYSLIP FLAG 11	ALPHA-NUMERIC	1	8
837	C	AF	FRICK TAPE FLAG	ALPHA-NUMERIC	1	8
838	C	AF	TAX MC OVERRIDE	ALPHA-NUMERIC	1	8
839	C	AF	TAX-C12	ALPHA-NUMERIC	6	8
840			NOT YET IN USE			
841	T	T3	TAX CLASS 1	NUMERIC	1	8
842	T	T3	TAX CLASS 2	NUMERIC	1	8
843	T	T3	TAX CLASS 3	NUMERIC	1	8
844	T	T3	TAX CLASS 4	NUMERIC	1	8
845	T	T3	TAX CLASS 5	NUMERIC	1	8
846	T	T3	TAX CLASS 6	NUMERIC	1	8
847	T	T3	TAX CLASS 7	NUMERIC	1	8
848	T	T3	TAX CLASS 8	NUMERIC	1	8
849	T	T3	TAX CLASS 9	NUMERIC	1	8
850			COBOL RETURN		3	
851			TEMPORARY HOLD		4	
852			CALL REPORT		4	
853			REV WORK		1	
Field Number Table						
FIELD NUMBER	SEG- MENT	TRANSACTION CODE	DESCRIPTION	TYPE OF FIELD	INPUT LENGTH	KEY MAINT
854			RANDOM IO		1	
855			RANDOM AREA		1	
856			IO RESULT		1	
857			KEY SIZE		2	
858			START KEY		17	
859			CALL CODES		14	
860			WORK AREA		1	
861			NOT YET IN USE			
862			NOT YET IN USE			
863			NOT YET IN USE			

864			NOT YET IN USE			
865			NOT YET IN USE			
866			NOT YET IN USE			
867			NOT YET IN USE			
868			ERROR MESSAGE		30	
869			STACK START			
870			STACK FIELD		1	
871	J	J	J USER CODE	ALPHA-NUMERIC	1	8
872			NOT YET IN USE			
873			NOT YET IN USE			
874	C	CX	QUERY PARAM 1	ALPHA-NUMERIC	10	8
875	C	CX	QUERY PARAM 2	ALPHA-NUMERIC	10	8
876	C	CX	QUERY PARAM 3	ALPHA-NUMERIC	10	8
877	C	CX	QUERY PARAM 4	ALPHA-NUMERIC	10	8
878	C	CX	QUERY PARAM 5	ALPHA-NUMERIC	10	8
879	C	CX	QUERY PARAM 6	ALPHA-NUMERIC	10	8
880	C	AJ	FREQ IDENT	ALPHA-NUMERIC	1	K
881	C	AJ	BLANK FIELD	ALPHA-NUMERIC	5	K
882	C	AJ	FREQ DESCRIBE	ALPHA-NUMERIC	12	K
883	C	AJ	ANNUAL FACTOR	NUMERIC	4	K
884	C	AJ	NBR WEEKS	NUMERIC	4	K
885	C	AJ	PERIODS PAID	NUMERIC	1	K
886	C	AJ	NEW PERIOD	ALPHA-NUMERIC	1	K
887	C	AJ	PAYMENT DATE	DATE YYMMDD	6	K
888	C	AJ	ANNIVRSRY DATE	DATE YYMMDD	6	K
889	C	AJ	PERIOD DATE	DATE YYMMDD	6	K
890	C	AJ	PERIOD NUMBER	NUMERIC	2	K
891	C	AJ	PAY CYCLE	NUMERIC	1	K
892	C	AJ	DED CYCLE	NUMERIC	1	K
893	C	AJ	SAVE PERIOD	DATE YYMMDD	6	K
894	C	AJ	SAVE NUMBER	NUMERIC	2	K
895	C	AJ	SAVE PAY	NUMERIC	1	K
896	C	AJ	SAVE DED	NUMERIC	1	K
897	C	AJ	ACTUAL HOURS	NUMERIC	5	K
898	C	AJ	LABOR PCT	NUMERIC	4	K
899			NOT YET IN USE			
900			NOT YET IN USE			

---

# Glossary of Terms

---

## **.EXE**

A binary file containing a program in machine language that is ready to be executed.

## **.INI**

A file that contains the parameters (values) used by the .exe file (program).

## **360-degree appraisal**

Appraisals that include evaluations from an employee's managers and supervisors, peers, subordinates, and even customers, clients, and suppliers.

## **820 (Premium Payment for Group Insurance Products) Transaction Set**

A specific transaction, standardized by the United States government, used between Covered Entities to relay remittance information and/or premium payments for health care insurance products.

## **834 (Benefit Enrollment and Maintenance) Transaction Set**

A specific transaction, standardized by the United States government, used between Covered Entities to relay information relating to the initial enrollment in and subsequent maintenance of individuals enrolled in a health care insurance product.

## **Absence data**

Employee-level absence information that is entered on the absences forms.

## **Absence point**

User-defined number that may be assigned for a particular absence and that can be totalled over time to

determine if an employee is within the accepted number of absences for a time period.

## **Absence type**

A classification of an employee absence, such as 'jury duty' or 'sick'. Employee absences are recorded by date and absence type.

## **Account timeout**

The period of time that elapses before a user's account becomes invalid because of inactivity.

## **Accumulator id**

A three-position, alphanumeric identifier for a benefits accumulator.

## **ACH**

US-specific acronym for Automated Clearing House. The ACH Network provides inter-bank clearing of electronic payments for participating depository financial institutions. The American Clearing House Association, Federal Reserve, Electronic Payments Network, and Visa act as ACH Operators—central clearing facilities through which financial institutions transmit or receive ACH entries.

## **Acrobat**

A suite of programs developed by Adobe Systems, Inc. For creating and distributing electronic documents. Programs in the suite allow you to create a portable document format (PDF) file for a document. You can then distribute the PDF file electronically to people who view the document with their freely distributed acrobat reader. People viewing a PDF file (or document) with the Acrobat Reader see the document with the exact layout intended by the author.

## **Action button**

An action button performs an action such as saving the information you entered or telling the system you finished reviewing a page. An action button consists of an icon (or button) accompanied by underlined text (link text). For example, at various places throughout eCyborg Interactive Workforce you may see an action button displaying a check mark accompanied by the underlined text 'save changes'. You can click either the text or the button to save your changes to the page.

### Activity types

Event that can happen to an employee during employment, for example, new hire, leave of absence, termination, and so forth.

### Activity code

Describes the clock transaction (ring) activity, such as clock start or meal end.

### Activity types

With the time and attendance solution, you can set up the system so that an employee or group of employees may clock in and out for up to eight different activities: clock-in (1), break 1 start (2), break 1 end (3), meal start (4), meal end (5), break 2 start (6), break 2 end (7), and clock end (8).

### Actuarial valuation

An examination of a pension plan to determine if contributions are being accumulated at a rate sufficient to pay the promised pensions.

### Adjustment

Adjustments increase or reduce to-date accumulations of earnings, deductions, and taxes. Adjustments update month-to-date, quarter-to-date, and year-to-date activity. You can also correct or eliminate deduction arrears, transfer amounts or hours from one earning or deduction to another, redistribute labor hours and dollars, and refund deductions. Adjustment transactions are processed the next time the organization is processed by the batch payroll programs during either a payroll or maintenance run.

### Administration home page

The administration page that displays when a user logs on using his or her administrator user ID and password. The administration page displays links to individual administrator pages (eCyborg Interactive Workforce, Human Resources Administration, Benefits Administration, and Payroll Administration).

### Administrative User ID

User ID created by an administrator with the role of eCyborg Interactive Workforce administrator. This ID differs from the employee user ID generated for the administrator.

### Agent types

Agent Indicator on the Employer Information Record (QE-SCR). The field's only allowable entries are selected from option list PR55, which consists of two agent types: Agent Indicator code and Common Pay master.

### Further definition of agent types:

1. An employer that wants to use an agent prepares Form 2678, Employer Appointment, and submits the form to an agent.

The agent submits to IRS the form received from the employer along with a written request for authority to act as an agent for an employer.

The IRS gives written approval.

2. A common paymaster is a corporation that pays an employee who works for two or more related corporations at the same time.

### Aggregate tax method

Method of calculating taxes in which year-to-date income is used to project annual wages (using prorating), on which taxes are calculated. With this method, the amount of tax withheld can vary from pay period to pay period. This method is useful in preventing a salesperson from being over withheld as the result of fluctuations in commission over various pay periods. It is activated in Payroll Administration by selecting aggregate/cumula tax (9) from the Withholding Method (PR09) option list on the Employee Tax Record Maintenance form. It is also referred to as cumulative tax calculation method.

### Anniversary date

Period-end date for the next regularly scheduled pay run for a specific pay frequency.

### Annualization

Process of calculating the annual amount of pay based on the number of pay periods and pay period amounts. Calculated by multiplying the number of pay periods in the year by the current taxable wages in the pay period.

### Annualization factor

The factor that is used to multiply current pay period wages to determine annual wages. For example, a monthly pay frequency has an annualization factor of 12. Payroll Administration typically calculates income

taxes on the basis of annual wages. The annualization factor is entered by selecting an option from the Annualization (PP33) option list on the Company Pay Frequencies form for each pay frequency.

**Annuitant**

Someone entitled to receive or currently receiving payments from an annuity.

**Annuity**

A contract providing an income for a specific period of time.

**Applicant**

A person who is applying for a job or position in your organization. Internal applicants come from within your organization while external applicants come from outside of your organization.

**Applicant organization**

Organization used to store and update applicant records.

**Appraisal rating**

A method of ranking the performance of an employee during a given period using options ranging from 1-outstanding to 5-unsatisfactory.

**Archive**

To transfer files to a slower, cheaper media (usually magnetic tape) to either free up the hard disk space they occupy or to create a backup copy of the files that can be used to restore programs and/or data if there is a system failure.

**Arrears**

Amount that could not be deducted from an employee's net pay because the available net amount was less than the deduction. This amount may be recorded for recovery in a future pay period.

**As-of reporting**

Ability to report on data for a specified date or date range.

**ASCII**

American Standard Code for Information Interchange. The basis of character sets used in almost all present-day computers; US-ASCII uses only seven bits to convey some control codes, space, numbers, most basic punctuation, and unaccented letters a-z and A-Z.

**Ask Me wizard**

A natural language, full-text search facility within the online help. This allows users to type in a question, the wizard interprets the question, and displays related topics.

**Audit record**

A snapshot of information entered on a form. Audit records are stored on the employee database and are displayed on audit reports in an is/was reporting format. Adjustments and time entries are stored as audit records and are extracted for a payroll run in which they update the employee's record.

**Audit report**

A report that is available after the running of a program; it lists created records as well as error messages for records that could not be created.

**Audit trail**

A report of changes made to your employee database, such as the Payroll Audit Trail (0101) report.

**Authorized absence**

Absences that are generally considered as paid time away from regularly scheduled work.

**Automatic plan**

A plan that has been defined with a default option and default pre- or posttax indicator (also known as core/default plan).

**Average deferral percentage**

Percentage used in nondiscrimination and compliance testing mandated by US law. The calculation is defined as the contribution divided by the compensation.

**Average rating**

A rating used for performance appraisal systems with categories weighted by relative importance, where the average score reflects the weighted scores.

**Back**

Takes the user back to the previous page.

**Badge**

Time and Attendance Administration can be set up to use two different types of badge readers. The type of badge your organization uses, is determined by your third party badge reader software. The two types of badges are magnetic badges and bar code badges.

### **Badge error**

Occurs when a badge is used to create a clock transaction (ring) and an employee has not been assigned to the badge.

### **Badge number**

Up to ten-character ID stored on employee badges and clock transactions (rings) that tie clock transactions (rings) to an employee on the *Employee Database* (on page 174).

### **Banner**

Banner forms separate groupings on forms produced from the Federal, State/Local, and Employee Queues.

### **Batch**

A group of transactions submitted to the batch payroll processing system. Also, a collection of time entries that corresponds to an employee group, such as department.

### **Batch control record**

Precedes all transactions separated by group; used to identify the company to which the transactions in that group apply. By entering anticipated totals for dollars and hours on the batch control record, you may verify your totals against those accumulated by the system.

### **Batch layout facility**

A program that produces a segment layout for loading forms via batch. This was formerly known as BATCHL.

### **Batch Master file (P20)**

Sequential version of the Payroll Master information from the Employee Database. The Batch Master File (P20) is accessed and maintained during a batch maintenance run or pay run.

### **Batch number**

An alphanumeric field on the batch control record containing a user-defined value used to identify a unique group of time entries or transactions.

### **Batch processing**

A processing method that runs in the background and requires limited intervention.

### **Batch transaction**

Precedes all transactions separated by group; used to identify the company to which the transactions in that

group apply. By entering anticipated totals for dollars and hours on the BATCH transaction, you may verify your totals against those accumulated by the system.

### **Benchmark job**

A standard or point of reference for determining total job points.

### **Beneficiary**

A person named by the participant in an insurance or pension plan to receive any benefit provided by the plan if the participant dies.

### **Benefits Business Partner**

See *Trading Partner* (on page 197).

### **Benefits control number**

A four-position, alphanumeric identifier that specifies which tables are accessed for an organization.

### **Benefits statement**

Report that indicates the coverage and cost of each benefits plan in which an employee participates.

### **Big option list**

A large option list that includes a search facility. This was formerly known as a big codeset.

### **Blocking factor**

Represents the number of physical records in a logical record.

### **Bridge loan**

A loan made to assist a relocated employee in purchasing a new residence before the sale of their old residence is complete.

### **Browser**

Software application used to locate and display web pages. Modern browsers give users access to graphics, text, and multimedia information, including sound and video.

### **Budget plan year**

A twelve-month period over which a salary budget is effective.

### **Budget scenario**

The result of creating one or more salary plans in order to see the effect of different increase policies on the budget.

**Budget setting**

The process of analyzing and selecting an organization's salary budget for the coming plan year.

**Cafeteria plan**

A specific type of flexible benefit plan that allows employees to select their benefits from a number of benefit plans. This term may be used interchangeably with flexible benefits plan.

**Calculation option list**

An option list that contains calculation formula. This was formerly known as a calculation codeset.

**Candidate**

A person who is applying for a job or position in your organization and is under consideration.

**Career planning**

Providing career incentives such as advancement and additional education and training for individual employees in order to meet projected organizational needs.

**Carrier record**

A carrier record supplies information from one application area to another application.

**Case-sensitive**

A program that distinguishes between uppercase (capital) and lowercase (small) letters. A case-sensitive program that expects you to enter all commands in uppercase will not respond correctly if you enter one or more characters in lowercase.

**Catalog**

A file (with the extension of .cat) that contains all the information necessary for Impromptu to access and retrieve information from a relational database. The catalog provides a business view of the data, as well as information about what database to access, where the database is stored, and how the tables in the catalog are joined in the data mart.

**Category code**

General term used to refer to the option selected from category (PP01 and PP02) option lists on the company earnings and company deductions forms. It is used to indicate the type of earning or deduction.

**CBSVB**

COBOL program used to produce reports and to access or maintain The Solution Series offline.

**CBT**

Computer-based training. Training through use of computer.

**CE/H**

Abbreviation for considered earnings/hours.

**Change control facility**

A facility for updating and comparing your system control repository. This was formerly known as MAINTI/MAINTO.

**Check box**

A standard windows control that displays a yes/no setting, either checked (yes) or unchecked (no).

**Check digit**

Unique identifier that is generated by the TBLCHK program and used by the system to check the table relationship records.

**Checklist**

A list of tasks to be performed in sequence. The checklist displays within the navigator area. Checklists link tasks and other checklists together to perform workflow functions. Users can display a checklist by selecting a checklist icon within the tasks in the navigator.

eCyborg Interactive Workforce specific—a list of tasks/pages generally displayed in a chart with hot spots (links) for the checklist items. The user clicks the link to access the page.

**Checklist item**

An item appearing within the navigator when a checklist is being displayed. Checklist items include tasks, dialogs and even other checklists.

**Checklist item status**

Defines the status of a checklist item. These can be:

- Available to perform
- Required
- Not available
- Already completed

### **Checklist margin**

The area of the navigator that displays the checklist item status when a checklist is being displayed.

### **Checklist wizard**

Tool used to create checklists.

### **Checkmark**

If in the done column of an eCyborg Interactive Workforce checklist, indicates that an item on a checklist is complete. Can also indicate OK, finished, submit, and so forth.

### **Class**

A class is an occurrence of a course that is specific to a location and a date that is being administered using Training Administration. For example, 'eCyborg: Using the Web Client' on Thursday, December 21, in Chicago is a class of the course 'eCyborg: Using the Web Client'.

### **Class evaluation results**

These are the results as entered on the evaluation forms filled out by the class participants upon completion of the class. These results are recorded on the class evaluation results form.

### **Cleared payment**

A check that has been cashed or deposited by the payee and has been returned, marked as such, to the payer's bank, credit union, or financial institution.

### **Client data file**

File containing information replicated from the System Control Repository. Used by client workstations to improve response time, since editing can be performed locally. May be located on each client workstation or may be located on a server and be shared by multiple client workstations on the network. Formerly known as the Client Control File.

### **Clock in and out**

Also referred to as swipe/swiping the clock. When an employee uses their badge to record an activity time, they must pass their badge through the badge reader. This action can be referred to as clocking in and out.

### **Clock transaction (ring)**

Record containing the information needed to create time entries for payroll processing. Clock transaction

(ring) information includes date, time, and badge number. A clock transaction (ring) is created when a badge is swiped through a clock.

### **Clock transaction warning**

Occurs when a clock transaction (ring) time falls outside of an employee's schedule warning times.

### **Closing costs**

The costs associated with the purchase of a new house.

### **CLP**

Abbreviation for certificates, licenses, and permits.

### **Co-ordinator**

A coordinator is an instructional institution, organization or person who administers training courses.

### **Codeset**

A list of valid code values and associated descriptions from which you may select an appropriate entry. This is now known as an option list.

### **Coefficient**

Customer-defined value used in the formula to calculate a new salary grade midpoint value.

### **Combined register (2222) report**

A report that provides a detailed printout of all earnings, hours, taxes, and deductions for all the payments and adjustments made on a payroll run. It is Report Generator 2222.

### **Command button**

A standard windows control that initiates a command or sets an option (previously known as push button).

### **Common pay master**

An organization that pays employees who work for different legal entities. If all the conditions related to a common paymaster are met, the FICA and, in most cases, the SUI does not have to start over.

### **Common tax organization**

A method of setting up taxation in an organization in which all necessary tax specification records are contained in a single organization. The common tax organization often handles tax specification records more efficiently, since it avoids duplication of the

federal tax records and of any state or local records used by multiple companies.

**Communication event**

A letter or email that can be triggered automatically or manually within the system. Communication events are set up by the system administrator and usually include data from a form or record.

**Compa ratio**

The ratio of a given salary compared with the midpoint of the salary range. The formula is the salary divided by the midpoint.

**Company logical Master record**

Complete record for a company. It may be composed of multiple physical records.

**Company validation table**

Table that validates that an Organization Control Number is valid and payments can be made.

**Competency**

A requisite capacity to perform a single or set of skills or activities.

**Complement limit**

A 'complement limit' is the maximum number of complement units that can be assigned to a position at any one time.

**Complement position**

A 'complement position' is a position that is included in complement control.

**Complement unit**

A 'complement unit' is the type of unit used to measure the value of a position, for example, headcount, fte, or hours.

**Compliance**

Conformity in fulfilling legal requirements.

**Component**

The first level of functional organization on the navigator or menu, such as employee resourcing or employee development.

**Component icon**

An icon that denotes the current component. There are a number of components within the system. Each component appears as an icon on the navigator.

**Component plan**

Any plan included under the flex master plan or grouped together under a group master.

**Condition**

Predefined criteria that can be added to a report's filter.

**Considered earnings**

An employee's paid earnings that are to be accumulated, based on plan rules, for use in determining credited service or calculations of final benefits amounts.

**Considered earnings/hours (CE/H) accumulators**

Used only in benefits plans to accumulate the earnings and hours an employee has acquired toward eligibility for a deferred plan. Accumulators may be retained on a monthly, quarterly, or annual basis.

**Considered hours paid**

Actual number of hours for which an employee was paid and that are to be accumulated based on plan rules.

**Considered hours worked**

Actual number of hours an employee worked. These hours are to be accumulated based on plan rules for use in determining credited service for a plan participant (or for a non-participant if eligibility has been met).

**Consolidated reporting**

Option that enables packaged reports to be processed for all organizations (consolidated).

Customer-defined value used in the formula to calculate a new salary grade midpoint value.

**Context-sensitive help**

Information about an object and its current condition. It answers the question 'what is this?'

**Contractual exchange rate**

An annually reviewed exchange rate usually based on an employees anniversary date and reviewed once a year; often used for salaries.

### **Contribution type**

The type of contribution being made to a benefits plan. The system allows for the deduction and accumulation of up to five different contributions per plan: basic employee pretax, basic employee posttax, supplemental employee pretax, supplemental post-tax, and organization.

### **Control 1-2**

Organization Control Number. A six-character, customer-defined code that represents a company or group of employees within an organization.

### **Control levels**

A hierarchy of values used to determine the breakdown of an organization for reporting purposes. The values are user-defined.

### **Control number**

An alphanumeric designation assigned to a table to define the table records that will be used for each organization.

### **Conversion**

A method for transferring data from either a manual or automated system into the system.

### **Conversion option**

The interval at which you are converting currency (for example: yearly, monthly, weekly).

### **Core plan**

One of the plans that make up the minimum benefits in which all eligible employees are required to enroll—for example, medical and life. Employees who fail to return enrollment forms with their benefit choices may be automatically enrolled in the core plans (also known as default plans).

### **Cost categories**

Cost categories are classifications or divisions used to separate costs for training into broad groupings, for example, equipment or operating costs.

### **Cost types**

Cost types are used to further define training costs. For example, the category of equipment could be further broken down into the cost type of overhead projector and monitor rental.

### **Costing**

Projecting the future cost of a benefits plan contribution for budget purposes.

### **Costing record**

Labor record.

### **Course**

A course is a separate unit of instruction in a subject being administered using the training administration solution. For example, 'eCyborg: Using the Web Client' is a course. This may be applied to a training course provided internally or externally.

### **Course directory**

A course directory is a list of all available courses.

### **Covered Entity**

A member of one of three groups subject to the administrative simplification provisions of HIPAA. The three types of Covered Entities are health plans (not fully insured), clearinghouses, and providers that conduct transactions electronically.

### **Covered Entity Group**

A Covered Entity Group represents in the system a partnership between two Covered Entities (benefits business partners) who transmit benefit information electronically between each other in the form of the 834 (Benefit Enrollment and Maintenance) and 820 (Group Premium Payment for Insurance Products) Transaction Sets.

### **CPI**

Characters per inch.

### **Credited service**

The number of years of employment for which an employee is given credit for use in determining final benefits amounts.

### **Crew**

A group of employees who rotate from one schedule assignment (shift) to another, following a rotation pattern.

### **Crew code**

A unique, one-character, alphanumeric identifier of a crew.

**Cross-reference keys**

Provide direct query access to data within the system database.

**Crossfoot**

An accounting term that means to compare or add figures to make them balance. If you receive an error message such as 'Entries do not Crossfoot', this means that some amounts entered into fields are not balanced.

**CSL**

Abbreviation for *Cyborg Scripting Language* (on page 171).

**Cube (Power Play)**

File (with the extension .mdc) that contains data organized into dimensions to provide for faster data retrieval and drill down reporting capabilities.

**Cumulative data**

Also called 'to-date data'. includes payroll earning, deduction, net pay, taxable wage, and tax to-date figures for employees.

**Cursor**

A special symbol, usually a solid rectangle or a blinking underline character, that signifies where the next character will be displayed on the screen. To type in different areas of the screen, you need to move the cursor. You can use the arrow keys or a mouse to move the cursor.

**Custom report format**

Custom reporting used by states because their reporting format is neither MMREF, ICESA, nor TIB-4.

**Customer-defined**

Values that depend on an organization-specific definition--for example, option list.

**CYB88X**

An English Language root program used to set the production version switch to on or off, in addition to other automatic settings.

**CYBMST**

Library file that contains the main batch payroll software. This file consists of COBOL programs, report generators, and system tables.

**Cyborg Scripting Language**

Fourth-generation programming language, previously called English Language.

**Data extract**

Method for extracting information from The Solution Series for the purpose of subsequently loading it into eCyborg Interactive Workforce databases.

**Data Extract File Splitter (RSPLIT)**

COBOL program that creates separate files for every representative table, using the Data mart Extract File as input.

**Data load**

The process of moving data from one system or media to another. It encompasses data mapping, data extraction and conversion, and the actual loading of the data. Also the method of loading data extracted from The Solution Series into eCyborg Interactive Workforce databases using programming scripts.

**Data mapping**

The process of identifying, comparing, and matching data (field to field) to be converted from one system or media to another.

**Data mart**

Relational tables with a defined structure that have been designed to automatically accept full data mart extract data seamlessly.

**Data mart Extract File (FILE36)**

Temporary system file holding the copied data from The Solution Series.

**Data store**

Repository where data is stored. Client applications request information from a Solution Series data store.

**Database**

A collection of information organized so that a computer program can quickly search for and select specific pieces of data. Think of a database as an electronic filing system.

**De-enrollment**

The process of shutting off plan benefits for an employee for reasons other than a separation activity.

### **Deduct credits by plan**

A method of distributing flexible benefit credits. The total monetary value for credits is prorated based on the employee's pay frequency. Credits are given to employees as earnings added to their pay; the cost of individual employee plans are collected through payroll deductions and listed on the employee's payment stub.

### **Deduct credits by plan method**

A method of distributing flexible benefit credits. Credits are given to employees as earnings added to their pay; the individual employee plan costs are then collected through payroll deductions.

### **Deduction**

An amount subtracted from available net pay. Deductions can be involuntary (child support or maintenance) or voluntary (pension plans).

### **Deduction cycle**

A predetermined schedule for taking voluntary deductions, based on the defined frequency.

### **Default**

Value that will appear if no entry is typed or selected.

### **Deferred compensation**

Any benefit that is not immediately payable to an employee, but is instead deferred to a later date. This term refers to retirement vehicles, including all defined benefit, defined contribution, stock, and thrift/savings plan.

### **Deferred plan**

Any benefits plan in which benefits are not immediately payable to an employee, but are deferred to some later date. This term refers to retirement vehicles, including all defined benefit, defined contribution, stock, and thrift/savings plans.

### **DEL-PE**

Utility program used to delete existing Phonetic Keys.

### **Delimiter**

A character that tells the system where an item of data ends and another starts.

### **Density**

Number of bytes per inch that a tape drive writes to a tape.

### **Dependent**

An individual who relies or depends on another for his or her support.

### **Dependent GTL**

Acronym for Dependent group term life insurance plan.

### **Dependent number**

A unique number in the eCyborg Interactive Workforce database that identifies an employee's spouse and his or her other dependents.

### **Deposit advice**

Pay document that looks similar to a check and shows the amount deposited to the employee's account.

### **Derived data**

Reporting Administration enhances data during the extract process by capturing and storing table-derived data, such as salary grade, that do not reside on the employee's record.

### **Detail page**

A page in eCyborg Interactive Workforce that displays detailed information. Summary pages contain links to the detail for each record.

### **Dialog box**

A secondary window that appears on the screen to present information or request input. Dialog boxes are generally temporary—they disappear after you enter the requested information.

### **Direct deposit**

Also referred to as an Electronic Funds Transfer, or EFT, direct deposit allows employees to authorize the organization they work for to automatically deposit all or part of net pay into one or more bank accounts.

### **Disability**

Inability to pursue certain occupational activities due to physical or mental impairment.

### **Disability insurance tax**

A tax required by some us states to be funded by employee-paid contributions to pay all or part of the cost of disability insurance coverage. In Payroll Administration, US State disability insurance tax records are established as Type 4 taxes.

**Disciplinary action**

Action taken against an employee for violation of an organization policy or procedure.

**Discretionary increase**

A salary increase amount or percentage determined by a manager according to the guidelines established by the organization.

**Display**

Make data or images display on a computer monitor.

**Display box**

An area on a form in which data is displayed (formally known as an inquiry field).

**Display text boxes**

An area on a form in which data is shown but cannot be modified.

**Disposable income**

For garnishment purposes in the U.S., an employee's earnings minus deductions required by state or federal law.

**Distributed Access Log Table**

Form that records the date, time, and last record distributed to a unique location.

**Distributed location**

A customer location where data changes are replicated and may be distributed. A DL is identified to the system by a unique 5-position alphanumeric node ID.

**Distributed Location NODE Control Table**

Form that defines the operating environment of your remote node. It allows you to assign an access password for the node and indicate whether the option list and table records will be distributed to the remote node.

**Distributed Rules Table**

Form that records the FILE 02 segments that are to be excluded from the remote location.

**Distribution**

The process of passing data from a source DL to one or more target DLs.

**Distribution rules**

A set of parameters that determine how data will be distributed from one DL to another. These are defined at each DL by the owner using the distribution rules forms. Distribution rules are stored in tables that are not replicated (thus, they cannot be distributed).

**DL**

Abbreviation for *distributed location* (on page 173).

**Docked toolbar**

Toolbar that is located on the edge of a program window.

**Document type**

Categorization of a document, such as birth certificate, performance review, drivers license, and so on.

**Double-click**

Click a mouse button twice in rapid succession.

**Drop-down list**

A drop-down list is a view of the acceptable entry options available for a text box.

**Drop-down list box**

A standard windows control that displays a current setting but can be opened to display a list of choices. The user selects a choice by double clicking on the choice. The user can type into the field, and the system moves the list of choices to the last letter typed.

**DSP08**

Program that displays the current contents of the Replication Holding File (FILE08).

**DSRF02**

Form that records the FILE02 segments that are to be excluded from the remote location.

**DSRSET**

Program that activates the Distributed Administration indicator on the PP-SCOPTS record of FILE01.

**DSSRRL**

Program that will display those remote sites that have not had data distributed to them within the past three days.

### **DSUSET**

Program that deactivates the Distributed Solution indicator on the PP-SCOPTS record of FILE 01.

### **Dynamic SQL**

Statements created by a program that must be interpreted and converted to executable sql statements at run time.

### **Earned income credit**

A refundable amount that reduces the tax owed by certain low-income individuals in the U.S. who meet adjusted gross income levels.

### **Earning**

Money paid in return for work performed or services rendered. In Payroll Administration, earnings are separated by earning numbers into various categories such as regular pay, overtime pay, shift pay, bonuses, and so forth.

### **Earnings category**

Used to categorize similar earnings. For example, all the overtime earnings can be grouped into category 01, all the shift differentials/premiums into category 06, and so forth.

### **EBCDIC**

Extended Binary Coded Decimal Interchange Code; binary code for alphabetic and numeric characters developed by IBM for its computers.

### **eCyborg Interactive Workforce Home**

Button on every page that returns the user to the eCyborg Interactive Workforce Home Page.

### **eCyborg Interactive Workforce Home page**

Home page that displays each time employees log on to eCyborg Interactive Workforce after completing the new user tasks on the New User Home page.

### **EDI**

Electronic Data Interchange. The electronic exchange of business transactions based on standardized guidelines.

### **EEOC**

Equal Employment Opportunity Commission.

### **Effective date**

Date on which an event takes place, for example, an enrollment or benefits plan change.

### **EFT**

Electronic Funds Transfer or direct deposit. Allows employees to authorize the organization they work for to automatically deposit all or part of net pay into one or more bank accounts.

### **EIC**

Abbreviation for *earned income credit* (on page 174).

### **EIN**

Employer Identification Number. A federally or state assigned number used to distinguish tax entities in the US.

### **EL**

Abbreviation for English Language, now called CSL (Cyborg Scripting Language).

### **Electronic documentation**

Documentation that can be viewed, searched, and printed on your computer.

### **Electronic Performance Support system**

Online tools that help users perform their job quickly and efficiently. EPSS can include online help, computer-based training (CBT), electronic manuals, wizards, and so on.

### **Email**

Literally 'electronic mail'. This is a message that is sent to one or more people within or outside of your organization by an automated email software package.

### **Employee cancellation**

An employee cancellation occurs when an employee is cancelled from attending a training class or training program.

### **Employee Database**

The file that contains organization and employee records. This is File02. It was formerly known as the Master File.

### **Employee Database record**

The complete record for an employee. It may be composed of multiple physical records.

**Employee logical Master record**

Complete record for an employee. It may be composed of multiple physical records.

**Employee Number**

Alphanumeric value of up to 10 characters that you define to be used to identify an individual as an employee. It acts as a key to retrieve an employee's record.

**Employee status**

Represents an employee's present standing in terms of activities processed for the employee, for example, active, inactive, part-time, full-time, and so forth.

**Encoding**

The process of applying a badge number to an actual badge. This badge number is then entered on the Badge Number Assignments form to assign it to an employee.

**English Language**

Former name of our fourth-generation programming language, now called Cyborg Scripting Language.

**Enhanced Pay Processing and Reporting**

Pay run and reporting process that can be executed online.

**Enrollment form**

A customer-defined form used by employees to record their benefits elections and any associated dependent and/or beneficiary information.

**Entitlement accrual**

An accumulation of hours for an employee benefit, such as sick leave or vacation time, commonly known as an accrual.

**Entity**

Each Organization Unit, Job, Position, and Incumbent is an entity. Together they are entities.

**Alternate definition:** The agency to which an organization submits its quarterly unemployment insurance reports.

**Entry field**

An area on a screen or browser page where the user can input information.

**Entry form**

An entry form is a form used to enter data.

**Environment**

The host platform and workstations where your system resides, and any communication protocols. Also, a workspace dedicated to a specific processing type. For example: development, test, and production.

**EPSS**

Abbreviation for *Electronic Performance Support system* (on page 174).

**Establishment Reporting**

Establishment Reporting occurs when an employer with several business locations chooses to file wage reports, broken down by location or unit, to the Social Security Administration. Each unit is identified by a four-character code, called an Establishment Number. The employer obtains approval from the SSA to use Establishment Reporting. Establishment Reporting does not apply to 1099s.

**Euro triangulation**

The conversion between one Euro currency and another.

**Event**

The combination of a trigger (changes made to system data) and an action (the creation of an email or letter). Events always consist of these two component halves.

**Exchange rate**

The ratio of the value of one currency to another.

**Exchange rate type**

The interval at which you update the exchange rate; for example, yearly, monthly, weekly.

**Excused absence**

Absences from regularly scheduled work that can be considered as either paid or unpaid time off.

**Execution scripts**

Generic reference to Job Control Language (JCL) for your operating system's command language.

**Extract file**

A data file generated to be used by another system or application.

### **Extract program**

Report generator 7F7F and its subroutines.

This program accesses the P20 file to extract the employee information. It extracts all necessary information and is required to run only once per quarter. It may need to be rerun if there are subsequent adjustments applied to the P20.

### **Federal Insurance Contributions Act**

The United States Federal Insurance Contributions Act imposes two taxes on both employers and employees. Tax is withheld from an employee's wages to finance the Old-Age, Survivor's, and Disability Insurance (OASDI) social security program and the Hospital Insurance (HI) Medicare program. Employers are then required to match the amounts withheld from employees. In Payroll Administration, employee information for FICA-OASDI social security tax is entered on tax record 101 and FICA-HI Medicare tax on tax record 103.

### **FEIN**

Federal Employer Identification Number.

### **FICA**

Abbreviation for Federal Insurance Contributions Act.

### **Field**

A data item on the database. This is usually displayed on a form as a text box.

eCyborg Interactive Workforce specific—A space allocated for a particular item of information. A tax form, for example, contains a number of fields: one for your name, one for your Social Security number, one for your income, and so on. Every field has a name (also called a field label).

### **Filing entity**

Agency to which an organization submits its quarterly report.

### **Filter**

Device used by report to select certain rows of information from the database, thus limiting the amount of data from the database to be viewed in the report.

### **Finished**

Users click Finished when they have completed all information on a checklist or other *Interactive Workforce* page.

### **Flat rate tax**

A US local tax that is calculated as a standard percentage rate and that is calculated in the same way for all employees (that is, factors such as marital status do not enter into the calculation). For many such local taxes, the Tax Authority File does not provide tax specification information. Instead, you need to enter a Tax Specification Record for the tax on a Tax Specification Information form, indicating the tax rate in the Flat Rate text box.

### **Flex credits**

Units granted to an employee in order to purchase benefits under a Flexible Benefits Program.

### **Flex Master Plan**

Defines your Flexible Benefits Program and ties component plans together as a group. Employees are enrolled in the Master Plan and then select the benefit plans in which they wish to participate—for example, medical, dental, and life. Flex master plans are set up in Benefits Administration and used by eCyborg Interactive Benefits to display benefit plans to users for initial and open enrollment.

### **Flex plan**

A benefit plan where, in addition to a core of basic benefits (if applicable), the organization/company allocates to each employee a credit for purchasing additional benefits tailored to their individual needs. Flexible benefit plans may include a flexible spending account.

### **Flexible Benefits Plan**

A specific type of benefit plan that allows employees to select their benefits from a number of benefit plans. This term may be used interchangeably with cafeteria plan.

### **Flexible Benefits Program**

A benefits program in which an organization may allocate to each employee a pool of credits or a monetary amount that is to be used to purchase benefits tailored to individual needs.

**Flexible Spending Arrangement**

A benefits welfare plan set up as an account in an employee's name that is used to reimburse the employee for certain personal expenses. In the United States, these accounts are provided by employers as a way for employees to pre-fund dependent care, legal services, or medical expenses with pretax currency.

**Floating toolbar**

Toolbar that is located within the program window.

**Folder**

Logical organization device for the content of a Cognos catalog.

**Foreign currency**

The currency into which or from which you must convert.

**Form**

Section of the Work Area in which data is entered and displayed, including text boxes and other controls. This was formerly known as a screen.

**Form 300**

Log of Work-Related Injuries and Illnesses, used to classify work-related injuries and illnesses and to note the severity and extent of each case. The delivered OSHA Form 300 Log of Work-Related Injuries and Illnesses report (20-RPT) is a facsimile of the Form 300.

**Form 300A**

Summary of Work-Related Injuries and Illnesses, used to show the totals for the year in each (illness or injury) category. The delivered OSHA Form 300A Summary of Work-Related Injuries and Illnesses report (21-RPT) is a facsimile of the Form 300A.

**Form 301**

Injury and Illness Incident Report. This must be completed when a recordable work-related injury or illness has occurred. This must be completed by the employer within 7 calendar days following receipt of information that the recordable work-related injury or illness occurred. This form must be kept on file for five years following the year to which it pertains. The delivered OSHA Form 301 Injury and Illness Incident Report (22-RPT) is a facsimile of Form 301.

**Form area**

An area of the window that contains a form.

**Form Builder**

A tool for use with The Solution Series for designing forms.

**Formal education**

Education that is obtained from a college or university.

**Forward**

Displays the next page.

**FSA**

Abbreviation for Flexible Spending Arrangement.

**FTE**

Full Time Equivalent. An FTE is the ratio of total working time to the time that represents full time employment for a single employee. For example, an FTE of .5 means working time is half of the time that represents full time employment.

**FTP**

File Transfer Protocol. A means of allowing a user on one computer to transfer files to and from another computer over a network

**Full Time Equivalent**

The ratio of total working time to the time that represents full time employment for a single employee. For example, an FTE of 0.5 means working half of the time that represents full time employment.

**Function keys**

Keys F2 through F12 on the keyboard that can be assigned to a bookmark or favorite.

**Funeral days**

Absences from regularly scheduled work due to a funeral, which at the discretion of the organization, can be considered as authorized or unauthorized, paid or unpaid time off.

**Gap analysis**

Comparison of a current state of being with a desired state of being. For example, you could perform a skill or competency gap analysis on individual employees or on the workforce as a whole, comparing the existing state of skills and competencies with the required state or level of skills and competencies.

### **Garnishment**

A legal procedure authorizing a deduction from an employee's earnings to satisfy a legal requirement.

### **General ledger**

File that provides a balanced payroll journal for the period. This file contains journal entries for labor expenses, withheld deductions, income, disability, UI, and other withheld taxes, net pay, and company-paid taxes. The interface may also be produced on paper.

### **General ledger interface**

A file that provides a balanced payroll journal for the period. This file contains journal entries for labor expenses, withheld deductions, income, disability, UI, and other withheld taxes, net pay, and company-paid taxes. The interface may also be produced on paper.

### **Go to details**

Displays a new page with detailed information. Used on summary pages.

### **Graphical User Interface**

The Solution Series provides integrated human resource and payroll functionality via the Microsoft Windows Graphical User Interface. These are the elements that display on your screen.

### **Grievance**

A formal complaint made by an employee against the organization usually because of an unsatisfactory working condition or other work-related dispute.

### **Gross wages**

The total of all earnings paid to an employee.

It is stored in the Total Pay (field 119 of the US Tax Authority File) field of the employee's US FICA tax record 101 (FICA-OASDI). This figure appears on the Combined Register (2222) report as Total Pay. It does not appear on US W-2 forms.

### **Group box**

A standard Windows control that groups a set of controls.

### **Group plan**

Defines any number of benefit plans tied together as a group. Group plans are used to define common eligibility and to cluster plans for reporting purposes.

### **GUI**

Abbreviation for Graphical User Interface.

### **Handicap**

Having a physical or mental disability that substantially limits activities especially in relation to employment or education.

### **Health and safety profile**

Data on the employee record that includes information such as the employee's blood type, language, physician, emergency contacts, and any disabilities.

### **HED**

Acronym for Hours, Earnings, and Deductions. Each earning or deduction must be established in The Solution Series with a unique identifying three-digit code. HEDs are used to record pay, hours worked, and deduction amounts and arrears for each employee.

### **Help**

Hot spot on an eCyborg Interactive Workforce page that displays step-by-step directions for completing the page.

### **HIPAA**

Acronym for Health Insurance Portability and Accountability Act of 1996. HIPAA is a law that protects citizens from being denied health insurance coverage due to a pre-existing condition, and protects the privacy and security of citizens' healthcare information by enforcing standards for the manner in which health information is transmitted electronically.

### **History record**

Part of an employee's payment history; a snapshot of a check paid to an employee or an adjustment made to an HED or tax.

### **Holiday days**

The time off that all employees are entitled to based on the decision of the organization or government regulation.

### **Home location**

The country in which an employee permanently resides.

**Home page**

The main page of a Web site that generally serves as an index or table of contents to other documents stored as pages on the site.

**HTML**

Abbreviation for **HyperText Markup Language**, the authoring language used to create documents on the World Wide Web. HTML defines the structure and layout of a Web document by using a variety of tags and attributes.

**I-beam**

Symbol that identifies the insertion point.

**ICESA**

Interstate Conference of Employment Security Agencies. Also used to refer to the specifications and instructions for reporting to participating states through magnetic media.

**Import facility**

A tool delivered with The Solution Series that moves data from an external source to any organization or employee form.

**Import record**

A line in a spreadsheet or delimited file that contains employee or company data.

**Impromptu**

Tool for interactive database reporting. Reports are built from the database using a pre-defined catalog.

**Inactive plan**

A benefits plan that no longer allows employee enrollment.

**Inactive tax record**

An employee tax record that is no longer in effect for a given employee. Neither wages nor taxes are accumulated for the particular tax record. However, any wages and/or taxes already accumulated remain until clearing is performed. Such clearing is usually performed in preparing the Employee Database for a new year. The inactive records can be deleted at this time. The process of making a tax inactive is called deactivating.

**Incumbent**

An incumbent is an employee linked with a specific position. The linking of an employee with a Position is an incumbency. An employee may be linked to more than one position; in other words, an employee with multiple incumbencies. A position to which more than one employee is linked has multiple incumbents.

**Information-level security**

These records grant access to employee and table data via specific password records.

**Initial Administrator**

Only user whose user ID and password are created during installation. The initial administrator always has authority to all administrative functions: eCyborg Interactive Workforce, Human Resources Administration, Benefits Administration, and Payroll Administration, and can assign administrative roles to others by creating administrative user IDs and passwords.

**Initial passwords**

Password generated by eCyborg Interactive Workforce for each user ID extracted from The Solution Series. Users must create a user-defined password when they log on to eCyborg Interactive Workforce for the first time.

**InitialAdmin**

See Initial Administrator.

**Inquiry form**

An inquiry form is a form used to view data already entered.

**Instructional text**

Any paragraph(s) on the page that explain the function of the page or fields to the user.

**Internal candidate**

An employee of your organization who is applying for another job or position in your organization.

**Internet**

A global network connecting millions of computers.

**Intranet**

A network belonging to an organization, usually a corporation accessible only by the organization's

members, employees, or others with authorization and used to share information.

### **Investment funds**

Different options or accounts available to employees for allocating their contributions, usually applicable to thrift/savings plans.

### **IPEDS**

Acronym for Integrated Postsecondary Education Data System.

*(Taken from NCES website at  
<http://nces.ed.gov/ipeds/AboutIPEDS.html>)*

“NCES has established the Integrated Postsecondary Education Data System (IPEDS) as its core postsecondary education data collection program (prior to IPEDS some of the same information was collected by the Higher Education General Information Survey-HEGIS). It is a single, comprehensive system that encompasses all identified institutions whose primary purpose is to provide postsecondary education.

IPEDS consists of institution-level data that can be used to describe trends in postsecondary education at the institution, state and/or national levels. For example, researchers can use IPEDS to analyze information on 1) enrollments of students, undergraduate, first-time freshmen, graduate and first-professional students by race/ethnicity and gender; 2) institutional revenue and expenditure patterns by source of income and type of expense; 3) salaries of full-time instructional faculty by academic rank and tenure status; 4) completions (awards) by type of program, level of award, race/ethnicity, and gender; 5) characteristics of postsecondary institutions, including tuition, room and board charges, calendar systems, etc.; 6) status of postsecondary vocational education programs; and 7) other issues of interest.”

### **IRP/BSS**

Information Reporting Program/Bulletin Board

### **IRS/MCC**

Internal Revenue Service/Martinsburg Computing Center

### **JDBC**

Java Database Connectivity Application Programming Interface. An application programming interface for

accessing tabular data sources from the JAVA programming language.

### **JMNTRUN**

Script used to assign check numbers and create history records.

### **Job**

A 'Job' is a generic description of a role within the organization—for example, Manager.

### **Job assignment**

A job associated with a particular employee.

### **Job code**

A designation for a job assignment.

### **Job streams**

A generic reference, Job Control Language, for your operating system's command language. Control statements that execute programs.

**Alternately:** Jobstreams

### **Job type**

A generic category that further defines a particular job.

### **JPAYRUN**

Script that is used to calculate pay and create payroll reports.

### **JQTRRUN**

Script that executes the new quarterly reporting process.

### **Jury duty**

This is compulsory service on court appointed juries. Employers are required by law to excuse jury duty related absences. They are not, however, required by law to pay the employee during this time away from the job.

### **JXP5QTR**

Stand-alone script used on all platforms to extract and compile the new quarterly COBOL program.

### **KEY-PE**

Utility program used to rebuild the QUERY Alternate Keys.

**KEYDEL**

Utility program used to delete existing QUERY Alternate Keys.

**Label**

Text that describes the information the user enters into the field.

**Labor record**

A record containing the hours, amounts, associated charge-to control levels, and function assigned on the employee's Payroll Home Location/Pay Allocations form (GG-SCR).

**Large number**

The Multicurrency Payroll component contains a large-number format time entry form that accepts a nine-digit number as opposed to the seven-digit maximum in the standard Payroll Administration.

**LDEFAULTS**

Term used for a text/list box default template.

**Leave of absence**

Occurs when an employee leaves the organization for a period of time, usually temporary, for personal reasons such as medical leave.

**Link/jump**

Reference to another document. Links (or hotspots) appear as underlined text. When you move your cursor over a link, the cursor changes to a hand. Click the link to access or jump to the referenced document.

**LIS file**

Standard file name used by all completed report files that are launched online. Files are automatically assigned this extension by the Online Initiation of the Pay Processing and Reporting feature.

**Local currency**

The currency in which the employee is paid.

**Local currency country**

The work location for tax purposes.

**Log file**

Informational file generated when a report is launched online. The log file is used by the Process Monitor to communicate the status, progress, and completion of a report.

**Log off**

Logs the user off the system. When referring to the Log Off button, use initial caps.

**Logical Employee Model**

A collection of default employee information that is used to create a model. Logical Employee Model templates are used when hiring new employees to save time and ensure that critical information is established consistently and correctly. These were formally known as LMODELS.

**LPI**

Lines per inch.

**Mailing address**

An address, other than your legal residence address, to which you have your mail sent.

**Maintenance payroll run**

A maintenance payroll run automatically updates organization and employee records, but it does not process time entries, calculate pay or generate payments, pay slips, or deposit advices. It is also used to create payment history records.

**Major activity**

Event that causes a change in an employee's employment status, such as a new hire, termination, or rehire.

**Mandatory field**

A field that requires the user to enter information before the user can exit the form or page.

**Manual payment**

A payment not created within The Solution Series.

**Map file**

Stores the predefined relationships between an import file and a form.

**Mass time entry creation**

Creating time entries for a group of employees through one program execution, such as for a paid holiday.

**Master File (0202) report**

A report that produces a formatted display of the data in an employee's current batch Employee Database record. This includes the wages and taxes accumulated for the employee, covering current, month-to-date,

quarter-to-date, and year-to-date information for individual tax codes. It is report generator 0202.

### **Matrix ID**

Unique identifier for each pay-for-performance matrix.

### **Menu**

A list of choices; the choices are generally links that take the user to another screen or page.

### **Menu bar item**

A menu that appears on the menu bar.

### **Message area**

An area of the window that contains messages or selection lists relevant to the current form. The Message Area can be turned on or off.

### **Method code**

One of many specific routines (usually delivered and identified by a two-character code) used to calculate earnings and deductions.

### **Midpoint**

The middle of the span of currency from the minimum to the maximum of the employee salary grade.

### **Minimart**

Relational tables you create so you can insert data from your Subset data extractions.

### **MMREF**

Magnetic Media Reporting and Electronic Filing. Also used to refer to the specifications and instructions for reporting to participating states through magnetic media. See Social Security Administration publication MMREF-1 for specifications and instructions for reporting quarterly wage information to participating states through magnetic media.

### **Model (PowerPlay)**

File (with the extension .pye) that contains definitions of PowerCube dimensions and measures, queries that define the data sources, and other details needed to create a defined cube. Defined models reflect performance requirements as well as requirements of the final user of the cube.

### **Monetary prerequisites**

A privilege or profit that an employee is entitled to that is incidental to regular wages or salary.

### **Moving expenses**

The expenses incurred by an employee due to moving from one location to another for employment purposes.

### **Multicurrency costing records**

Records that maintain a history of the deduction and deposit transactions for each exchange rate.

### **Multiple master**

A file compression technique that duplicates the current employee Permanent Master Record as many times as there are payments to that employee during one pay period. These multiple masters are detail records reflecting the amounts for the payment being made (current), and the adjusted MTD, QTD, and YTD totals. The system uses multiple master records to create history records showing the current payment figures only.

### **NACHA**

American not-for-profit trade association that develops operating rules and business practices for the Automated Clearing House (ACH) Network and for other areas of electronic payments.

### **Navigation bar**

In eCyborg Interactive Workforce the Navigation bar shows the name of the page you are using, for example, 'Mailing Address'. The top line of the Navigation bar shows the path you took from the Home page to reach the present page. Links on the Navigation bar let you return to the home page or log off the system.

### **Navigator**

Left pane of the work area which forms the main method of moving through the forms. From the Navigator users select the component, process, and task in which they are interested.

### **NCES**

Acronym for National Center for Educational Statistics.

*(Taken from NCES website at <http://nces.ed.gov/ipeds/AboutIPEDS.html>)*

Primary federal entity for collecting and analyzing data that are related to education in the United States and other nations.

**Net credit method**

A method allocating flex credits. An employee's cost of benefits is calculated as either a net cash earning or a net deduction from the employee's pay. The net amount is the difference, either plus or minus, between the credits allocated to the employee and the cost of his or her flex benefits choices.

**New hire**

Process of hiring a new employee for your organization.

**New period pay run**

Pay run for which the earning and deduction cycles are incremented. New-period pay runs are typically part of a regular pay schedule, and do not include off-cycle runs.

**New user**

A user of eCyborg Interactive Workforce who has not yet completed reviewing and updating their personal information on the New User Home page.

**New User Home page**

Home page that displays for new users of eCyborg Interactive Workforce until they complete reviewing and updating their personal information.

**Node**

A Distributed Location.

**Node ID**

A unique 5-position identifier for a node. The naming convention is defined by the user.

**Notional value**

A value input for informational purposes only.

**Number registered**

This is the number of employees registered for a training class. It is updated and displayed on the Class Schedule form.

**Object**

Each System Control Repository record type is assigned an object code. A single record type can have several object codes assigned to allow limited display.

**Object key**

A field that allows you to specify the System Control Repository record group you want to display. The value

of this field is dependent on the type of information you want to display.

**Obsolete plan**

A benefits plan that will no longer be used.

**Off cycle**

An off-cycle payroll run is an additional payroll for the period just completed. An off-cycle payroll run is commonly used to process nonstandard payments, such as bonuses. It is sometimes referred to as an additional or bonus payroll run.

**Off-cycle pay run**

Additional payroll for the period just completed. An off-cycle payroll run is commonly used to process nonstandard payments, such as bonuses. It is sometimes referred to as an additional or bonus payroll run.

**One-stop document**

Documentation that contains reference information, step-by-step procedures, and training exercises.

**Online**

Turned on and connected, for example, printers are on-line when they are ready to receive data from the computer. Users are considered on-line when they are connected to a computer service through a modem. That is, they are actually on the line.

**Open enrollment**

A period of time during which employees can enroll in or change their benefit choices for the upcoming year, generally in October or November.

**Operator ID**

A four-character code that identifies the user to the system.

**Option**

An item in the option list for a field. This was formerly known as a codeset item.

eCyborg Interactive Benefits and Benefits Administration specific—In Benefits, the plan coverage that an employee selects, such as single or family coverage.

### **Option button**

A standard Windows control that allows you to select from a fixed set of mutually exclusive options (previously known as radio button).

### **Option list**

A list of valid options and associated descriptions from which you may select an appropriate entry. This was formerly known as a Codeset.

eCyborg Interactive Workforce specific—Options available in The Solution Series that the eCyborg Interactive Workforce administrator loads in to eCyborg Interactive Workforce. The options are then available in the drop-down list boxes in eCyborg Interactive Workforce.

### **Organization**

A group of employees who are employed in a common structure, governed by the same set of rules or policies, and eligible for the same earnings and deductions. For example, your organization may be structured into parts that represent employee groups such as active, union, retirees, applicants, and so forth.

The organization is the highest level in The Solution Series structure and is identified by a six character code. All records are organized by organization.

Formerly known as a company or Control 1-2.

### **Organization Level 3**

A customer-defined value used to determine the breakdown of an Organization for Human Resource reporting or selection purposes. This control level may be translated to a division, plant site, section, and so forth, as defined by you.

### **Organization Level 4**

A customer-defined value used to determine the breakdown of an Organization for Human Resource reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Organization Level 5**

A customer-defined value used to determine the breakdown of an Organization for Human Resource reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Organization Level 6**

A customer-defined value used to determine the breakdown of an Organization for Human Resource reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Organization levels**

Hierarchy of values used to determine the breakdown of an organization for reporting purposes. The values are customer-defined.

### **Organization Number**

A six-character user-defined code that represents an organization; the highest level of the organizational structure in Payroll Administration.

Formerly known as a Control 1-2.

### **Organization Unit**

An organization unit ('Org Unit') is a grouping of Positions within an organization (for example, Accounts Department).

### **Organization Validation table**

A table that validates that an organization is valid and payments can be made.

### **Organization-specific tax setup**

A method of implementing Tax Specification Records in which each organization involved in tax processing contains all the specification records required to process taxes for its employees, as opposed to a common tax organization.

### **OSHA**

U.S. Department of Labor Occupational Health and Safety Administration.

### **Override file**

A file used to maintain COBOL or Report Generator changes to the system.

### **P2EDIT**

Transaction editor program. This is the first program executed as part of the batch payroll process. This program performs edits on the input transactions.

**P4CALC**

Program that performs payroll calculations and updates to the sequential Batch Master File during a payroll run.

**P5PRNT**

Program that formats and creates all output, including payments, reports, tapes, and records that must be recycled to future payroll runs.

**Packaged reporting**

A processing mode in which a job is scheduled to be run at a certain time.

**Packaged reports**

Delivered reports developed using a Fourth Generation scripting language, Cyborg Scripting Language (CSL).

**Paid absence**

Employee absence that will be paid by the organization. A time entry will be created for this absence.

**Parallel run**

The process of executing the same programs simultaneously on two separate systems to obtain the same or similar results.

**Parameter form**

A form that is displayed when certain programs are called from the Navigator or menus. The form facilitates entering parameters for the program.

**Password**

A secret series of characters, generally user defined, that enables you to access a computer, a software application, or a file. On multi-user systems, each user must enter his or her password before the computer will respond to commands.

In eCyborg Interactive Workforce, the password ensures that unauthorized users cannot access user-specific information.

**Password aging**

The period of time that elapses before a user-defined password expires and the user must change his or her password.

**Pay allocation**

A means of allocating, on a percentage basis, employee labor hours and amounts to multiple sets of control

levels 3 through 6 and function to accurately reflect employees whose labor must be charged to more than one area within an organization.

**Pay document**

A pay slip or deposit advice with its associated pay stub.

**Pay extract**

Process that extracts data from the online Employee Database.

**Pay frequency**

The interval at which a group of employees is paid. Examples are weekly and semi-monthly. Also referred to as a payroll period.

**Pay merge**

Process that updates the Employee Database with the current data from the recently processed pay run.

**Pay schedule**

A predetermined schedule for a calendar year, identifying period-end and payment dates for each pay frequency.

**Pay stub**

A pre-printed form, corresponding to a check or deposit advice that lists all earning, gross pay, taxes, deduction, and net pay information for an employee.

**Pay-for-performance matrix**

Chart representation of the variables that result from the combination of salary increase information, how much to give and when.

**Payment history record**

A record documenting the detail information for a payment or adjustment. Multiple payment history records may be generated for an employee, reflecting multiple adjustments or payments. These records include all earning, deduction, and tax information included in the payment or adjustment.

**Payroll home location**

The location where the employee is normally assigned to work and where labor distribution information is charged. An employee's home location comprises specific Payroll Levels and is always assigned Allocation Number 01 on the Payroll Home Location/Pay Allocations form. The Function field may

also be used as part of a home location, depending on your specific requirements.

### **Payroll Level 3**

A customer-defined value used to determine the breakdown of an organization for Payroll reporting or selection purposes. This control level may be translated to a division, plant site, section, and so forth, as defined by you.

### **Payroll Level 4**

A customer-defined value used to determine the breakdown of an organization for Payroll reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Payroll Level 5**

A customer-defined value used to determine the breakdown of an organization for Payroll reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Payroll Level 6**

A customer-defined value used to determine the breakdown of an organization for Payroll reporting or selection purposes. This control level may be translated to a division, plant site, section, and so on, as defined by you.

### **Payroll period**

A defined period of time for which an employer pays wages to employees.

### **Payroll Process Control**

A series of forms used during the Payroll Process to determine the type of run (payroll run or maintenance run). Allows you to specify the pay frequencies to be paid and which reports are to be produced.

### **Payroll run**

An automated process that updates organization and employee records, processes time entries, calculates employee pay, generates pay documents and payroll reports, including the Combined Register. It also produces a variety of special interface outputs.

### **PCL**

Printer Control Language.

### **PDF**

Portable Document Format. A file format that captures formatting information from a variety of desktop publishing applications, making it possible to have formatted documents appear on the screen and be printed. To view a file in PDF format, you need Adobe Acrobat Reader, a free application distributed by Adobe Systems.

### **Peer-group appraisal**

Appraisal that uses performance evaluations completed by an individual employee's co-workers or project team members.

### **Pending de-enrollment segment**

Plans for which an employee is enrolled, but has lost eligibility, as listed on the Pending Plan Enrollment/De-Enrollment form.

### **Pending eligibility segment**

Plans for which an employee is eligible but not enrolled, as listed on the Pending Plan Enrollment/De-Enrollment form.

### **Performance appraisal**

A periodic assessment and ranking of an employee's skills and accomplishments.

### **Performance appraisal rating**

A method of ranking the performance of an employee during a given period using options ranging from 1-Outstanding to 5-Unsatisfactory.

### **Performance rating**

A method of ranking the performance of an employee during a given period using options ranging from 1-Outstanding to 5-Unsatisfactory.

### **Performance-related pay**

Monetary payments made to employees based on how well an employee has fulfilled job expectations.

### **Permanent Master Record**

Individual employee record in the Employee Database. This record contains all the information about an employee, such as name, address, pay frequency, and salary.

Permanent records of employee earnings, taxes, and deductions in the Employee Database. This is the only record group that contains month-to-date, quarter-to-date, and year-to-date accumulations.

**Perquisites**

Property or privileges extended to an employee.

**Personal days**

Authorized absences that are generally considered as paid time away from regularly scheduled work, but can be either paid or unpaid.

**Phonetic keys**

The keys you use to access employee data using the phonetic spelling of an employee's last name.

**Pixel**

The smallest rectangular area of an image on a screen.

**Plan deactivation**

A process that makes a plan inactive and prevents future employee enrollment.

**Plan ID**

A three-position, alphanumeric identifier for a plan in the system.

**Plan shutdown**

The process of de-enrolling an employee from all benefits plans because of a separation activity.

**Plan year**

The 12-month period over which a salary budget is effective.

eCyborg Interactive Workforce specific—The calendar, policy, or fiscal year in which the records of a Benefits plan are maintained.

**Policy tables**

Highest level tables that are used to record the generic (or master) rules for an organization or group of employees. These included your organization's rules relating to working time procedures, such as clocking in and out, docking for lateness, and overtime. Each policy consists of a Policy Master table and one or more Policy Activities table.

**Pop-up menu**

A menu that appears when you use the second mouse button within the system. This menu contains context

sensitive commands and options that relate to the object you have clicked on.

**Portable document format**

See PDF.

**Position**

A specific role with an organization—for example, Accounts Manager.

**Alternative definitions:**

1. to place an object in a specified location.
2. a location, particularly in a record

**Position Administration Control Number**

Two-character alphanumeric value that tells Position Administration which tables to use for a specific company.

**Position complement**

A 'Position complement' is the value of a Position. The organization complement is the total value of all Positions included in the complement.

**Position in range**

The difference between a given salary and the minimum of the salary range, divided by the difference between the range's maximum and minimum, and expressed as a percentage.

**Posttax**

A contribution made after taxes have been withheld from earnings.

**PowerPlay**

Business intelligence tool for multidimensional analysis of corporate data, also known as Online Analytical Processing (OLAP).

**Premium**

The amount of money an organization agrees to pay an insurance company for a policy or annuity, or the amount contributed by an employee to the employer to cover the employee's portion of the total premium.

**Prenotification**

Informing a bank or credit union that an employee will be using direct deposit with them in the future. To be safe, fill out the Direct Deposit Information form (H9-SCR) two pay periods in advance of the first deposit

date. This ensures that a prenotification record is provided to the bank or credit union in a timely manner.

### **Pretax**

A contribution made before taxes have been withheld from earnings.

### **Primary account**

The account set up in eCyborg Interactive Workforce to receive an employee's pay or reimbursement checks. After deductions and deposits to additional (secondary) accounts, the remainder of pay is deposited into the primary account.

### **Process**

A subset of a component that logically groups tasks on the Navigator or menu. For example, the process 'Maintain Employee Details' contains tasks such as 'Basic Employee Information' and 'Personal Information'.

**Alternate definition:** An action that brings about a result.

### **Process bar**

The graphical representation of a process on the navigator. Each process bar is within a Component.

### **Process Monitor**

Utility that allows you to monitor the progress of reports and payroll processes launched online.

### **Program**

A program is a series of classes being administered using Training Administration. For example, 'The Training Schedule for January-June 2004' may be a program consisting of eight different classes.

#### **Alternative definitions:**

1. A form or other set of instructions within the system, accessed directly from the Command dialog box. For example, form EF-SCR is a program.
2. A set of coded instructions that enables a computer to perform a desired sequence of operations.

### **Prompt**

Expression set up to enable users to filter data by typing or picking a value, or a series of values, when the report opens.

### **Protected amount**

The amount of disposable income protected from garnishment in the US. This amount may vary from state to state.

### **Prototype HED**

An HED defined on a benefits form for use in recording employee/organization contributions when an employee is enrolled in a benefits plan. This allows the setup and maintenance of payroll deductions using Benefits Administration.

### **Provider**

A provider is an instructional institution, organization, or person who is available to teach training courses.

### **Push button**

A button on the interface that appears depressed when clicked on (now known as command button).

### **Qualifier**

A drop-down list (field) referring to a data element on the HIPAA 834 (Benefit Enrollment and Maintenance) and 820 (Group Premium Payment for Insurance Products) transmissions used to describe the type of information supplied in a related data element (field). For example, a Telephone Qualifier drop-down list identifies whether the telephone number supplied in a Telephone Number text box is a home, work, or fax telephone number.

### **Quarter-end P20**

P20 is a file structure used in batch payroll processing. The quarter-end P20 file contains payroll data that is used in the quarterly processing.

### **Quarterly Processor**

Combination of forms, report generators, and COBOL programs that extracts quarterly wage reporting information and produces all of the selected quarterly reports and formatted filing output (paper, tape, or diskette) for all selected employees in all selected states.

### **Quartile**

Points that represent the division of a salary grade range into four equal parts.

**Query alternate keys**

The keys you use to access the employee master record in an order other than by primary key.

**Query primary keys**

The keys you use to direct your QUERY program to a record type.

**Quick Hire**

The process of hiring an employee by entering one two-panel form with the required data elements rather than entering a series of forms.

**Radio button**

A button on a form that selects an option, the radio buttons that make a field are mutually exclusive (now known as an option button).

**Recall**

Return a laid-off employee to active status, usually with no affect to benefits.

**Reciprocal taxation**

Reciprocal tax withholding refers to agreements made between US states and (or) localities regarding income tax calculation and reporting for compensation paid to an employee who lives in one state or locality and works in another.

**Reconciliation number**

The unique identification number printed on each pay document. Reconciliation numbers are used in the payment reconciliation procedure to verify cleared and outstanding payments.

**Record**

A complete set of fields, such as the fields that make up a tax form or a name and address record.

**Alternate definitions:**

1. To set down for preservation in writing or other permanent form.
2. The reporting format for magnetic media as in Tape Record Type such as ICESA, TIB-4, or MMREF
3. Information or data on a particular subject collected and preserved

**Recruitment**

Process of finding and hiring new employees who meet the needs of your organization.

**Recycle File**

P05IN; A file that contains employee data and pay document information required for payment reconciliation. It also contains time entries to be processed and paid at a later date. This file is used to pass data to the next payroll or maintenance run.

**Registration**

Registration is the act of enrolling an employee in a class.

**Registration number**

A three-digit registration number is assigned to employees for tracking purposes when they register for a training class. This enables the order in which the employees registered to be viewed.

**Regulatory bulletin**

Contains updates to varying components of The Solution Series necessary to maintain regulatory compliance. Tax specifications supplied are delivered in regulatory bulletins (RBs).

**Rehire**

The process of hiring a former employee of your organization. Typically, a break in service is incurred and benefits must start over (usually requiring a new adjusted seniority date if used in benefits tracking).

**Reimbursement account**

The account into which employee's travel and other expense type reimbursement checks are directly deposited.

**Reinstatement**

The process of returning a former employee to active status within a certain time period (such as 90 days), thus qualifying the employee to have certain benefits restored to the original hire date.

**Reject time**

The point at which an error condition will occur. An error condition must be manually corrected/approved and approved before a time entry can be generated by the system.

### Relocation

The process of moving an employee from one organization to another geographic location, whether the move be domestic or international. This process also applies to applicants who are being relocated as part of the hire process.

### Remaining net pay

The 'bucket' of money that is left after all employee deductions and taxes have been taken from the employee's gross pay. This 'bucket' of money can then be used for multiple deposits if the enterprise sets up multiple deposit HEDs.

### Replication

The automatic process of writing changes made in the Employee Database and option lists and tables in the System Control Repository to the Replication Holding File (FILE08).

### Replication Application

English Language program (DSAPLY) that reads records from the Replication Packet File (FILE20) produced by the Replication Reception program (DSRECV) and updates the System Control Repository and Employee Database accordingly.

### Replication Distribution

Two COBOL programs that work together to distribute and receive updates. The Replication Distribution Program (DSTRIB reads either the Replication Holding File (FILE08) or a Replication Packet File (FILE20), selects data applicable to a specific DL and writes all necessary data to a new output-only Replication Packet File (FILE21). The resulting FILE21 will be processed on the remote DL via the DSRECV Replication Reception program.

### Replication Holding File

FILE08. This file contains additions, changes, and deletions to the System Control and the Employee Database. Data is distributed from and written to this file, based on the data distribution rules configured for the target DL by the source DL.

### Replication Packet File

(FILE21/20). This file contains data changes and is created specifically to update a target DL. This information may include Company/Employee data,

tables and option lists, and time entry and adjustment records.

### Report

Batch program that creates report extract records.

#### Alternate definitions:

1. A detailed accounting of an activity; for example, the quarterly report of unemployment insurance details to a filing entity.
2. As in 'Report Select character', this represents the selection frequency for this output.

### Report Generator

A program that produces the batch payroll and the batch payroll reports.

### Report Group

A series of packaged reports that are created using the Report Group Activities form and are run together.

### Report Group Scheduler

This is the program that allows you to schedule reports. This was formerly known as the Report Scheduler.

### Report parameters

Specific guidelines for determining the information to be processed by a given report or program.

### Report Viewer

Utility that allows you to view, search, print, save, and delete reports that are launched online.

### Reporting Administration

HRMS reporting and analysis tool that links the database functionality found in The Solution Series with sophisticated, yet easy to use business intelligence tools from Cognos®.

### Requisition

A formal request to fill a vacancy or vacancies.

### Requisition candidate

A candidate for a vacancy represented on a requisition.

### Requisition limit

A total unit value of a requisition.

### Requisition unit

The value of a requisition expressed as an FTE, hours, salary or headcount.

**Retirement**

Occurs when an employee retires from the organization.

**Return**

The activity of an employee returning as an employee to active status, usually following a leave of absence.

**Alternative definitions:**

1. key on keyboard used to perform a carriage return; can also be known as Enter.
2. to send back to the originator; for example, the Tape Return Information form (QT-SCR) gives the address to which a filing entity would return a magnetic tape containing errors.

**Review process**

A method used by an organization to evaluate an employee's salary or performance in a standard, timely manner.

**Roll-up reporting**

Option that enables packaged reports to be processed within organizations (roll-up).

**Rotation pattern**

A way of describing the working pattern for a group of employees (crew) who regularly work different shifts. A crew is a group of employees who together regularly work the same schedules according to a rotation pattern.

**Safety standards**

Legally mandated workplace safety standards.

**Salary budget record**

Defines, for each employee, the budgeted increase amount, percentage, and effective date for a specific salary plan year, and the prorated effect of this increase on the budget in terms of amounts and percentages for each employee.

**Salary grade**

A range of salary amounts associated with a particular job.

**Salary grade range**

A range of salary amounts associated with the salary grade for a particular job.

**Salary plan**

A set of rules or guidelines used to budget for salary increases for the coming year.

**Salary plan year**

A 12-month period over which a salary plan is effective.

**Salary range**

The span of salary amounts from the minimum to the maximum of the employee salary grade.

**Salary review**

A periodic evaluation of an employee's compensation.

**Salary review authorization form**

Hard copy format of the employee criteria necessary to review and approve proposed salary increases.

**SAT file**

The Solution Series form appearance table. Simple text file that reflects the form's layout.

**Save Changes**

Saves the page (form) the user completed. (When you click 'Save Changes', eCyborg Interactive Workforce saves the information on the page whether or not the user made changes.)

**Schedule Activities table**

Identifies activity types for each point in a work day where the process of clocking in and out should be dealt with. Each Schedule activity also contains time parameters that will be used to calculate whether an employee will be docked or credited time.

**Schedule assignments**

Also referred to as a schedule. This term refers to the details of the Schedule Master tables to which an employee is assigned. These details include the date the assignment took place, the Schedule Number and Sub-Schedule Number, and (if applicable), the crew to which the employee is assigned.

**Schedule error**

Occurs when a clock transaction (ring) time falls outside of an employee's schedule reject times.

**Schedule Master table**

Used to set up your organization's Time and attendance rules (such as HEDs and the minimum number of hours

an employee must work before a meal deduction is made). A Schedule Master table is associated with a Calendar Routine, earnings Code, and Shift Premium table by entering the appropriate identifier.

### **Schedule number**

A unique three-character alphanumeric identifier used to partially identify a schedule table.

A Schedule Master table is used to associate schedules with a Policy Master table. A Schedule Master table can also be used to override values held on its corresponding Policy Master table.

Employees are assigned to a Schedule Master table. Schedule Master tables may be created for individual employees, or for use by groups of employees or by crews.

### **Screen**

Now known as a form.

### **Scroll bar**

When information on a page takes up more than one screen of your monitor, the system adds scroll bars to the right side of the screen. On the scroll bar:

- Click the up arrow to move line by line to the top of the page
- Click the down arrow to move line by line to move to the bottom of the page
- Click the double arrows to move several lines up or down the page

Click and drag the bar in the scroll area to manually move up or down the page.

### **Search argument**

The value from an employee's master record used to search benefits tables to apply plan rules to specific groups of employees.

### **Search type**

The definition of a field from an employee's master record to use as the search argument.

### **Secondary account(s)**

Additional account or accounts at financial institutions that employees set up in eCyborg Interactive Workforce receive a portion of their pay. A primary account must be defined before an employee can set up additional accounts.

### **Security Officer**

The assigned employee who is responsible for the setting up and monitoring of the security your system.

### **SEIN**

State Employer Identification Number; required by some states for wage and tax reporting.

### **Selectable File08 Display**

Program that displays FILE08 information based on a selectable record key, starting key, time, and date.

### **Self-adjusting taxes**

Taxes for which the system automatically recalculates the tax on a cumulative year-to-date basis on each payroll run.

In the U. S. these include FICA taxes: Social Security (tax record 101) and Medicare (tax record 103). The purpose of this calculation is to avoid any differences (of pennies) in FICA tax paid versus FICA tax due at year-end due to rounding on a pay period basis. In addition, certain state disability taxes and employee-paid state unemployment insurance taxes also self-adjust.

### **Sequential Master File**

P20IN; The batch processing version of the Employee Database. This file contains organization and employee data, tax tables, and the object code for programs.

Also known as the Batch Master File.

### **Service interruption**

A period of time during which an employee did not maintain an active working status in the organization.

### **Service method**

A calculation option list that determines the method for calculating credited service.

### **Session**

When users log onto a software application, they begin a session. When they log off, they end the session.

**Alternate definition:** The period of time during which a class is held.

### **Shift**

An employee schedule assignment for a given day. For a rotation pattern, this is a Sub-Schedule Number.

**Alternative definition:** key on keyboard, typically used to describe key combinations for a shortcut key.

### **Shift Differential/ Premiums**

Premium (or allowance) added to an employee's regular earnings, overtime earnings, or both. It is represented by a shift code or HED Number.

### **Shift premium**

A premium (or differential) added to an employee's regular earnings, overtime earnings, or both. It is represented by a shift code or HED Number.

### **Shortcut key**

Key that corresponds to a command name on a menu, such as CTRL+Z.

### **Shortcut menu**

A menu that appears when you right-click within The Solution Series 4. This menu contains context-sensitive commands and options that relate to the object (form, Navigator, and so on) on which you have clicked.

### **SIC**

Standard Industrial Classification.

### **Sick days**

The time off that an employee is allowed to take due to illness as a result of an employment contract or organizational policy.

### **SOAP**

Simple Object Access Protocol. This is an XML-based protocol that exchanges structured and typed information on the Web.

### **SOC**

Standard Occupational Classification.

### **Solution View**

An online utility that provides the tools for creating new forms, fields, and report programs without the direct use of Cyborg Scripting Language.

### **Source DL**

The node that owns the data being distributed. Depending on the rules established, the same DL can alternate from source to target.

### **Special assessment**

Extraordinary or temporary taxes, such as additional employer-paid or employee-paid contributions to state unemployment programs or to mandatory health insurance programs.

### **Spinbox**

A control on the interface composed of a text box and increment and decrement buttons that allow you to adjust a value from a limited range of possible values.

### **Spreadsheet application**

Software for recording ledger entries, creating worksheets, graphing data, and other accounting functions.

### **SQL**

Structured Query Language retrieves information from a relational database so it can populate Impromptu reports.

### **Standalone Time and Attendance**

Customers who are using Time and Attendance Administration but not Payroll Administration.

### **Standard Pay Processing and Reporting**

Pay run and reporting process that is executed in batch.

### **Static data**

Includes organization and employee information, such as name and salary.

### **Static SQL**

Data Definition Language (DDL) and Data Manipulation Language (DML) statements embedded in application programs.

### **Status bar**

The bar that appears at the bottom of The Solution Series window. The Status Bar displays useful information, such as your current session number, the currently displayed organization and employee, and so forth.

### **Statutory employee**

Any of the four categories of workers who are independent contractors under common law and are treated by statute as employees. These include: (1) a driver who distributes beverages (other than milk) or meat, vegetables, fruits, or bakery products; or who picks up or delivers laundry or dry cleaning, if the

driver is your agent or is paid by commission.

(2) certain types of full-time insurance sales reps

(3) an individual who works at home on materials supplied by you that must be returned to you

(4) certain full-time travelling or city salespeople.

Social Security and Medicare (FICA) taxes may or may not be withheld. Income taxes are not withheld from a statutory employee. A statutory employee will receive a W-2 with the 'Statutory Employee' box checked.

### Sub-schedule number

A two-digit numeric text box used to further identify a schedule table.

### Succession planning

Finding and developing employees for placement into identified key positions that are expected to become vacant sometime in the future.

### SUI

State Unemployment Insurance.

### Summary page

To help you see information at a glance, eCyborg Interactive Workforce uses summary pages. The summary page displays a short view of detailed information. For example, all your emergency contacts appear on a summary page. You delete the contact or proceed to the detail for the contact from the summary page.

### Summary plan

A customer-owned description of a benefits plan.

### Supplemental wages

Wages that are separate from regular earnings may be classified as supplemental wages and taxed using the default method. The default method means using a set percentage specified by the tax authority. Examples of such earnings are bonuses and commissions.

### Surplus

A 'surplus' is an exceeded complement position.

### Switches

Text or list box options that allow decisions to be recorded; in some cases, to be turned on or off.

### System administrator

An individual responsible for maintaining a multi-user computer system, including a local-area network (LAN). Typical duties include:

- Adding and configuring new workstations
- Setting up user accounts
- Installing system-wide software
- Performing procedures to prevent the spread of viruses
- Allocating mass storage space

### System Control Repository

This is the file that contains system definitions for The Solution Series, (FILE01). This was formerly known as the Control File.

### System Generator

A type of Report Generator that performs system functions, such as defining data elements and system messages.

### Table

Contains an organization's rules and policies and controls what actions take place at the employee level.

**Alternative definition:** means of displaying information in columns and rows.

### Table Definition Record

Table containing data about the Position Administration table records, including the location of keys to associated tables.

### Target DL

The node that receives the data being distributed. Depending on the rules established, the same DL can alternate from target to source.

### Task

The lowest level of organization on the Navigator or menu; generally equivalent to a form, checklist, or dialog.

**Alternate definition:** a function to be performed; objective

### Task icon

An icon denoting a task. Task icons describe the type of task, including Forms, Checklists, Dialogs and others.

**Tax authority**

A government agency to which an employer and employee has statutory tax obligations. The tax authorities for which you handle taxes exist at the federal, state/province, and local levels.

**Tax Authority File**

A supplied file that contains all the tax-specific information needed to calculate taxes for tax authorities. This includes wage-bracket tables for different marital statuses and information relating to allowances and standard deductions. The sources for the contents of this file are tax specifications published by the various tax authorities.

**Tax class**

Defines taxability for pre-tax deductions. These contributions can vary by tax authority. In the U.S.:

- \* Tax Class 1 defines taxability for 401(k) plans (maintained for all but local authorities)
- \* Tax Class 2 defines Section 125 plans (maintained for federal only)
- \* Tax Class 3 defines tax-sheltered annuities (maintained for federal only)
- \* Tax Class 4 - 9 fields are customer-defined

**Tax code**

The three-character to seven-character supplied reference code that identifies a tax and that serves as the link between the Tax Specification Record and the employee tax record.

**Tax specification**

Each tax authority publishes tax specification information that specifies how each tax must be administered. This information specifies how employers should calculate taxes and how taxes should be withheld from employees (if withholding applies). The tax specifications can be in the form of tax formulas and (or) tax tables.

**Tax Specification record**

A record on your Employee Database that contains the tax specifications for a tax. The record contains all the information, as obtained from the governmental authority, needed to calculate tax amounts for the tax. The record may contain more than one tax; for

example, US state Tax Specification records contain information for both state income tax and state unemployment insurance. Once a Tax Specification record is activated, tax specification information from the supplied tax files can be loaded onto the record on your Employee Database.

**Tax table**

A set of information required to calculate a tax, for a specific set of employee parameters. Tax tables are stored and maintained in Tax Specification records. A table typically includes wage and bracket information and data relating to allowances, such as personal exemptions and to standard deductions. There can be several tables relating to marital and resident status in a given Tax Specification record.

**Tax type**

Income tax, unemployment insurance (including special assessments), and disability insurance are examples of tax types. At the federal level, there are also entitlement taxes: Social Security (FICA-OASDI) and Medicare (FICA-HI). The types of taxes to collect and the methods of collecting them vary by tax authority.

**Tax-related Regulatory Bulletin**

A regulatory bulletin contains the updates to tax specifications, consisting of a bulletin document and a tax file that contains the updated tax specifications.

**Taxability**

The term refers to whether an hours, earnings, and deductions amount is to be included in taxable wages to be accumulated for a specific tax. If the hours, earnings, and deductions amount is excludable, then the amount is not included in taxable wages. If the hours, earnings, and deductions amount is taxable, then the amount is included in taxable wages. The term fully excludable or fully taxable implies that more than one type of tax is being referenced, for example, state income tax and state unemployment insurance in the US.

**Taxable wage base**

The taxable wage base represents the maximum amount of an employee's wages on which tax is levied and after which there is no liability. A wage base in the US typically is in effect for FICA, unemployment taxes, and disability.

### TDR

Table Definition Record.

### Template

A basis from which to create a custom item. For example, you can use an existing report as a template for your custom report.

### Temporary password

A set of alphanumeric characters used with a user ID to limit access to a software application. The system requires that users replace their temporary password with a user-defined password within a certain number of days.

### Termination

The activity of an employee no longer being employed by the organization.

### Test environment

A separate organization or system partition used only for testing.

### Text box

A control on the interface in which text can be entered and edited (formerly known as a field).

### Text qualifier

The character surrounding an item between delimiters. All values between the qualifier are data items and are not scanned for a delimiter. This allows a delimiter character, such as a comma, to be a valid data item. Example:

```
"item 1","item 2","item 3, 4 and 5"
```

This string contains three data items:

Item 1

Item 2

Item 3, 4 and 5

Although the third item contains a comma, it is ignored as a delimiter because it is between the text qualifier of speech/quotation marks (").

### Third Party Administrator

An organization outside of a company that administers employee benefit plans on the company's behalf.

### TIB-4

Technical Information Bulletin (Social Security Administration Publication 42-007). Also used to refer to the specifications and instructions for reporting to participating states through magnetic media.

### Time entry

The form in which you enter the hours worked for an employee. This was formerly known as a Time Card.

### Time entry extract file

A file of time entries external to Time and Attendance Administration that is used to feed to payroll.

### Time entry validation

The Time Entry Validation/Creation program identifies and assigns an activity, for example Clock In (1), to each clock transaction (ring) when performing the validation function. Each clock transaction must be assigned to an activity, in order for time entry hours to be calculated for an employee, for a particular shift. This program validates clock transactions (rings) and generates time entries.

### Timeout

The period of time that elapses before a user's eCyborg Interactive Workforce account becomes invalid because of inactivity.

### TLCN

Tape Library Control Number - a government-provided control number.

### ToolTip

Text that displays under a tool to describe its function when the mouse is held over the tool.

### Top-down appraisal

Appraisal made by a supervisor or manager of an employee's capabilities. Such an appraisal is generally based on the supervisor's or manager's day-to-day observation of an employee's work performance and will usually include an appraisal interview with the employee.

### TPA

Acronym for *Third Party Administrator* (see "Third Party Administrator" on page 196).

**Track**

One of the concentric magnetic rings that form separate data storage areas on an electronic medium.

**Trading Partner**

A Covered Entity with whom another Covered Entity transmits health information electronically through the use of an Electronic Data Interface. Trading partners must create an agreement as to the specific information that will be passed during their transactions. The Electronic Data Interface is standardized by the provisions set forth in the Health Insurance Portability and Accountability Act of 1996 (HIPAA). Also referred to as Business Partner.

**Trainer**

Trainers are set up on the Provider Index Form. They are instructional institutions, organizations or persons who are available to teach a training class.

**Trainer code**

The trainer code is a four-character value that represents a trainer. This value resides in Option List TR38.

**Training area**

The training area is recorded on the Class Schedule Form. It is typically defined as the section of the organization to which the training applies, such as manufacturing.

**Training class results**

These are the class details and absence information recorded on the Process Class Results form. Details recorded include the objectives met when taking a training class.

**Training class status**

The status value is updated and displayed on the Class Schedule Form. It tracks whether the training class is cancelled, full or available.

**Training course code**

The training course code is a six-character value that represents a training course. This value resides in Option List TR33 and is associated with a course title.

**Training plan**

A plan of training courses that an employee will attend in the future to achieve the necessary skills to perform a job.

**Training reason**

The reason for training is used to identify why a training request has been made. For example, the purpose of the training to act as a refresher, to acquire new skills, and so forth.

**Training request**

A training request is a request for an employee to attend a specific course or class. A formal request for training is not essential. This step could be omitted and the employee could be registered directly in the course of his or her choice.

**Transaction Set**

Refers to HIPAA 834 (Benefit Enrollment and Maintenance) and 820 (Group Premium Payment for Insurance Products) transmissions meaning the smallest meaningful set of information exchanged between *trading partners* (see "Trading Partner" on page 197). The transaction set consists of a transaction header segment, one or more data segments in a specified order, and a transaction trailer segment.

**Transfer**

Process of moving an employee from one organization to another organization, such as moving an applicant from the applicant organization to the active employee organization.

**Alternative definitions:**

1. To move data or files from one computer to another
2. Electronic Funds Transfer or direct deposit
3. Allows employees to authorize the organization they work for to automatically deposit all or part of net pay into one or more bank accounts.

**Trend analysis**

Reporting or statistics that indicate the rate of change in costs and other elements of a benefits plan.

**Trigger**

A set of conditions that must occur for an email or letter communication event to start. This can involve

the creation, deletion, or modification of forms or checklists within the system.

### **Tuition reimbursement**

Remuneration made to employees for tuition expenses.

### **Type of training request**

The type of training request indicated whether the employee was required to attend the training or whether he or she asked to attend the training.

### **UI**

Unemployment Insurance.

### **UI Number**

Unemployment Insurance Number (or SUI number, State Unemployment Insurance number); required by all states.

### **Unauthorized absence**

Absences that are generally not considered paid time away from regularly scheduled work.

### **Underlined text**

In browser applications, text that provides a link to another screen or page.

### **Unemployment insurance tax**

A tax required by some US states to be funded by employee-paid contributions to pay all or part of the cost of unemployment insurance coverage. On Payroll Administration, state unemployment insurance tax records are established as Type 2 taxes.

### **Unpaid absence**

Employee absence that will not be paid by the organization. A time entry will not be created for this absence.

### **Upward appraisal**

Appraisal that calls for evaluations by those who work under the direction of the employee being evaluated.

### **URL**

Acronym for uniform resource locator. A standard way of specifying the location of an object, typically a web page, on the Internet. URLs are the form of address used on the World-Wide Web. They are used in HTML documents to specify the target of a hyperlink that is often another HTML document (possibly stored on another computer).

### **User class**

Cognos Impromptu assigns security according to configured user profiles. These security profiles are configured by your Impromptu administrator.

### **User code**

A set of characters (up to eighteen alphanumeric characters) that, along with the password, identify the user to the system as a valid user when they log on.

The user code is case-sensitive (upper case, lower case) and must be entered using the correct case.

### **User defined password**

A set of alphanumeric characters created by users that allows them to view and update information in a software application.

### **User ID**

A set of characters that identify you to the software application. The application contains a list of authorized users by user ID. When you attempt to log on, the system checks the list of authorized users to determine whether you have authority to use the application.

### **User profile**

Used for security purposes to determine what you can and cannot do while you are using the system, and which parts of the system you can access. A user profile is created and maintained for you by a Security Officer. Each user of the system will have a user profile.

### **Vacancy**

An open position that needs to be filled, or an unfilled complement position

### **Vacation days**

The time off that an employee is entitled to as a result of an employment contract or due to length of service.

### **Validation**

The process where the Time Entry Validation program identifies and assigns an activity to a clock transaction (ring) when performing the validation function.

### **Variant forms**

Method of displaying country-specific variation of delivered forms.

**Waive**

The act of choosing not to enroll in an optional benefits plan.

**Warning**

An employee's schedule warning times.

**Warning time**

Used to set a period of time after which an employee will appear on the exception report for a particular activity. A Warning condition will allow the creation of a time entry. A Reject condition will not. This is part of the Time and Attendance Administration.

**Welfare benefit plan group**

First level of the logical organization of welfare benefit plans in eCyborg Interactive Workforce.

**Welfare benefit plan subgroup**

Second level of the logical organization of welfare benefit plans in eCyborg Interactive Workforce.

**Welfare plan**

Any insurance or other benefit plan that provides immediate benefits to a participant—for example, medical insurance.

**What-if mode**

Method for processing a report that allows viewing of information without updating of employee records.

**Window**

A standard Windows object that displays information. A window is a separately controllable area of the form that typically has a rectangular border.

**Wizard**

A form if user assistance that automates a task through a dialog with the user.

**Work area**

The Solution Series screen. It includes the menus, toolbars, Navigator, forms area, message area, and status bar.

**Work instructions**

Specific tasks to be completed during the migration of data and files from test to production.

**Work restrictions**

Restrictions that prevent an employee from participating in specific workplace functions.

**Worker's compensation**

Legislation in the US that provides compensation to employees who suffer work-related injuries.

**Workforce competency**

The capacity of the overall workforce to perform required functions and sets of activities.

**XHTML**

Extensible HyperText Markup Language, used by the help pages for eCyborg.

**XML**

Extensible Mark-up Language is a language for documents containing structured information.

**XML Schema Definition Language**

A language for describing and constraining the content of XML documents.

**XSLT**

Extensible Stylesheet Language Transformations. A language for transforming XML documents into other XML documents.

**Year End Master File**

P20OUT file from the final payroll run of the year



# Index

---

- . . . . .
- .EXE ..... 163
- .INI ..... 163
- I**
- 1ST (021) Set index to 1 ..... 128
- 3**
- 360-degree appraisal ..... 163
- 8**
- 820 (Premium Payment for Group Insurance Products) Transaction Set ..... 163
- 834 (Benefit Enrollment and Maintenance) Transaction Set ..... 163
- A**
- A1C (035) Add 1 to counter ..... 120
- About This Manual ..... 3
- Absence data ..... 163
- Absence point ..... 163
- Absence type ..... 163
- Accessing History and Labor ..... 85
- Account timeout ..... 163
- Accrual Method Code ..... 82
- Accumulator id ..... 163
- ACH ..... 163
- Acrobat ..... 163
- Action button ..... 163
- Activity ..... 164
- Activity code ..... 164
- Activity types ..... 164
- Actuarial valuation ..... 164
- ADA (054) Add days to date ..... 123
- ADD (032) Add to counter ..... 120
- Add/Subtract ..... 74
- Adding Report Generators ..... 55
- Adding, replacing, and deleting report generators ..... 55
- Additional Instructions ..... 135
- Adjustment ..... 164
- Administration home page ..... 164
- Administrative User ID ..... 164
- AE transaction ..... 59
- Agent types ..... 164
- Aggregate tax method ..... 164
- Alphabetic List of Instructions ..... 108
- AMO (086) Add months to date ..... 124
- Analyzing the results ..... 96
- Anniversary date ..... 164
- Annualization ..... 164
- Annualization factor ..... 164
- Annuitant ..... 165
- Annuity ..... 165
- AP (221) Account Payable amount (all taxability type code 16 earnings) ..... 136
- Appendices ..... 105
- Applicant ..... 165
- Applicant organization ..... 165
- Application data in the Sequential Master File ... 9
- Appraisal rating ..... 165
- Archive ..... 165
- Arithmetic operations ..... 73
- Arrears ..... 165
- ASCII ..... 165
- Ask Me wizard ..... 165
- As-of reporting ..... 165
- Audit record ..... 165
- Audit report ..... 165
- Audit trail ..... 165
- Authorized absence ..... 165
- Automatic Earnings/Deductions (HED Method Code) ..... 81
- Automatic plan ..... 165

Average deferral percentage.....	165	CAL (197) Call another generator .....	134
Average rating .....	165	Calculation option list .....	167
AX1 (024) Set all indexes to 1 .....	129	Candidate .....	167
<b>B</b>		Career planning .....	167
Back.....	165	Carrier record .....	167
Badge.....	165	Case-sensitive .....	167
Badge error .....	166	Catalog .....	167
Badge number .....	166	Category code .....	167
Banner .....	166	CBSVB .....	167
Batch .....	166	CBT.....	167
Batch control record .....	166	CCF (042) Convert counter.....	117
Batch layout facility .....	166	CE/H .....	167
Batch Master file (P20) .....	166	CFC (056) Convert to counter.....	117
Batch number .....	166	Change control facility.....	167
Batch processing .....	166	Check box .....	167
Batch transaction .....	166	Check digit.....	167
Benchmark job .....	166	Checklist .....	167
Beneficiary .....	166	Checklist item .....	167
Benefits Business Partner.....	166	Checklist item status .....	167
Benefits control number .....	166	Checklist margin .....	168
Benefits statement .....	166	Checklist wizard.....	168
BF1 (222) Browse FILE01. Find and read a record from FILE01 .....	136	Checkmark .....	168
Big option list.....	166	CHL (062) Chop left.....	119
Blocking factor.....	166	CHR (063) Chop right.....	120
BLZ (099) Blank leading zeros .....	119	Class.....	168
BO1 (174) Bump output position by 1 or more.....	125	Class evaluation results.....	168
BO2 (175) Bump output position by 2 .....	125	Cleared payment .....	168
BOP (055) Bump output position.....	123	Client data file.....	168
Bridge loan .....	166	Clock in and out .....	168
Browser .....	166	Clock transaction (ring) .....	168
BSC (065) If next record is the same Organization 1-2 .....	114	Clock transaction warning .....	168
BSE (066) If next record same employee.....	114	Closing costs .....	168
BSK (016) Build sort key .....	122	CLP .....	168
Budget plan year.....	166	CML (177) Change move length .....	126
Budget scenario .....	166	CNZ (076) If current HED not zeros .....	114
Budget setting.....	167	Codeset.....	168
<b>C</b>		Coefficient .....	168
CAD (048) Condense address .....	132	Combined extract reporting .....	47
Cafeteria plan .....	167	Combined register (2222) report.....	168
		Command button.....	168
		Common pay master .....	168
		Common tax organization.....	168
		Communication event .....	169

Compa ratio .....	169	Course .....	170
Company logical Master record .....	169	Course directory .....	170
Company validation table .....	169	Covered Entity .....	170
Compare counter fields.....	73	Covered Entity Group .....	170
Compare instructions.....	113	CPI .....	170
Compares fields/literals .....	64	Credited service.....	170
Competency.....	169	Crew .....	170
Complement limit.....	169	Crew code .....	170
Complement position.....	169	Crossfoot .....	171
Complement unit .....	169	Cross-reference keys .....	171
Compliance.....	169	Cross-references .....	4
Component .....	169	CSL .....	171
Component icon.....	169	CTG (234) complement to Gregorian date conversion.....	117
Component plan.....	169	CTR (034) Move counter to output.....	120
Condition .....	169	Cube (Power Play) .....	171
Considered earnings .....	169	Cumulative data .....	171
Considered earnings/hours (CE/H) accumulators .....	169	Cursor.....	171
Considered hours paid .....	169	Custom report format .....	171
Considered hours worked .....	169	Customer-defined.....	171
Consolidated reporting .....	169	CYB88X.....	171
Context-sensitive help .....	169	CYBMST .....	171
Contractual exchange rate .....	169	Cyborg Scripting Language .....	171
Contribution type.....	170		
Control 1-2 .....	170	<b>D</b>	
Control levels .....	170	D transaction .....	57
Control number.....	170	Data extract .....	171
Conventions used in this manual .....	4	Data Extract File Splitter (RSPLIT).....	171
Conversion.....	170	Data load .....	171
Conversion option .....	170	Data mapping .....	171
Converting D transactions to online form entries	57	Data mart.....	171
Co-ordinator .....	168	Data mart Extract File (FILE36) .....	171
Core plan .....	170	Data store .....	171
Cost categories .....	170	Database.....	171
Cost types .....	170	Date conversions .....	116
Costing .....	170	Debugging.....	95
Costing record .....	170	DED (070) If HED is deduction.....	115
Counter Compares .....	113	Deduct credits by plan.....	172
Counter instructions.....	72	Deduct credits by plan method.....	172
Counter manipulation instructions.....	120	Deduction .....	172
Counters.....	72	Deduction cycle.....	172
Counters, Arithmetic, and Conditional Branching .....	71	De-enrollment .....	171
		Default.....	172

Deferred compensation.....	172	Dynamic SQL .....	174
Deferred plan.....	172	<b>E</b>	
Deleting Report Generators.....	56	E\$ (047) Edit with \$.....	118
Delimiter .....	172	Earned income credit .....	174
DEL-PE .....	172	Earning.....	174
Density .....	172	Earnings category.....	174
Dependent .....	172	EBCDIC.....	174
Dependent GTL.....	172	ECF (096) Edit counter format field for plus or minus .....	117
Dependent number .....	172	eCyborg Interactive Workforce Home.....	174
Deposit advice.....	172	eCyborg Interactive Workforce Home page ....	174
Derived data .....	172	EDI.....	174
Detail page.....	172	Edit instructions .....	116
Dialog box.....	172	EDN (235) Edit Counter Format to -----99	118
Direct deposit .....	172	EDT (060) Edit counter to ----,---99.....	117
Disability .....	172	EEOC.....	174
Disability insurance tax .....	172	Effective date .....	174
Disciplinary action .....	173	Efficiency Considerations .....	47
Discretionary increase.....	173	Efficient generator instructions.....	48
Display .....	173	EFT .....	174
Display box .....	173	EIC.....	174
Display text boxes.....	173	EIN.....	174
Disposable income .....	173	EL .....	174
Distributed Access Log Table.....	173	Electronic documentation .....	174
Distributed location.....	173	Electronic Performance Support system .....	174
Distributed Location NODE Control Table....	173	ELZ (236) Edit Counter to 9(7)V99.....	119
Distributed Rules Table.....	173	Email.....	174
Distribution .....	173	Employee cancellation .....	174
Distribution rules.....	173	Employee Database.....	174
DIV (036) Divide counter .....	120	Employee Database record.....	174
DL .....	173	Employee logical Master record .....	175
Docked toolbar .....	173	Employee Number .....	175
Document type .....	173	Employee status .....	175
Double-click.....	173	Encoding .....	175
Drop-down list.....	173	English Language.....	175
Drop-down list box.....	173	Enhanced Pay Processing and Reporting.....	175
DSK (090) Dummy sort .....	125	Enrollment form.....	175
DSP08 .....	173	Entitlement accrual .....	175
DSRF02.....	173	Entity.....	175
DSRSET.....	173	Entry field .....	175
DSSRRL.....	173	Entry form.....	175
DSUSET.....	174	Environment.....	175
DTExy (232) Edit date .....	116		
DV2 (172) Divide with 2 decimal result .....	121		

EP (097) Edit packed (IBM users only) .....	119	Foreign currency .....	177
EPSS .....	175	Form .....	177
ERN (067) If HED is earnings .....	114	Form 300 .....	177
Establishment Reporting .....	175	Form 300A .....	177
Euro triangulation .....	175	Form 301 .....	177
Event .....	175	Form area .....	177
EX1 (224) Exchange FILE01. Swap FILE01 records .....	136	Form Builder .....	177
EXC (051) Save the index .....	129	Formal education .....	177
Exchange rate .....	175	Format transactions .....	27
Exchange rate type .....	175	Forward .....	177
Excused absence .....	175	FSA .....	177
Execution scripts .....	175	FST (084) Set employee tax to first state .....	130
Extract file .....	175	FTE .....	177
Extract program .....	176	FTP .....	177
Extract Record .....	41, 43	Full Time Equivalent .....	177
Extract Record Compares .....	113	Function keys .....	177
Extract record manipulation instructions .....	122	Funeral days .....	177
Extract Record Sort .....	41		
Extract record work area .....	44	<b>G</b>	
Extract record work area manipulation .....	44	Gap analysis .....	177
		Garnishment .....	178
<b>F</b>		General ledger .....	178
FC (073) Set header HED to category .....	130	General ledger interface .....	178
Federal Insurance Contributions Act .....	176	Generating multiple payment documents .....	90
FEIN .....	176	Generator and the COBOL Programs .....	20
FICA .....	176	Generator content .....	21
Field .....	176	Generator instructions .....	47
Field Number Table .....	18	Generator Instructions .....	107, 108
Filing entity .....	176	Generator instructions - Unconditional branching .....	45
Filter .....	176	Generator maintenance .....	54
Finished .....	176	Generator Maintenance .....	53
Flat rate tax .....	176	Glossary of terms .....	3
FLD (017) Move .....	122	Go to details .....	178
Flex credits .....	176	Graphical User Interface .....	178
Flex Master Plan .....	176	Grievance .....	178
Flex plan .....	176	Gross wages .....	178
Flexible Benefits Plan .....	176	Group box .....	178
Flexible Benefits Program .....	176	Group plan .....	178
Flexible Spending Arrangement .....	177	GRP (057) Move group .....	124
Floating toolbar .....	177	GTC (233) Gregorian to complement date conversion .....	116
FML (045) Edit name .....	119	GUI .....	178
Folder .....	177		

### **H**

H2 transaction, column 13, extract debugging record format.....	103
Handicap .....	178
Health and safety profile .....	178
HED.....	178
HED initialization exit point .....	82
HED processing exit point .....	83
Help.....	178
HIPAA.....	178
History processing.....	86
History record.....	178
Holiday days.....	178
Home location .....	178
Home page .....	179
How to get additional help .....	4
How to use this manual .....	3
HTML .....	179
HUP (140) Set H To-date up 1 .....	130

### **I**

I-beam .....	179
ICESA .....	179
IFF (085) Select employee if frequency .....	133
Import facility.....	179
Import record.....	179
Impromptu.....	179
Inactive plan .....	179
Inactive tax record.....	179
Incumbent.....	179
Index.....	4
Index loop examples.....	74
Index manipulation instructions .....	128
Inefficient generator instructions.....	47
Information-level security .....	179
Initial Administrator.....	179
Initial passwords.....	179
InitialAdmin .....	179
Inquiry form .....	179
Installing a Method Code .....	80
Instructional text.....	179
Instructions to access FILE01 .....	136
Interface File Output .....	88

Internal candidate.....	179
Internet .....	179
Intranet.....	179
Introduction.....	1, 64, 72
Introduction to Debugging Report Generators...	96
Introduction to Report Generator .....	5
Introduction to Sample Report Generator .....	138
Investment funds.....	180
IPEDS .....	180
IRP/BSS .....	180
IRS/MCC .....	180

### **J**

JDBC.....	180
JMNTRUN.....	180
Job.....	180
Job assignment.....	180
Job code .....	180
Job streams.....	180
Job type.....	180
JPAYRUN .....	180
JQTRRUN .....	180
JUP (145) Set J To-date up 1 .....	131
Jury duty .....	180
JXP5QTR.....	180

### **K**

KEYDEL .....	181
KEY-PE .....	180

### **L**

Label .....	181
Labor processing.....	86
Labor record.....	181
Large number.....	181
LDEFAULTS.....	181
Leave of absence.....	181
Link/jump.....	181
LIS file .....	181
LIT (018) Move a literal .....	123
Loading and selecting generators.....	54
Local currency .....	181
Local currency country .....	181

Log file ..... 181  
 Log off..... 181  
 Logic instructions ..... 45  
 Logical Employee Model ..... 181  
 LPI..... 181

**M**

Mailing address ..... 181  
 Maintenance payroll run..... 181  
 Major activity ..... 181  
 Mandatory field ..... 181  
 Manual payment..... 181  
 Map file ..... 181  
 Mass time entry creation ..... 181  
 Master File (0202) report..... 181  
 Master File print ..... 90  
 Matrix ID..... 182  
 MDY (008) Edit date to MM/DD/YY ..... 116  
 MEM (186) Reserve memory..... 126  
 Menu..... 182  
 Menu bar item ..... 182  
 Message area ..... 182  
 Method code..... 182  
 Method Codes..... 79  
 Method Codes general rules ..... 80  
 Midpoint..... 182  
 Minimart..... 182  
 Miscellaneous compares..... 114  
 Miscellaneous instructions ..... 131  
 MMREF..... 182  
 Model (PowerPlay)..... 182  
 Monetary perquisites ..... 182  
 Moving expenses..... 182  
 MPC (181) Multiply percent ..... 121  
 MPY (037) Multiply counter ..... 121  
 MSD (094) Move system date..... 125  
 MST (095) Move system time..... 125  
 MTO (188) Move to 'Other' record..... 126  
 MTW (201) Move data to a work field ..... 127  
 Multicurrency costing records ..... 182  
 Multi-master processing ..... 87  
 Multiple forms - example ..... 92  
 Multiple master..... 182

Multiply/Divide..... 74  
 MUN (064) Move unique number..... 124  
 MVW (195) Move Data to or from a Work Area  
 ..... 127  
 MZR (029) Move zero (counter format) ..... 123

**N**

NACHA ..... 182  
 Navigation bar..... 182  
 Navigator..... 182  
 NCES ..... 182  
 NCT (069) If earnings not combo or tips ..... 115  
 NET (058) Set HED to net pay ..... 130  
 Net credit method..... 183  
 New hire..... 183  
 New period pay run ..... 183  
 New user ..... 183  
 New User Home page ..... 183  
 NMD (071) If deduction not memo ..... 115  
 NME (068) If earnings not memo ..... 115  
 NNP (072) If ORGANIZATION 1-2 HED not net  
 pay ..... 115  
 Node..... 183  
 Node ID..... 183  
 Notes ..... 4  
 Notional value ..... 183  
 NPA (216) Net Pay Available..... 135  
 NSE (074) If HED not shift differential earnings  
 ..... 115  
 NTE (075) If HED not tips earnings ..... 115  
 Number registered..... 183  
 Numeric conversions..... 117  
 NXT (019) Set index up 1 ..... 128

**O**

Object..... 183  
 Object key ..... 183  
 Obsolete plan..... 183  
 Off cycle..... 183  
 Off-cycle pay run ..... 183  
 One-stop document ..... 183  
 Online..... 183  
 Open enrollment..... 183

Operating a Method Code .....	81	Pay schedule.....	185
Operating report generators in the Sequential Master File .....	9	Pay stub.....	185
Operator ID .....	183	Pay-for-performance matrix.....	185
Option.....	183	Payment history record .....	185
Option button.....	184	Payroll home location .....	185
Option list.....	184	Payroll Level 3.....	186
Organization .....	184	Payroll Level 4.....	186
Organization Level 3.....	184	Payroll Level 5.....	186
Organization Level 4.....	184	Payroll Level 6.....	186
Organization Level 5.....	184	Payroll Overview Diagram .....	6
Organization Level 6.....	184	Payroll period.....	186
Organization levels.....	184	Payroll Process Control.....	186
Organization Number .....	184	Payroll run.....	186
Organization Unit.....	184	PCL .....	186
Organization Validation table.....	184	PDF .....	186
Organization-specific tax setup.....	184	Peer-group appraisal .....	186
OSHA.....	184	Pending de-enrollment segment.....	186
Other considerations.....	48	Pending eligibility segment.....	186
Override file .....	184	Performance appraisal.....	186
OVT (218) Overtime Pay Amount (all category 01 earnings).....	135	Performance appraisal rating.....	186
<b>P</b>		Performance rating.....	186
P2EDIT .....	184	Performance-related pay .....	186
P4 transaction .....	61	Permanent Master Record.....	186
P4CALC.....	13, 185	Perquisites .....	187
P4CALC and P5PRNT.....	13	Personal days .....	187
P5PRNT .....	185	PEX (194) Exit from a Perform .....	112
Packaged reporting.....	185	PF4 (193) PERFORM paragraph 254 through PEX .....	112
Packaged reports .....	185	Phonetic keys .....	187
Paid absence .....	185	Pixel .....	187
Paragraph names .....	64	Plan deactivation .....	187
Parallel run .....	185	Plan ID .....	187
Parameter form.....	185	Plan shutdown.....	187
Passing Control .....	61	Plan year .....	187
Password .....	185	Policy tables .....	187
Password aging.....	185	Pop-up menu .....	187
Pay allocation .....	185	Portable document format.....	187
Pay document.....	185	Position .....	187
Pay extract.....	185	Position Administration Control Number .....	187
Pay frequency.....	185	Position complement.....	187
Pay merge.....	185	Position in range .....	187
		Posttax.....	187
		PowerPlay .....	187

Practice .....	49, 69, 78
Premium .....	187
Prenotification .....	187
Pretax .....	188
Primary account.....	188
Printing .....	89, 90
Process.....	188
Process bar.....	188
Process Monitor.....	188
Program .....	188
Prompt .....	188
Protected amount.....	188
Prototype HED .....	188
Provider .....	188
PRV (020) Set index down 1 .....	128
Push button.....	188
<b>Q</b>	
Qualifier.....	188
Quarter-end P20.....	188
Quarterly Processor .....	188
Quartile.....	188
Query alternate keys.....	189
Query primary keys .....	189
Quick Hire .....	189
<b>R</b>	
R0 format transaction (used by P4CALC).....	27
R1 format transaction (used in P5PRNT).....	31
R2 format transaction (used in P5PRNT).....	36
R3/R4 format transactions (used by P5PRNT)..	37
R5/R6 format transactions (used by P5PRNT)..	38
R7 format transaction (used by P4CALC).....	40
R7/S7 coding considerations .....	40
RAD (049) Restore address .....	133
Radio button .....	189
RCL (185) Recall .....	126
Recall.....	189
Reciprocal taxation.....	189
Reconciliation number.....	189
Record .....	189
Recruitment .....	189
Recycle file.....	88
Recycle File.....	189
REG (217) Regular Pay Amount (all category 05 and 15 earnings).....	135
Registration .....	189
Registration number.....	189
Regulatory bulletin.....	189
Rehire .....	189
Reimbursement account.....	189
Reinstatement.....	189
Reject time .....	189
Relocation .....	190
Remaining net pay.....	190
REP (059) Examine replacing.....	119
Replacing Report Generators .....	56
Replication .....	190
Replication Application .....	190
Replication Distribution .....	190
Replication Holding File.....	190
Replication Packet File .....	190
Report.....	190
Report Generator.....	190
Report Generator - where it fits in.....	7
Report Generator Debugging Aid .....	102
Report Generator Debugging Checklist .....	98
Report Group.....	190
Report Group Scheduler.....	190
Report parameters .....	190
Report Selection.....	57
Report Viewer .....	190
Report with totals .....	68
Reporting Administration.....	190
Requisition .....	190
Requisition candidate .....	190
Requisition limit.....	190
Requisition unit.....	190
RET (198) Return to calling generator.....	134
Retirement.....	191
Return.....	191
Review process .....	191
RF1 (223) Read FILE01. Read the next FILE01 record.....	136
RM (192) Reverse Move.....	127
Roll-up reporting.....	191

Rotation pattern .....	191	Sequential Master File (P20) Diagram.....	8
RST (030) Reset last move.....	123	Sequential Master File processing with P4CALC and P5PRNT.....	16
Running the test.....	96	Service interruption.....	192
<b>S</b>		Service method.....	192
S02 (093) Set to counter 2.....	121	Session .....	192
S1 sort transaction (used by P5PRNT).....	21	SET (401) Set Current Counter Number.....	121
S7 coding considerations.....	25	Setting up the test.....	96
S7 sort transaction (used by P4CALC).....	25	SFC (176) Sort FORMS CODE.....	126
Safety standards.....	191	SHC (141) Set H To-Date to current.....	130
Salary budget record.....	191	Shift.....	192
Salary grade.....	191	Shift Differential/ Premiums .....	193
Salary grade range.....	191	Shift premium .....	193
Salary plan.....	191	SHL (082) Set header and employee HED to last earnings.....	130
Salary plan year .....	191	SHM (142) Set H To-date to month-to-date ...	130
Salary range.....	191	Shortcut key .....	193
Salary review.....	191	Shortcut menu .....	193
Salary review authorization form .....	191	SHQ (143) Set H To-date to quarter-to-date ...	130
Sample Field Number Table.....	19, 146	SHY (144) Set H To-date to year-to-date .....	130
Sample Field Table.....	145	SIC .....	193
Sample Report Generator .....	137, 138	Sick days .....	193
SAT file.....	191	SJC (146) Set J To-date to current .....	131
Save Changes .....	191	SJM (147) Set J To-date to month-to-date .....	131
SBL (083) Set header HED to last earnings....	130	SJQ (148) Set J To-date to quarter-to-date.....	131
Schedule Activities table .....	191	SJY (149) Set J To-date to year-to-date .....	131
Schedule assignments.....	191	SLC (092) Set to next lower number counter ..	121
Schedule error .....	191	SNC (091) Set to next higher numbered counter	121
Schedule Master table .....	191	SOAP .....	193
Schedule number .....	192	SOC.....	193
Screen .....	192	Solution View .....	193
Scroll bar .....	192	Sort 5G.....	81
SDD (087) Time span between dates .....	125	Sort key .....	42
Search argument.....	192	Sort transactions.....	21
Search type .....	192	Source DL .....	193
Secondary account(s) .....	192	SPC (046) Blank extract record .....	123
Security Officer .....	192	Special assessment .....	193
Segment index .....	66	Special usage of SEL for company/organization level HED and TAX segments .....	67
SEIN.....	192	Spinbox .....	193
SEL (022) Set immediate .....	128	SPL (219) Special Pay amount (all categories except 01, 05, and 15).....	135
Selectable File08 Display .....	192	Spreadsheet application.....	193
Self-adjusting taxes .....	192		
Sequential Master File.....	192		
Sequential Master File - its story.....	9		

SQL .....	193	Taxable wage base .....	195
Standalone Time and Attendance .....	193	Tax-related Regulatory Bulletin .....	195
Standard Pay Processing and Reporting .....	193	TDR.....	196
Static data .....	193	Template.....	196
Static SQL .....	193	Temporary password.....	196
Status bar .....	193	Termination.....	196
Statutory employee.....	193	Test environment.....	196
STO (184) Store .....	126	Testing/trace phase.....	102
SUB (033) Subtract from counter.....	120	Text box .....	196
Subroutines.....	47	Text edits.....	119
Sub-schedule number .....	194	Text qualifier.....	196
Succession planning .....	194	The Generator and P4CALC .....	20
Suggestions and feedback.....	4	Third Party Administrator .....	196
SUI .....	194	TIB-4.....	196
Summary page.....	194	Time entry .....	196
Summary plan.....	194	Time entry extract file.....	196
Supplemental wages .....	194	Time Entry Method Code.....	81
Surplus.....	194	Time entry processing exit point.....	83
Switches.....	194	Time entry validation .....	196
System administrator.....	194	Timeout .....	196
System Control Repository.....	194	TLCN .....	196
System Flow Diagram- the road from P20 to outputs.....	12	TNZ (077) If tax amount not zero.....	114
System Generator .....	194	To-date index .....	65
System initialization generators in the Sequential Master File .....	9	ToolTip.....	196
<b>T</b>		Top-down appraisal.....	196
Table.....	194	TPA .....	196
Table Definition Record .....	194	Track .....	197
Table of contents .....	3	Trading Partner.....	197
Target DL .....	194	Trainer .....	197
Task .....	194	Trainer code .....	197
Task icon .....	194	Training area .....	197
Tax authority .....	195	Training class results.....	197
Tax Authority File .....	195	Training class status .....	197
Tax class.....	195	Training course code.....	197
Tax code .....	195	Training plan .....	197
Tax specification .....	195	Training reason.....	197
Tax Specification record.....	195	Training request .....	197
Tax table.....	195	Transaction Set.....	197
Tax type.....	195	Transfer .....	197
Taxability.....	195	TRC (098) Trace .....	131
		Trend analysis .....	197
		Trigger.....	197
		Tuition reimbursement .....	198

Type of training request ..... 198

### U

UI ..... 198

UI Number ..... 198

Unauthorized absence..... 198

Unconditional branching ..... 112

Underlined text..... 198

Unemployment insurance tax..... 198

Unpaid absence ..... 198

Upward appraisal..... 198

URL..... 198

User class ..... 198

User code..... 198

User defined password ..... 198

User exit points..... 82

User ID..... 198

User profile..... 198

Using Report Generator..... 51

### V

VAC (229) Vacation Pay Amount (all category 02  
and 03 earnings)..... 136

Vacancy..... 198

Vacation days ..... 198

Validation..... 198

Variant forms..... 198

### W

Waive ..... 199

Warning..... 199

Warning time..... 199

WAT (205) Write audit trail..... 134

WCR (006) Write payments..... 131

Welfare benefit plan group..... 199

Welfare benefit plan subgroup ..... 199

Welfare plan ..... 199

What-if mode..... 199

Window ..... 199

Wizard..... 199

Work and save indexes..... 67

Work area..... 199

Work instructions ..... 199

Work restrictions.....199

Worker's compensation.....199

Workforce competency .....199

Writing a Method Code.....80

Writing Logic.....63

WRR (088) Write to recycled file.....131

WRT (005) Write to extract file.....131

### X

XHTML .....199

XIT (007) Exit report .....132

XML.....199

XML Schema Definition Language .....199

XS (015) Exit sort .....132

XSLT .....199

XTD (023) Set to-date.....129

### Y

Year End Master File .....199

YMD (216) complement to year month day date  
conversion.....116

### Z

Z29 (182) Zero counters 02 through 09 .....121

ZAC (050) Zero all counters .....121

ZC (031) Zero counter .....120

ZS2 (043) Edit 2 decimals .....118

ZS4 (044) Edit 4 decimals .....118

ZSC (061) Edit counter to *ZZZZZZZVZZ* .....118

# Hewitt

---

## **eCyborg Collaborative Platform 5.1**

**3.1**

Document Issue Status: 4th Release  
Document Issue Level: 3.1  
Document Issue Date: April 2004  
Software Version: 5.1

### **Copyright Notice**

Copyright© 2004 by Hewitt Associates LLC. All rights reserved.

### **Trademarks**

Cyborg Systems® and eCyborg® and The Solution Series®  
are registered trademarks of Hewitt Associates LLC.

The ASP Solution™ is a trademark of Hewitt Associates LLC.

Other third-party trademarks, service marks, logos, and tradenames that may appear,  
but are not specifically mentioned, are the property of their respective owners.



Hewitt Associates LLC, Suite 1700, 120 South Riverside Plaza, Chicago, Illinois 60606-3911

# Contents

Part 1	1
<hr/>	
Part 1 Introduction	1
<hr/>	
Chapter 1	3
<hr/>	
<b>Introduction</b> .....	<b>3</b>
Welcome .....	4
Audience.....	4
How to get additional help.....	5
Suggestions and feedback.....	5
<hr/>	
Chapter 2	7
<hr/>	
<b>Working with the Collaborative Platform</b> .....	<b>7</b>
eCyborg Collaborative Platform .....	8
SOAP and XML .....	9
What is a schema? .....	9
Authentication.....	9
XML and the Collaborative Platform .....	10
Data mapping.....	10
XML document transformation .....	13
View definition file .....	14
Unsupported forms .....	14
Request process.....	15
Receiving a request.....	15
Test harness.....	16
Read side functionality .....	16
Write side functionality .....	17

**Chapter 3** **19**

---

<b>Read functionality</b> .....	<b>19</b>
Introduction.....	20
Data sources.....	21
Default.xml .....	21
Table definition .....	21
Validate the SQL statement .....	25
Starting the SQL tool.....	25
Command line .....	26
Interactive mode .....	27
SQL-like syntax.....	27
Interactive usage.....	28
Non-interactive usage .....	29
Query map file .....	31
Test the Read functionality .....	33

**Chapter 4** **35**

---

<b>Write functionality</b> .....	<b>35</b>
Structure of the mapping file .....	36
Database information.....	36
Adding Solution Series records .....	37
Modifying Solution Series records .....	38
Structure of the incoming XML document.....	39
Transforming an incoming XML document .....	40
XSLT applied .....	40
Defining a Web Service View .....	41
Invoking the Web Service.....	42
Processing the request .....	42
Error content defined.....	44
Test the Write functionality .....	45

---

<b>Part 2</b>	<b>47</b>
---------------	-----------

---

<b>Part 2 Creating Collaborative Schema</b>	<b>47</b>
---	-----------

---

<b>Chapter 5</b>	<b>49</b>
------------------	-----------

---

<b>Creating a schema - Read side .....</b>	<b>49</b>
Introduction.....	50
Identifying the data source .....	51
Modifying the table definition file .....	52
Starting the eCyborg SQL tool.....	53
Testing the SQL statement.....	54
Adding field elements to the table definition file.....	55
Creating the map file.....	57
Creating a stylesheet .....	60
Editing ecp-views.xml .....	63
Performing a Read side test .....	64

<b>Chapter 6</b>	<b>65</b>
------------------	-----------

---

<b>Creating a schema - Write side .....</b>	<b>65</b>
Introduction.....	66
Identifying the data source .....	67
Modifying the table definition file .....	68
Starting the eCyborg SQL tool.....	70
Testing the SQL statement.....	71
Adding field elements to the table definition file.....	72
Creating the map file.....	74
Editing ecp-views.xml .....	79
Performing a Write side test.....	80

<b>Part 3</b>	<b>83</b>
<hr/>	
<b>Part 3 Appendices</b>	<b>83</b>
<hr/>	
<b>Appendix A</b>	<b>85</b>
<hr/>	
<b>Collaborative Platform Included Solution Series Forms.....</b>	<b>85</b>
Solution Series forms.....	86
<hr/>	
<b>Appendix B</b>	<b>87</b>
<hr/>	
<b>Error handling and codes.....</b>	<b>87</b>
Error handling.....	88
Codes and explanation.....	88
<hr/>	
<b>Appendix C</b>	<b>91</b>
<hr/>	
<b>eCyborg SQL Tool Syntax.....</b>	<b>91</b>
SELECT statement.....	92
INSERT statement.....	93
UPDATE statement.....	94
GROUP BY clause.....	95
ORDER BY clause.....	96
TOP keyword.....	97
CONTAINS function.....	98
String function.....	99
Date function.....	100
WHERE clause.....	101
<hr/>	
<b>Glossary of Terms</b>	<b>105</b>
<hr/>	
<b>Index</b>	<b>107</b>
<hr/>	

PART 1

# Introduction

---

## In This Section

Introduction .....	3
Working with the Collaborative Platform.....	7
Read functionality.....	19
Write functionality.....	35

CHAPTER 1

**Introduction**

---

**In This Chapter**

Welcome.....4

## Welcome

This document provides information about the eCyborg Collaborative Platform. The following chapters provide a conceptual explanation of the eCyborg suite of products, background on the inner workings of the Collaborative Platform and how to use it, and reference material.

## Audience

The primary audience for the eCyborg Collaborative Platform document has a technical background. However, the document covers the following different areas of interest:

- overview of eCyborg concepts and relationships
- background on XML and collaborative schema
- details about collaborative functionality

The following tables list the different sections and the content for each. The Collaborative Platform document contains six chapters and three appendices. The six chapters are as follows with their content listings:

<b>Chapter</b>	<b>Content</b>
Introduction	Overview of the document and general eCyborg concepts
Working with the Collaborative Platform	Background on the platform and a discussion of its components
Read functionality	Information about data sources, SQL commands, mapping files, and testing.
Write functionality	Information about data sources, SQL commands, mapping files and testing.
Creating a Schema - Read	Tasks for preparing your files and using the 'read' functionality
Creating a Schema - Write	Tasks for preparing your files and using the 'write' functionality

The three appendices are as follows with their content listings:

<b>Appendix</b>	<b>Content</b>
Included Solution Series forms	List of Solution Series forms
Error handling and codes	List of error codes and explanations
Collaborative Platform SQL tool syntax	Listing of beyond-the-basic functions for the Collaborative Platform SQL tool

## **How to get additional help**

If you need additional technical assistance or have questions about training, contact Customer Support. Additionally, you can visit our web site [www.hewitt.com/ecyborg](http://www.hewitt.com/ecyborg) for the latest schedule of available courses and course descriptions.

## **Suggestions and feedback**

We value your feedback on the performance support materials. Please forward any comments about this document to Customer Support.

CHAPTER 2

## Working with the Collaborative Platform

---

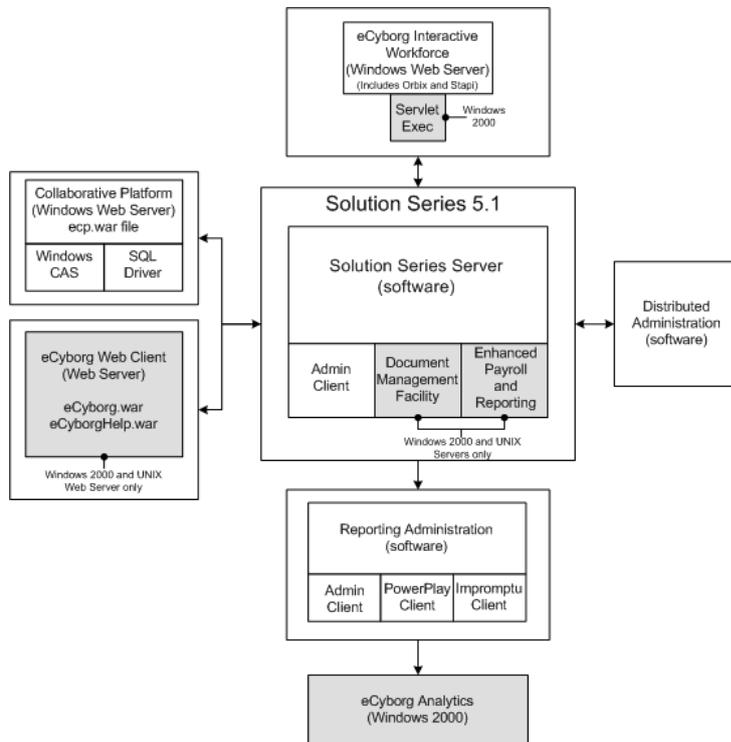
### In This Chapter

eCyborg Collaborative Platform .....	8
XML and the Collaborative Platform .....	10
Request process .....	15
Test harness .....	16

# eCyborg Collaborative Platform

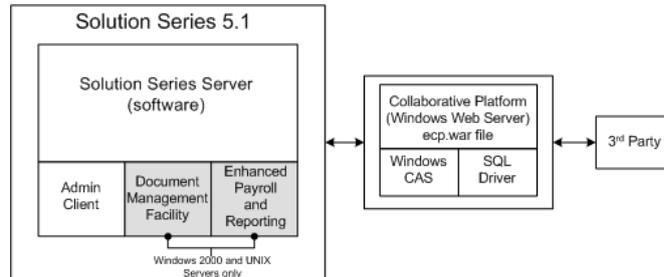
The eCyborg Collaborative Platform is a web application. It provides standards-based access to Solution Series data through a Simple Object Access Protocol (SOAP) interface using XML data schemas. It is a central component of the eCyborg application and is one of the most important interfaces to access employee data.

The following diagram presents the conceptual relationship of eCyborg 5.1 products and the role that the Collaborative Platform plays in the relationship:



Although eCyborg offers extensive functionality for employees and Human Resources departments, the Collaborative Platform provides a standard access mechanism for outside applications needing employee data.

The Collaborative Platform allows other (3rd party) applications to interface real-time with eCyborg.



## SOAP and XML

The Simple Object Access Protocol (SOAP) is an XML based protocol that lets applications exchange information over HTTP. SOAP provides a way to communicate between applications running on different operating systems using Internet browsers and servers.

### HR-XML consortium

The HR-XML Consortium is an organization dedicated to the development and promotion of a standard suite of XML specifications to enable e-business and the automation of human resources-related data exchanges.

By developing and publishing open data exchange standards based on Extensible Markup Language (XML), the Consortium provides the means for any company to transact with other companies without having to establish, engineer, and implement many separate interchange mechanisms.

## What is a schema?

A schema is a flexible communication format used by the Collaborative Platform. A schema is written in XML and describes the structure of any XML document. When data is sent from a third party to Solution Series, a schema provides a common language across the web. Schemas are derived from 'standard types' so they can be reused and referenced from the same document.

## Authentication

Using Solution Series security, you authenticate client applications in the eCyborg system by assigning them a user code and password. Because the data transferred between eCyborg and client applications outside the firewall is not encrypted, you should use Secure Socket Layer (SSL) technology.

## XML and the Collaborative Platform

The current release of the Collaborative Platform produces XML documents that conform to two HR-XML schemas:

- PersonName
- PostalAddress

PersonName and PostalAddress extract mapped data from the Employee Information form (EF-SCR). We deliver mapping definitions that allow you to create XML documents that conform to the HR-XML Consortium standards. The Collaborative Platform also produces documents that conform to the following schema for which HR-XML standards do not currently exist:

- Contact

The purpose of the 'Contact' schema is to demonstrate the Write functionality of the web collaborative platform. The Contact schema maps data from the (16-SCR) Emergency Contacts form.

### Data mapping

The Collaborative Platform uses a map file to define the retrieved data and to create an XML file. It extracts data and formats it according to the map file. The map file determines what the data is, the name and structure of the new file, and where to store the data.

#### Structure of the mapping file

The mapping file is an XML document containing the following information:

Elements	Definitions
Database	Defines the Collaborative Platform SQL driver and URL used to connect to the Solution Series data store
Query	Contains an SQL statement specifying the columns the mapping file retrieves
Root	Specifies the name of the destination root element and the name of the elements that represent either Solution Series records or database rows
Insert	Allows the SQL driver to add a new record to a table
Update	Changes one or more fields in one or more rows

For example, if the 'LastName' element has a 'code' attribute that represents the 'name code' field from the employee record and the employee's LAST\_NAME as the content, the map file would represent it in the following way:

```
<element name="LastName">  
  <content name="code">NAME_CODE</content>  
  <content>LAST_NAME</content>  
</element>
```

## Mapping file example

The following is an example of a mapping file used to extract basic employee information from The Solution Series. The Collaborative Platform creates an XML document where the root element (the overall structure of the generated XML document) is called 'PersonInfo' and the data for each record appears within a 'Person' element.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/cyb51run"
    driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>

  <query sql="SELECT * FROM EMP_BASIC_DETAILS WHERE COMPANY_NUMBER = ? AND
    EMPLOYEE_NUMBER = ? AND NAME_CODE = 1">
    <parameters>
      <parameter ordinal="1">
        <name>COMPANY_NUMBER</name>
        <type>STRING</type>
      </parameter>
      <parameter ordinal="2">
        <name>EMPLOYEE_NUMBER</name>
        <type>STRING</type>
      </parameter>
    </parameters>
  </query>

  <root name="PersonInfo" rowName="Person">
    <element name="FirstName">
      <content>FIRST_NAME</content>
    </element>
    <element name="LastName">
      <content>LAST_NAME</content>
    </element>
    <element name="MiddleName">
      <content>MIDDLE_NAME</content>
    </element>
    <element name="Prefix">
      <content>NAME_PREFIX</content>
    </element>
    <element name="Suffix">
```

In the example, the word (*element*) is the name of the element. The word (*name*) is an attribute called 'name'. The word (*FirstName*) is the value of that attribute.

According to XML standards, an element name and an attribute name must begin with a letter and can be followed by any combination of letters, digits, or any of the following characters:

Character	Definition
.	Period or full stop
-	hyphen
_	underscore
:	colon

Attribute values (*FirstName*) can contain any character, with the exceptions detailed below.

### Exception 1

In an XML file, the (&) is represented by the following escape sequence:

- &amp;

Also, the less than symbol (<) is represented by the following escape sequence:

- &lt;

For example, if the value of a Solution Series field called 'EMP<SPOUSE&CHILD' is required in a content element, it would be expressed in the following way:

```
<element name='SpouseAnd Child'>
  <content>EMP&lt;SPOUSE&amp;CHILD</content>
</element>
```

In the SQL statement contained in the mapping file, the same field should appear in the following way:

```
<query sql="SELECT EMP_FIRST_NAME, EMP&lt;SPOUSE&amp;CHILD FROM
EMP_BASIC_DETAILS WHERE EMPLOYEE_NUMBER =?"/>
```

### Exception 2

Element names should not contain the 'at' symbol (@). However, the Collaborative Platform SQL driver may in some circumstances generate field names that contain the '@' symbol. You can rename the field in the table definition file or in the data extraction map. For example, to rename a field called 'REQUISITION\_NUMBER@1' in the data extraction map, enter the following:

```
<element name="REQUISTIONNUMBER1">
  <content>REQUISITION_NUMBER@1</content>
</element>
```

Because the value of the name attribute does not have to be the exact name of the Solution Series field, the following example is also correct:

```
<element name="WhateverYouLike">
  <content>REQUISITION_NUMBER@1</content>
</element>
```

The examples also show how Solution Series field names or database column names are redefined in the XML output file. Redefining names is useful when the XML document returned to the SOAP client application must conform to a particular XML schema.

### Mapping file output example

The following is an example of the temporary XML document produced from the preceding mapping file:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <PersonInfo>
    <Person>
      <FirstName>Steve</FirstName>
      <LastName>Austin</LastName>
      <MiddleName>Ian</MiddleName>
      <Prefix>Mr</Prefix>
      <Address1>2314 W MILWAUKEE AV</Address1>
      <Address2>APT 8</Address2>
      <City>CHICAGO</City>
      <State>IL</State>
      <Zip>60614</Zip>
      <Country>US</Country>
    </Person>
  </PersonInfo>
```

## XML document transformation

The Collaborative Platform uses the SQL driver to connect to a Solution Series data source to generate an SQL Result Set. A Result Set is a tabular data structure in which each record corresponds to a row and each field corresponds to a column. The mapping file allows you to specify which fields appear in the generated XML document, but the structure of the document is tied to the two-dimensional structure of an SQL Result Set.

For maximum flexibility, the Collaborative Platform allows you to specify an Extensible Stylesheet Language Transformations (XSLT) stylesheet. The stylesheet is applied to the basic generated document, producing a new document that is returned to the client application. However, the use of a stylesheet is optional. If a view definition does not specify that a transformation be applied, the Collaborative Platform returns the basic generated document to the client application.

### Using XSLT

By using XSLT, a single mapping file can be accessed to produce XML documents that conform to different schema. For example, the two HR-XML schema: PersonName and PostalAddress. The data required to generate an XML document conforming to either schema is in a single Solution Series table:

- EMP\_BASIC\_DETAILS

The Collaborative Platform includes a mapping file that retrieves all the columns from the EMP\_BASIC\_DETAILS table. It also includes two XSLT stylesheets:

- PersonName.xsl
- PostalAddress.xsl

Applying the PersonName.xsl to the basic XML document generated by the Collaborative Platform produces a document that conforms to the HR-XML PersonName schema. Applying the PostalAddress.xsl to the generated document produces a document that conforms to the HR-XML PostalAddress schema.

### View definition file

The Collaborative Platform uses a view definition file to determine which views are available to client applications. Each entry in the view definition file provides the name of a view, the mapping file, and an XSLT stylesheet. The stylesheet produces the file that returns to the client application.

The Collaborative Platform reads the file each time it receives a request so views can be added or removed while the application is running. The following example is a view definition that defines views for the two HR-XML views: PersonName and PostalAddress.

```
<ecp:views xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">
  <ecp:view name="PersonName">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PersonName.xsl</ecp:xsl>
    <ecp:allow-write value='false'>
      <ecp:transform/>
    </ecp:allow-write>
  </ecp:view>
  <ecp:view name="PostalAddress">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PostalAddress.xsl</ecp:xsl>
    <ecp:allow-write value='false' />
  </ecp:view>
  <ecp:view name="Contact">
    <ecp:map>ContactMap.xml</ecp:map>
    <ecp:xsl></ecp:xsl>
    <ecp:allow-write value='true'>
      <ecp:transform/>
    </ecp:allow-write>
  </ecp:view>
</ecp:views>
```

### Unsupported forms

The following are types of forms that the Collaborative Platform does not support:

- Forms that are entry only (for example, TC1SCR)
- Forms that have parameter sections or pages (for example, PROFIL)
- Wizard-type forms that take you through a multi-step process (for example, 55FSCR)
- Multi-paneled forms (position management forms)

Once you determine if you can access your selected form data using the SQL tool, you can add the form to the table definition file.

## Request process

In the Collaborative Platform, business logic is separate from the content of the inbound XML documents. When the platform accepts an inbound XML document, the document contains only data and no explicit binding to the business logic that produces the desired response.

The Collaborative Platform applies its business logic to the inbound XML document and the document content determines the processing workflow. An inbound request contains the Solution Series user code and password of the principal (name of the view to process) and a set of applied constraints to determine the data to be retrieved. The following example displays a request for the PersonName view:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
  <ecp:ViewName
xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">PersonName</ecp:ViewName>
  <ecp:UserCode xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">****</ecp:UserCode>
  <ecp>Password xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">****</ecp>Password>
</soap-env:Header>
<soap-env:Body>
  <ecp:constraint xmlns:ecp="http://www.hewitt.com/eCyborg/ecp"
name="COMPANY_NUMBER">
    999999
  </ecp:constraint>
  <ecp:constraint xmlns:ecp="http://www.hewitt.com/eCyborg/ecp"
name="EMPLOYEE_NUMBER">
    1001
  </ecp:constraint>
</soap-env:Body>
</soap-env:Envelope>
```

*Note:* In the example, the '\*\*\*\*' UserCode and '\*\*\*\*' Password are those assigned to the client (requesting) application.

## Receiving a request

Upon receiving a request, the Collaborative Platform reads the view definition file and produces a PersonName response using the instructions found in the mapping document:

- PersonMap.xml

The Collaborative Platform combines the SQL statement in PersonMap.xml with constraints (the 'where' condition) from the request to produce the following query:

```
Select * from EMP_BASIC_DETAILS where COMPANY_NAME = '999999' and
EMPLOYEE_NUMBER = '1001'
```

The query executes and the SQL results create a temporary XML document defined by the mapping file.

## Test harness

To test the connectivity of your system, 'read' and 'write' test harness web pages are included with the ecp.war file. If you installed the Collaborative Platform on a server, you can access the test pages using the following URLs:

- Test read: `http://<host:port>/ecp/read.htm`
- Test write: `http://<host:port>/ecp/write.htm`

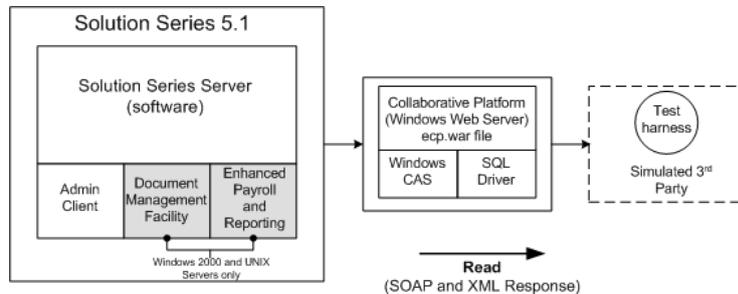
The <host:port> is the host and port number of the web server hosting the Collaborative Platform. The following example displays a URL configured for localhost access on port 8080:

- `http://localhost:8080/ecp/read.htm`

The following 'read' and 'write' sections explain the corresponding functionality for testing the web collaboration service. The test harness is a simulated third party, sending and receiving test data to confirm connectivity.

### Read side functionality

To test for connectivity, the 'read' side test allows the simulated third party a way to submit a request for data from the Solution Series application through the Collaborative Platform.

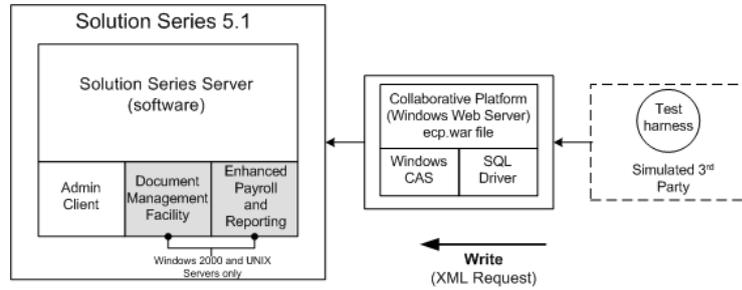


The XML response from Solution Series, returned in a SOAP envelope and viewed in a browser, contains either the requested data or an error message. The body of the example message contains the extracted data:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header/>
  <soap-env:Body>
    <PersonName>
      <GivenName>JUNE</GivenName>
      <FamilyName>MEYER</FamilyName>
      <Affix type="FormOfAddress">Ms</Affix>
    </PersonName>
  </soap-env:Body>
</soap-env:Envelope>
```

## Write side functionality

The 'write' side test allows the simulated third party to send data (an update) as an XML file to the Solution Series application through the Collaborative Platform. If the "write" operation is successful, Solution Series sends a SOAP response detailing how many and which rows of data were affected.



CHAPTER 3

# Read functionality

---

## In This Chapter

Introduction .....	20
Data sources.....	21
Validate the SQL statement .....	25
Query map file.....	31
Test the Read functionality .....	33

## Introduction

After installing the Collaborative Platform, you can produce XML documents that conform to two HR-XML schemas:

- PersonName
- PostalAddress

The schema extract the data contained in these XML documents from the Employee Information form (EF-SCR), which represents a small sample of the data captured by the Solution Series. To tap into Solution Series data in response to requests from the Collaborative Platform, you can create additional schemas.

## Data sources

You must create a schema when an application needs Solution Series data. When creating schema, the first step is to find out on which Solution Series form(s) the data exists. For example, a schema may only map to a single SQL table which means a single Solution Series form.

If the schema uses data from more than one form, there are two options. First, write a custom CSL form that provides an inquiry-only composite view of the required data. Second, create more than one schema (one for each form) and use multiple schemas in the client (requesting) application.

After you have identified the form (or forms), the next step is to test the schema with the eCyborg SQL tool. The testing step is important because you cannot access all Solution Series forms with SQL.

## Default.xml

The 'Default.xml' file is an XML table definition file. The definition file contains some of the most commonly used Solution Series forms that are available to the eCyborg SQL driver as database tables. The SQL driver uses an environment specific definition file. If an environment definition file is not defined, the SQL driver uses the 'Default.xml'. You should review the contents of the XML table definition file to determine if the form/table you need is already included.

## Table definition

A table definition file without a table or form looks like the following example:

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-tables>
</jdbc-tables>
```

The example above identifies the XML declaration (the first line) and the jdbc-tables element beginning and ending tags. Tables are children of the <jdbc-tables> tag. At a minimum, the table element must contain a form and table element and the form element must have a valid Solution Series form name. The following example shows the Default.xml file with the Solution Series EF-SCR form added and a table name:

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-tables>
  <table>
    <screen>EF-SCR</screen>
    <name>EMP_BASIC_DETAILS</name>
  </table>
</jdbc-tables>
```

A single table called EMP\_BASIC\_DETAILS that reads its data from EF-SCR is defined. You can access the table with an SQL query.

### Field elements

In the preceding example, all fields on the EF-SCR are read if an SQL statement is executed. Some Solution Series forms like the EF-SCR are multi-paneled. Multi-paneled means that the displayed form may present additional information. By default, if you have not defined fields in your table entry within the default.xml when you use a multi-paneled form as a table, a composite of all fields that appear on the form are returned as columns in the SQL result set.

By adding field elements to the table element of the default.xml, only those fields listed can be extracted with an SQL statement. Trying to access fields that are not defined will result in an SQL error. The following example shows some of the fields from the EF-SCR added to the table element beneath the name of the database table: EMP\_BASIC\_DETAILS.

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-tables>
  <table>
    <screen>EF-SCR</screen>
    <name>EMP_BASIC_DETAILS</name>
    <field>
      <name>EMP_NAME_CODE</name>
      <alias>NAME_CODE</alias>
    </field>
    <field>
      <name>EMP_NAME_PREFIX</name>
      <alias>NAME_PREFIX</alias>
      <options description="true" map="" />
    </field>
    <field>
      <name>EMP_FIRST_NAME</name>
      <alias>FIRST_NAME</alias>
    </field>
    <field>
      <name>EMP_MIDDLE_NAME</name>
      <alias>MIDDLE_NAME</alias>
    </field>
    <field>
      <name>EMP_LAST_NAME</name>
      <alias>LAST_NAME</alias>
    </field>
  </table>
</jdbc-tables>
```

The 'name' element is the only required element of the 'field' element.

### Field Options element

The 'options' element currently supports two functions:

- forces a field to display the description (if linked to an option list)
- maps a set of values from a field to a different set of values

### Other field-related information

Company number and employee number fields are automatically added fields in a table and cannot be changed.

The field name (<field> element) must match the automatically created name from the original Solution Series field name but cannot be the actual field name. Replacing the dash (-) with an underscore (\_) changes the Solution Series field name. Any forward slashes in the field name must also be converted to underscores and any inquiry-only field is appended with '-INQ'.

The SQL command line tool can provide a list of the generated field names for a specified table. In order to determine if the field is associated with an option list, you must view The Solution Series form that contains the field. Due to the way the fields are used and duplicated within a form, field names can appear with sequence numbers appended to the name and may never contain data.

### Field alias element

An 'alias' is an optional element that provides an alternate name for a field. The alternate name may be contained in the request from the client application. The following example shows the definition of EF-SCR with a single field called SEX\_CODE. The field is given an alias of 'GENDER' and has the option set to display the option list description. The map option contains an empty string which means that no map is defined. A map is a reference to a lookup table file that converts Solution Series data to other values.

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-tables>
  <table>
    <screen>EF-SCR</screen>
    <name>EMP_BASIC_DETAILS</name>
    <field>
      <name>SEX_CODE</name>
      <alias>GENDER</alias>
      <options description="true" map="" />
    </field>
  </table>
</jdbc-tables>
```

### Map option

The map option is available for any field type but is more useful for a field connected to an option list. If you use a map option, the data it maps depends on the description setting. If the description is false, the lookup table maps the Solution Series option list codes to other code values (although it can map to anything you like). If the description is true, the lookup table maps the description to other description values. The following is an example of mapping options list descriptions for GENDER to Swedish language equivalents.

```
<?xml version="1.0" encoding="UTF-8"?>
<jdbc-tables>
  <table>
    <screen>EF-SCR</screen>
    <name>EMP_BASIC_DETAILS</name>
    <field>
      <name>SEX_CODE</name>
      <alias>GENDER</alias>
      <options description="true" map="gender_swedish.properties" />
    </field>
  </table>
</jdbc-tables>
```

The lookup table (in this example, `gender_swedish.properties`) is a text file that you can edit with any text editor. The `gender_swedish.properties` file contains the following lines:

```
Male=Man
Female=Kvinna
Unclassified=Okänt
```

In this example, when data is read from EF-SCR, the option list (PP41) code is converted to an option list description. The option list description is then converted to a value from the map file (`gender_swedish.properties`). If no match is found in the map file, the original value is returned.

## Validate the SQL statement

Use the eCyborg SQL test tool to check the syntax of the SQL statement in your mapping file. The SQL tool connects to a database and allows queries to run or displays other information about the database. You can use the SQL tool interactively or with command line prompts. The properties file reads the database connection properties or you can specify the database URL, login name, and password on the command line. Specifying your own database properties overrides the values of the properties 'URL', 'user', and 'pass' in the properties file. You can set the following variables in the `sqltool.properties` file in addition to the connection properties described in the eCyborg SQL Driver documentation.

Variable	Description
<code>url</code>	The database URL.
<code>user</code>	User name for database access.
<code>password</code>	Password for database access.
<code>line-length</code>	Maximum line length when printing query results. A value of zero means do not truncate lines. The default is 80 in interactive mode or 0.

The following is a Properties file example:

```
url=jdbc:cyborg:cybtest
user=xxxx
password=xxxx
```

The user code and password in the properties file example above are those assigned the client (requesting) application.

### Quick test

The easiest way to determine if a form will work is to test it with the SQL command line tool. The following example is a possible scenario for testing:

```
SELECT * FROM EMP_BASIC_DETAILS
```

If you execute the SQL statement example above, you will find that an employee has not been selected. The eCyborg SQL driver reads data from forms for individual employees. If a query is run without specifying an employee, it will run for several standard test employees in the '999999' company.

## Starting the SQL tool

Running the `sqltool.exe` program starts the eCyborg SQL Tool. By default, the connection properties are read from `sqltool.properties`. The directory identified in the first line of the example below is where you installed the eCyborg SQL driver:

```
C:\apps\jdbc>sqltool
Cyborg SQL Tool v1.1
Connected
=>
```

You can specify an alternative properties file, with the '-P' switch.

```
C:\apps\jdbc>sqltool -P sql.properties
```

The '-P' option specifies a file containing connection properties. The connection properties contain all the parameters needed to establish a connection to the eCyborg SQL server. In the example, the connection properties connect to the default environment on a server named 'cybserver1' as user 'xxxx'.

```
user=xxxx
password=xxxx
url=jdbc:cyborg:cybserver1/default
```

When the SQL tool establishes a connection, the Collaborative Platform displays a prompt. To see a list of the available commands, type 'help' and to exit the SQL Tool, type 'quit'.

## Command line

Parameters in angle brackets are required and parameters in square brackets are optional.

Option 1	Option 2	Description
-h	--help	Print command line help text.
-P <FILE>	--properties <FILE>	Read the connection properties from FILE instead of the default, sqltools.properties.
-u <URL>	--url <URL>	Set the database url to URL, overriding the value set in the properties file, if any.
-l <NAME>	--login <NAME>	Set the login name to NAME, overriding the value set in the properties file, if any.
-p <PASS>	--password <PASS>	Set the password to PASS, overriding the value set in the properties file, if any.
-q <QUERY>	--query <QUERY>	Execute a SELECT query in non-interactive mode. QUERY is the SQL text and must be quoted.
-t -tables	--list -tables	List tables in non-interactive mode.
-f <TABLE>	--list-fields <TABLE>	List fields of TABLE in non-interactive mode.
-L <NUMBER>	--line-length <NUMBER>	When printing query results, truncate the lines at NUMBER characters. A value of 0 means do not truncate lines. The default is 80 in interactive mode and 0 otherwise.

## Examples

The following are examples of specific command line options and their output:

- `sqltool -P default.properties -t`  
Lists all the tables, taking the connection properties from the file `default.properties`.
- `sqltool -P default.properties -t E%`  
Lists all the tables beginning with 'E' (not case sensitive).
- `sqltool -P default.properties -f EMP_BASIC_DETAILS`  
Lists all the fields in the table `EMP_BASIC_DETAILS`.
- `sqltool -P default.properties -q "SELECT * FROM EMP_BASIC_DETAILS WHERE COMPANY_NUMBER = '999999' AND EMPLOYEE_NUMBER = '1001'"`  
Runs the given query and prints out a line with all the field names, followed by a line for each row of data.

## Interactive mode

If you do not specify an operation on the command line, the SQL tool starts in interactive mode. You can enter the following commands at the prompt in upper or lower case:

Command	Description
help or ?	Print interactive mode help text.
tables	List the tables in the database.
fields <TABLE>	List the fields in TABLE.
select ...	Execute a SELECT query.
<NUMBER>	Reprint the query results starting at column index NUMBER.
length <NUMBER>	Set maximum line length to NUMBER characters. A value of 0 means do not truncate lines. The default is 80 in interactive mode.
quit or q	Exit the program.
update	Changes one or more fields in one or more rows.
insert	Allows the SQL driver to add a new record to a table.

## SQL-like syntax

The SQL-like syntax section focuses on the eCyborg SQL SELECT statements that respond to requests from the Collaborative Platform. The current available operations are listed below:

- List the tables in the database, including table type and any comment.
- List the fields in a table, including the data type and length, and whether the field is a key field.
- Execute a query (SELECT only at this time).

The Collaborative Platform supports SQL updates (UPDATE and INSERT). The SQL statements are included in the mapping file.

### Basic usage

The SELECT statement retrieves data from a table. The Collaborative Platform views each Solution Series form as a table. Note that table names, field names, and keywords are not case-sensitive. The following example details the format of a basic SELECT statement:

- SELECT <FIELD\_LIST> FROM <TABLE\_NAME>

The TABLE\_NAME is the name of the table and the FIELD\_LIST is a comma-delimited list of field names. You can use an asterisk (\*) as a wildcard character to select all fields in a table. You can give the fields an ALIAS (a different name) in the results by using the AS keyword.

Examples	Description
SELECT * FROM EMP_BASIC_DETAILS	Selects all fields in the table EMP_BASIC_DETAILS.
SELECT FIRST_NAME, MIDDLE_NAME, LAST_NAME from EMP_BASIC_DETAILS	Selects just the specified fields from the table.
SELECT FIRST_NAME AS first, MIDDLE_NAME AS middle, LAST_NAME AS last from EMP_BASIC_DETAILS	Selects the same fields as the preceding example, but the output fields are renamed 'first middle last'.

### Interactive usage

The following sections provide commands to list system tables and fields.

#### List tables

To list the available tables, use the 'tables' command:

```
=>tables
COMPANY_ADDRESS ..... (TABLE)
COMPANY_PROPERTY ..... (TABLE)
DRIVER_LICENSE ..... (TABLE)
EMERGENCY_CONTACTS ..... (TABLE)
...
```

#### List fields

To list the fields in a table, use the 'fields' command followed by the name of the table:

```
=>fields EMP_BASIC_DETAILS
COMPANY_NUMBER ..... (VARCHAR, 6, key)
EMPLOYEE_NUMBER ..... (VARCHAR, 10, key)
NAME_CODE ..... (INTEGER, 3, key)
NAME_PREFIX ..... (VARCHAR, 0)
...
```

## Testing your SQL statement

To run a SELECT query, enter the SQL statement at the prompt. Examine the results to determine if the data contains the information required by the client application. For example, some fields may be repeated and some may not be required. Some may be linked to option lists but the description is required. In these instances, the table definition file should be edited as well as the field elements.



Refer to 'Modifying the table definition file (on page 52)' and 'Adding field elements to the table definition file (on page 55)' for information on editing the table definition table.

By default, the results are truncated at 80 characters in interactive mode to avoid line wrapping. You can set a different value in the properties file, on the command line, or at the SQL Tool prompt.

```
=>SELECT * FROM EMP_BASIC_DETAILS WHERE COMPANY_NUMBER = '999999' AND
EMPLOYEE_NUMBER = '1001'
-----
| COMPANY_NUMBER | EMPLOYEE_NUMBER | NAME_CODE | NAME_PREFIX
-----
| 999999          | 1001             | 1         | Ms
-----
```

By entering the index at the prompt, you can reprint the results starting at a different column index.

```
=>13
-----
| ADDRESS_2 | CITY          | STATE | COUNTRY | ZIP_CODE |
-----
| UNIT 4    | EL SEGUNDO   | IL    | US      | 93101    |
-----
=>
```

*Note:* If the results are hard to read, you can run the SQL statement in non-interactive mode from the command prompt and send the output to a file.

## Non-interactive usage

The following sections provide commands to list system tables and fields.

### Testing the SQL statement - non-interactive usage

To run a SELECT query, use the '-q' switch followed by the SQL statement in quotes. If a maximum line length has been set either in the properties file or on the command line, the results are truncated at the maximum line length.

```
C:\apps\jdbc>sqltool -q "SELECT * FROM EMP_BASIC_DETAILS WHERE COMPANY_NUMBER =
'999999' AND EMPLOYEE_NUMBER = '1001'"
-----
| COMPANY_NUMBER | EMPLOYEE_NUMBER | NAME_CODE | NAME_PREFIX
-----
| 999999          | 1001             | 1         | Ms
-----
```

### List tables - non-interactive usage

Use the '-t' switch, to list the available tables:

```
C:\apps\jdbc>sqltool -t
COMPANY_ADDRESS ..... (TABLE)
COMPANY_PROPERTY ..... (TABLE)
DRIVER_LICENSE ..... (TABLE)
EMERGENCY_CONTACTS ..... (TABLE)
...
```

### List fields - non-interactive usage

Use the '-f' switch, to list the fields in a table followed by the name of the table:

```
C:\apps\jdbc>sqltool -f EMP_BASIC_DETAILS
COMPANY_NUMBER ..... (VARCHAR, 6, key)
EMPLOYEE_NUMBER ..... (VARCHAR, 10, key)
NAME_CODE ..... (INTEGER, 3, key)
NAME_PREFIX ..... (VARCHAR, 0)
...
```

### Exit codes

If running in non-interactive mode from a batch file, you can use the following exit codes to determine the outcome of a run:

Code	Meaning
0	Success
1	Command line argument is missing or is in wrong format
2	Unknown option or unexpected argument
3	Error reading properties file
4	No database URL specified
5	SQL error
6	Unknown error

## Query map file

You use a text editor to create a new file that becomes the XML map file containing your query. Map file names are case-sensitive and a map file has several distinct parts.

### XML declaration

The XML declaration specifies the version of XML and the encoding information being used by the map file. You can enter this information based on your system or copy the declaration from one of the eCyborg delivered mapping files. The following is an example from an eCyborg delivered map file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

### Database element

The database element of the map file defines the eCyborg SQL driver and the URL used to connect to the Solution Series data store:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
<database url="jdbc:cyborg:localhost/default"
  driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>
```

### Query element

The query element is the eCyborg SQL parameterized statement that specifies the columns to be retrieved. The query element contains a 'parameters' sub-element within which the name and data type of each of the SQL statements parameters is defined. The name of each parameter dictates the name of the constraint that must be supplied by a client application that requests a view based on the mapping file.

Using the values provided by the client application, you must build SQL statements. By using question marks (?) instead of hard-coded values, these parameter markers are bound to application variables. To retrieve a specific employee's data, the Collaborative Platform has only to set the value of the variables and execute the statement.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
<database url="jdbc:cyborg:localhost/default"
  driver="com.cyborg.jdbc.client.XmlDriver" >
</database>
<query sql="SELECT * FROM EMP_BASIC_DETAILS WHERE COMPANY_NUMBER = ? AND
EMPLOYEE_NUMBER = ? AND
NAME_CODE = 1"/>
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
    </parameter>
    <parameter ordinal="2">
      <name>EMPLOYEE_NUMBER</name>
      <type>STRING</type>
```

```
</parameter>
<parameter ordinal="3">
  <name>NAME_CODE<>
    <type>STRING</type>
  </parameter>
</parameters>
```

Collaborative Platform Data Types
-----------------------------------

STRING
--------

INTEGER
---------

DECIMAL
---------

DATE
------

### Root element

The 'root' element provides the name of the destination root element and the name of the elements that represent either Solution Series records or database rows.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/default"
    driver="com.cyborg.jdbc.client.XmlDriver">
  </database>
  <query sql="SELECT * FROM EMP_BASIC_DETAILS"/>
  <root name="PersonInfo" rowName="Person">
    <content>FIRST_NAME</content>
  </root>
</mapping>
```

### Element element

The 'element' element provides the new element name and any attribute or content elements. The 'element' element may also provide any attribute or content elements. For example, if the 'LastName' element has a 'code' attribute that represents the 'name code' field from the employee record and the employee's LAST\_NAME as the content, the map file would represent it as the following:

```
<element name="LastName">
  <content name="code">NAME_CODE</content>
  <content>LAST_NAME</content>
</element>
```

## Test the Read functionality

The Collaborative Platform test harness included in the ecp.war file verifies that your connection is working correctly. You can access the client application using an HTML page that is included in the ecp.war file:

- read.htm

The read.htm allows you to enter a 'view' and its corresponding query parameters for a given collaboration. Once you have entered your data, submit the request to the Collaborative Platform and wait for a SOAP response to confirm the connection.

**See also:**

- *Using the Test Harness*

## CHAPTER 4

# Write functionality

---

## In This Chapter

Structure of the mapping file .....	36
Transforming an incoming XML document .....	40
Defining a Web Service View .....	41
Invoking the Web Service.....	42
Test the Write functionality .....	45

## Structure of the mapping file

The Collaborative Platform uses a mapping file to specify all aspects of the data transfer including the structure of the incoming data and the SQL statement used to update the Solution Series. The mapping file is an XML document that contains several elements of information:

### Database

The “database” element defines the eCyborg SQL driver and URL used to connect to the Solution Series data store.

### Insert

The “insert” element has an attribute named “SQL”. The value of the SQL attribute is a parameterised SQL statement that the platform uses to add data to the Solution Series. The “insert” element contains a “parameters” sub-element that defines the name and type of each of the SQL statement’s parameters. The Collaborative Platform replaces the parameters with values supplied by the client application.

### Update

The “update” element also has an attribute named “SQL”. The value of the SQL attribute is a parameterised SQL statement that the platform uses to update Solution Series data. The “update” element contains a “parameters” sub-element that defines the name and type of each of the SQL statement’s parameters. The Collaborative Platform replaces the parameters with values supplied by the client application.

### Root

The “root” element represents the overall structure of an incoming XML document. Attributes of the “root” element specify the root element and the name of the elements that represent individual records in an incoming XML document.

## Database information

To set up a successful data transfer, the mapping file must designate two types of information:

- information establishing a connection to the data source
- information describing the SQL operation performed during the processing of the mapping file

The mapping file designates the necessary information through two child elements of the “root”. The first deals with information on the database:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/default"
    driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>
```

The database element holds information on the URL of the data source and on the driver to use. In the example, you connect to a Solution Series environment named “default” using the eCyborg SQL driver.

## Adding Solution Series records

The mapping file must contain an “insert” element to add records to the Solution Series. The “insert” elements and its sub-elements, provide the mechanism for binding data in an incoming XML document to Solution Series fields.

When the Collaborative Platform processes the mapping in the example below, it replaces the first SQL parameter with the value of the COMPANY\_NUMBER element from the incoming document and the second parameter with the value of the EMPLOYEE\_NUMBER element. The value of each “parameter” element’s “ordinal” attribute binds it to a specific parameter.

Each “parameter” element must contain an XPath expression that describes the location of the required data in the incoming document. The Collaborative Platform selects the first node that corresponds to the XPath expression. In order to avoid unexpected results, you should always use absolute XPath expressions that resolve to a single “text node”.

```
<insert sql="INSERT INTO EMERGENCY_CONTACTS VALUES(?, ?, ?, ?, ?, ?, ?, ?)">
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/CompanyName/text()/</expr>
    </parameter>
    <parameter ordinal="2">
      <name>EMPLOYEE_NUMBER</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmployeeNumber/text()/</expr>
    </parameter>
    <parameter ordinal="3">
      <name>KEY</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/Key/text()/</expr>
    </parameter>
    <parameter ordinal="4">
      <name>CONTACT</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/Contact/text()/</expr>
    </parameter>
    ...
  </parameters>
</insert>
```

## Modifying Solution Series records

The mapping must contain an “update” element to update Solution Series records. The “update” element, and its sub elements, provides a mechanism for binding data in an incoming XML document to Solution Series fields. However, in accordance with the semantics of the SQL “update” statement, the record must already exist.

In the example, when the Collaborative Platform processes the mapping file, it replaces the first SQL parameter with the value of the CONTACT element from the incoming document. It replaces the second parameter with the value of the CONTACT\_AREA\_CODE element. The value of each “parameter” element’s “ordinal” attribute binds it to a specific parameter.

```
<update sql="UPDATE EMERGENCY_CONTACTS SET CONTACT = ?, CONTACT_AREA_CODE = ?,  
CONTACT_ADDRESS = ? WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER = ? AND KEY = 1">  
  <parameters>  
    <parameter ordinal="1">  
      <name>CONTACT</name>  
      <type>STRING</type>  
      <expr>/ContactInfo/Contact/Contact/text()</expr>  
    </parameter>  
    <parameter ordinal="2">  
      <name>CONTACT_AREA_CODE</name>  
      <type>INTEGER</type>  
      <expr>/ContactInfo/Contact/AreaCode/text()</expr>  
    </parameter>  
    <parameter ordinal="3">  
      <name>CONTACT_ADDDDRESS</name>  
      <type>STRING</type>  
      <expr>/ContactInfo/Contact/Address/text()</expr>  
    </parameter>  
    <parameter ordinal="4">  
      <name>COMPANY_NUMBER</name>  
      <type>STRING</type>  
      <expr>/ContactInfo/Contact/CompanyNumber/text()</expr>  
    </parameter>  
    <parameter ordinal="5">  
      <name>EMPLOYEE_NUMBER</name>  
      <type>STRING</type>  
      <expr>/ContactInfo/Contact/EmployeeNumber/text()</expr>  
    </parameter>  
  </parameters>  
</update>
```

## Structure of the incoming XML document

The mapping file also specifies the structure of an incoming XML document. The “root” portion of the mapping file specifies the structure of an incoming XML document. It provides the binding between elements in the incoming document and parameters in the SQL statement. In the example below, the root element specifies that the incoming XML document should contain an element named ‘PersonInfo’. The ‘PersonInfo’ element should contain all data intended for the Solution Series, and individual records should each be enclosed in an element named ‘Person’.

```
<root name="PersonInfo" rowName="Person">
  <element name="FirstName">
    <content>FIRST_NAME</content>
  </element>
  <element name="LastName">
    <content>LAST_NAME</content>
  </element>
  <element name="MiddleName">
    <content>MIDDLE_NAME</content>
  </element>
  <element name="Prefix">
    <content>NAME_PREFIX</content>
  </element>
  <element name="Suffix">
    <content>NAME_SUFFIX</content>
  </element>
  <element name="AddressLine">
    <content>ADDRESS</content>
  </element>
  . . .
</root>
```

Mapping files are reusable for multiple document schemas. You can create a number of generic mapping files, each defining one common Human Resources object:

- Company
- Department
- Job
- Employee

As a result, you can use the mapping files to control the exchange of data with multiple collaborating applications.

## Transforming an incoming XML document

The Collaborative Platform uses the eCyborg SQL driver to connect to a Solution Series data source. An SQL driver table is a tabular data structure where each record corresponds to a “row” and fields within each record correspond to “columns”. Regardless of the original format of the enclosing document, the Collaborative Platform must convert all XML data in a single incoming message to this tabular format in order to update the Solution Series. The Collaborative Platform allows you to apply an XSLT Stylesheet to an incoming document in order to produce the simplified tabular format that maps to a Solution Series table.

### XSLT applied

By using XSLT, a mapping file can serve as the data transfer definition for a variety of incoming documents. For example, the HR-XML Consortium has defined the “PersonName” and “PostalAddress” schemas. The data contained in a document that conforms to either “PersonName” or “PostalAddress” map to fields in a single Solution Series table:

- EMP\_BASIC\_DETAILS

The Collaborative Platform has two XSLT stylesheets that transform PersonName to PersonInfo and PostalAddress to PersonInfo. In the view definition file, you can specify that the transformations are applied to all incoming documents and reuse the mapping file.

## Defining a Web Service View

The Collaborative Platform uses a view definition file to determine which views are available to client applications. Each entry in the view definition file specifies the name of a view, the mapping file, and optionally two XSLT stylesheets. The “ecp:xsl-in” element of a view definition specifies the stylesheet that the Collaborative Platform applies to incoming documents. The example below is a view definition file that defines views for two HR-XML schemas:

- PersonName
- PostalAddress

When the Collaborative Platform receives a document that conforms to the HR-XML PersonName schema, it uses the XSL transformations specified in PersonNameIn.xsl to produce an intermediate document that is compatible with the definitions in PersonMap.xml. It is this intermediate document that the Collaborative Platform parses to obtain the values that are substituted into the SQL statement in the mapping file.

The following example also illustrates how to define a read-only view. Given the definition for PostalAddress, using this view to update the Solution Series causes the Collaborative Platform to return a SOAP fault to the collaborating application.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ecp:views xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">
  <ecp:view name="PersonName">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PersonName.xsl</ecp:xsl>
    <ecp:allow-write value='true'>
      <ecp:transform>PersonNameIn.xsl</ecp:transform >
    </ecp:allow-write>
  </ecp:view>

  <ecp:view name="PostalAddress">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PostalAddress.xsl</ecp:xsl>
  </ecp:view>
</ecp:views>
```

## Invoking the Web Service

The Collaborative Platform is a document-centric web service. The business logic is kept separate from the content of the XML documents that it uses. The service accepts an XML document that contains only data with no explicit binding to the applied business logic that will produce the response. The inbound document is not explicitly mapped to the business logic. That is, specific methods are not invoked on the service.

The Collaborative Platform applies its business logic to the inbound XML document and the document content determines the processing workflow. An inbound message to the Collaborative Platform contains the security credentials of the principal, the name of the view to be processed, and the name of the required operation. In the following example, see ‘insert’:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
<soap-env:Header>
  <ecp:ViewName
xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">PersonName</ecp:ViewName>
  <ecp:UserCode xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">USER</ecp:UserCode>
  <ecp>Password xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">PASS</ecp>Password>
  <ecp:operation
xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">write</ecp:operation>
</soap-env:Header>
<soap-env:Body>
  <PersonName>
    <GivenName>June</GivenName>
    <FamilyName>Meyer</FamilyName>
    <Affix type="formOfAddress">Ms</Affix>
  </PersonName>
</soap-env:Body>
</soap-env:Envelope>
```

## Processing the request

Upon receiving a request, the Collaborative Platform reads the view definition file indicating the mapping file that should be used to write a “PersonName” to the Solution Series. The PersonName element is extracted from the incoming SOAP message and used as the root of a temporary XML document. If a stylesheet has been specified, it is applied to this temporary document to produce an intermediate XML document that matches the definition contained within the mapping file. The following is an example of an intermediate document:

```
<?xml version="1.0" encoding="UTF-8" ?>
<PersonInfo>
  <Person>
    <FirstName>June</FirstName>
    <LastName>Meyer</LastName>
    <MiddleName/>
    <Prefix>Ms</Prefix>
    <Address1/>
    <Address2/>
    <City/>
    <State/>
```

```

    <Zip/>
    <Country/>
  </Person>
</PersonInfo>

```

If a stylesheet is not specified, the intermediate XML document is built directly from the `PersonName` element. If the mapping file defines an update element, the Collaborative Platform uses the XPath expression defined for each parameter to extract the required data values from the intermediate document. It then uses the element's SQL to attribute to construct an SQL UPDATE statement that is applied to the data store. For example:

- UPDATE EMP\_BASIC\_DETAILS SET NAME\_PREFIX = 'Ms', FIRST\_NAME = 'June', LAST\_NAME = 'Meyer' WHERE ..

If the SQL UPDATE fails and the mapping file defines an insert element, the Collaborative Platform uses the XPath expression defined for each parameter to extract the required data values from the intermediate document. It then uses the element's SQL attribute to construct an SQL INSERT statement that is applied to the data store. For example:

- INSERT INTO EMP\_BASIC\_DETAILS VALUES('Ms', 'June', 'Meyer')

The result of the SQL operation is reported to the collaborating application through an `ecp:Acknowledgement` document embedded in the body of a SOAP response message. For example:

```

<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header/>
  <soap-env:Body>
    <ecp:Acknowledgement xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">
      <ecp:RowsAffected>1</ecp:RowsAffected>
    </ecp:Acknowledgement>
  </soap-env:Body>
</soap-env:Envelope>

```

The collaborating application should check the value contained in the `ecp:RowsAffected` element to determine whether the write to the Solution Series succeeded.

## Error content defined

The XML response from Solution Series, returned in a SOAP envelope and viewed in a browser, contains either the requested data or an error message. The following is an example of an error message:

```
<soap-env:Envelope xmlns:soap env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header />
  <soap-env:Body>
    <soap-env:Fault>
      <faultcode>Client</faultcode>
      <faultstring>The request contains insufficient or invalid information and
        cannot be processed.</faultstring>
      <detail>
        <ecp:Fault xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">Unknown view:
          'CompanyProperty'</ecp:Fault>
        </detail>
      </soap-env:Fault>
    </soap-env:Body>
  </soap-env:Envelope>
```

## Test the Write functionality

Use the Collaborative Platform test harness included in the ecp.war file to verify that your connection is working correctly. You can access Solution Series using an HTML page that is also included in the ecp.war file:

- write.htm

The write.htm allows you to enter a View name defined in the ecp-views.xml file and open the corresponding XML data file. Once you have located the file, you submit the request to the Collaborative Platform and wait for a SOAP response to confirm the connection.

### **See also**

- *Using the Test Harness*

PART 2

## Creating Collaborative Schema

---

### In This Section

Creating a schema - Read side .....	49
Creating a schema - Write side .....	65

## CHAPTER 5

# Creating a schema - Read side

---

## In This Chapter

Introduction .....	50
Identifying the data source.....	51
Modifying the table definition file .....	52
Starting the eCyborg SQL tool .....	53
Testing the SQL statement.....	54
Adding field elements to the table definition file.....	55
Creating the map file .....	57
Creating a stylesheet .....	60
Editing ecp-views.xml .....	63
Performing a Read side test .....	64

## Introduction

The following directions provide information on configuring the Collaborative Platform to produce XML documents that conform to the Read side functionality of the collaborative relationship.

*Note: These exercises should be done in your test system environment.*

## Identifying the data source

A client application has requested hours and pay rate information for a specific employee:

- June Meyer - employee number 1001 and company code 999999

The hours and pay rate information is on the Employee Pay Rate or Salary form (H1-SCR). Perform the following steps to find the form containing the information you want:

### 1. Review the table definition file

You should review the contents of the XML table definition file in a text editor to determine if the form/table you need is already included.



*For practice, use a text editor to open the default.xml file.*

### 2. Use the Find command

Use the text editor Find command, to locate the form containing the data you need to send to the requesting web services application.



*For practice, search for the H1-SCR using your text editor's Find/Search command.*

## Modifying the table definition file

You must add a screen to the default.xml because the H1-SCR is not defined in the table definition file. If you have closed the default.xml file, re-open it using a text editor, and perform the following steps:

**1. Position cursor at end of table entries**



*For practice, move to the end of the last table entry in the file ('default.xml').*

**2. Insert the table tag to start the new form/table entry**

To maintain readability of the file, follow the indenting pattern shown for the eCyborg delivered entries for the EF-SCR. Use the tag to show the beginning of the new form/table entry.



*For practice, enter '<table>' and press Enter.*

**3. Enter the screen element**

Enter the beginning screen tag followed by the Solution Series form name and then close the screen tag.



*For practice, enter '<screen>H1-SCR</screen>'.*

**4. Enter the name element**

The table name is one you create and referenced by SQL. Use underscores (\_), not dashes (-), in the table name. Enter the beginning table name tag followed by the table name and then close the name tag.



*For practice, enter '<name>PAY\_RATE</name>'.*

**5. Close the table tag**

Your entry of the form (screen) and table name tags is now complete, close the table entry.



*For practice, enter '</tables>'.*

**6. Save the file**



*For practice, save the addition to the file before closing the file.*

## Starting the eCyborg SQL tool

1. **From the command prompt, type 'sqltool' to start the SQL Tool in interactive mode**

For example: C:\apps\jdbc>sqltool



*For practice, type 'sqltool' at the command prompt and press Enter.*

2. **Review the system response**

After pressing Enter, you should see the following display:

```
C:\apps\jdbc>sqltool
Cyborg SQL Tool v1.1
Connected
=>
```

## Testing the SQL statement

Perform the following steps to enter a Select statement in SQL and check the results:

**1. Enter a SELECT statement**

Enter a SQL SELECT statement to determine you can retrieve data from the H1-SCR. Enter a basic SELECT statement to test for data retrieval.



*For practice, at the '=' prompt, type: 'SELECT \* FROM PAY\_RATE' and press Enter.*

**2. Review the results**

If you cannot retrieve data from a Solution Series form directly, you will see various error responses.



*For practice, review the output from your query.*

**3. List the field names**

By using the 'Fields' command, you can list the field names in the table (Pay\_Rate) you created for the H1-SCR.



*For practice, at the prompt enter 'fields PAY\_RATE' and press Enter.*

**4. Take note of the field names**

The requesting client application wants to know the normal pay HED number, rate of pay, and normal number of hours worked by employee 1001. Record the names of the fields containing data that must be extracted to respond to the client application.



*For practice, record: 'HED\_NBR\_001', 'PAY\_RATE', and 'NORMAL\_HOURS'.*

**5. Exit the eCyborg SQL tool**

To exit the eCyborg SQL tool, type Quit and press Enter.

## Adding field elements to the table definition file

The client application extracts data from the table you created in the table definition file (default.xml) for the Employee Pay Rate or Salary form (H1-SCR). To extract the data, add fields to the table definition file for the H1-SCR entry:

### 1. Open the table definition file

Using a text editor, open the default.xml file.



*For practice, use your preferred text editor to open the default.xml file.*

### 2. Locate the table entry to modify

Using the text editor's Find/Search command, locate the entry previously made for the H1-SCR form.



*For practice, locate the H1-SCR entry.*

### 3. Add the field names

Follow the indenting pattern shown for the delivered entries for the EF-SCR. Position the cursor beneath the table name element and enter the field tag and the field name for each piece of data that will be extracted.



*For practice, enter these field name elements 'HED\_NBR\_001', 'PAY\_RATE', and 'NORMAL\_HOURS'.*

### 4. (Optional) Create an alias for the field name elements

The client application may specify that data returned be labeled in a particular way. You can modify labels by using the 'Alias' option.



*For practice, create the aliases of 'HED\_NUMBER', 'HOURLY\_RATE', and 'HOURS' for the field elements 'HED\_NBR\_001', 'PAY\_RATE', and 'NORMAL\_HOURS', respectively.*

Your current entry should look like the following example:

```
<table>
  <screen>H1-SCR</screen>
  <name>PAY_RATE</name>
    <field>
      <name>HED_NBR_001</name>
      <alias>HED_NUMBER</alias>
    </field>
    <field>
      <name>PAY_RATE</name>
      <alias>HOURLY_RATE</alias>
    </field>
    <field>
      <name>NORMAL_HOURS</name>
      <alias>HOURS</alias>
    </field>
  </table>
</jdbc-tables>
```

**5. Save and close the table definition file**



*For practice, save and close the default.xml file.*

## Creating the map file

Using a text editor, open a new file in the conf. directory located in the ecp.war file:

- ServletExec data/default/ecp/war/web.inf/conf

The new file will become the XML map file containing your query.

### 1. Enter the XML declaration

The XML declaration specifies the version of XML you are using and the encoding information. You can enter this information based on your system or copy the declaration from one of the eCyborg delivered mapping files.



*For practice, copy the declaration from the eCyborg delivered PersonMap.xml into your newly opened map file.*

Your entry should look similar to the following example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

### 2. Identify the database element

The 'database' element of the map file defines the eCyborg SQL driver and the URL used to connect to the Solution Series data store.



*For practice, enter the mapping tag and the database element for the eCyborg SQL driver and the URL used to connect to the Solution Series data store on your system.*

Your entry should appear similar to the following example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/default"
    driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>
```

### 3. Enter the query element

Type your SQL-like statement specifying the columns you want retrieve. Remember that the query element has a sub-element: parameters.



*For practice, enter the query sql tag followed by: "SELECT \* FROM PAY\_RATE WHERE COMPANY\_NUMBER = ? AND EMPLOYEE\_NUMBER = ?" within the query element. Then enter the 3 parameter sub-elements.*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/default"
    driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>
  <query sql="SELECT * FROM PAY_RATE WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER
= ?">
    <parameters>
```

```
<parameter ordinal="1">
  <name>COMPANY_NUMBER</name>
  <type>STRING</type>
</parameter>
<parameter ordinal="2">
  <name>EMPLOYEE_NUMBER</name>
  <type>STRING</type>
</parameter>
</parameters>
</query>
```

## 4. Enter the root elements

The 'root' element provides the name of the destination root element and the name of the elements that will represent either Solution Series records or database rows.



*For practice, enter the root name as 'PAY\_RATE' and the rowName as 'Employee'.*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
<database url="jdbc:cyborg:localhost/default"
  driver="com.cyborg.jdbc.client.XmlDriver" >
</database>
<query sql="SELECT * FROM PAY_RATE WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER =
?">
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
    </parameter>
    <parameter ordinal="2">
      <name>EMPLOYEE_NUMBER</name>
      <type>STRING</type>
    </parameter>
  </parameters>
</query>
<root name="EmployeeInfo" rowName="Employee">
```

## 5. Add the element element

The 'element' element provides the new element name and any attribute or content elements.



*For practice, enter the element element as: '<content>HED\_NBR</content>'.*

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
<database url="jdbc:cyborg:localhost/default"
  driver="com.cyborg.jdbc.client.XmlDriver" >
</database>
<query sql="SELECT * FROM PAY_RATE WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER =
?">
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
    </parameter>
    <parameter ordinal="2">
```

```
<name>EMPLOYEE_NUMBER</name>
<type>STRING</type>
</parameter>
</parameters>
</query>
<root name="EmployeeInfo" rowName="Employee">
  <element name="HEDNumber">
    <content>HED_NUMBER</content>
  </element>
  <element name="HourlyRate">
    <content>HOURLY_RATE</content>
  </element>
  <element name="NormalHours">
    <content>HOURS</content>
  </element>
</root>
</mapping>
```

If the content element does not contain a valid field name, eCyborg displays an error message.

## 6. Save your map file

Save your newly created map file with an XML extension. Remember, map files are case sensitive.



*For practice, save your xml map file as PayMap.xml.*

## Creating a stylesheet

If the client application requires data in a format not used by the Solution Series, you may have to create a stylesheet to produce a properly formatted response to the client. Using a text editor, open a new file in the conf. directory that will contain the stylesheet for your query response, and perform the following steps:

### 1. Enter the XML declaration

The XML declaration specifies the version of XML you are using and the encoding information. You may enter this information based on your system or copy the declaration from one of the eCyborg delivered mapping files.



*For practice, copy the declaration from the eCyborg delivered PersonName.xml into your newly opened stylesheet file.*

Your entry should look similar to the following example:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" />
<xsl:strip-space elements="*" />
  <xsl:template match="/">
```

### 2. Identify the map file

Specify the name of the map file you want to apply the stylesheet to:



*For practice, enter the map tag and the stylesheet application elements.*

Your entry should appear similar to the following example:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" />
<xsl:strip-space elements="*" />
  <xsl:template match="/">
    <PayMap>
<xsl:apply-templates />
  </PayMap>
</xsl:template>
```

### 3. Enter the template name

Type the name of the stylesheet template you want to use.



*For practice, enter the 'template match' as 'Person with the field names selected as 'HEDNumber', 'HourlyRate', and 'NormalHours'.*

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" />
<xsl:strip-space elements="*" />
  <xsl:template match="/">
    <PayMap>
<xsl:apply-templates />
```

```

</PayMap>
</xsl:template>
<xsl:template match="Person">
<xsl:apply-templates select="HEDNumber" />
<xsl:apply-templates select="HourlyRate" />
<xsl:apply-templates select="NormalHours" />
</xsl:template>

```

#### 4. Enter the template match for the fields

Apply the template match to the specific fields in the stylesheet:



*For practice, enter the template match for the fields 'HEDNumber', 'HourlyRate', and 'NormalHours'.*

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" />
<xsl:strip-space elements="*" />
<xsl:template match="/">
  <PayMap>
<xsl:apply-templates />
</PayMap>
</xsl:template>
<xsl:template match="Employee">
<xsl:apply-templates select="HEDNumber" />
<xsl:apply-templates select="HourlyRate" />
<xsl:apply-templates select="NormalHours" />
</xsl:template>
<xsl:template match="HEDNumber">
  <HEDNumber>
<xsl:apply-templates />
</HEDNumber>
</xsl:template>
<xsl:template match="HourlyRate">
  <HourlyRate>
<xsl:apply-templates />
</HourlyRate>
</xsl:template>
<xsl:template match="NormalHours">
  <NormalHours>
<xsl:apply-templates />
</NormalHours>

```

#### 5. Close the template and stylesheet tags

Close the template and stylesheet tags to complete your stylesheet entries. Your stylesheet should look like the following example:

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="xml" />
<xsl:strip-space elements="*" />
<xsl:template match="/">
  <PayMap>
<xsl:apply-templates />
</PayMap>
</xsl:template>
<xsl:template match="Person">
<xsl:apply-templates select="HEDNumber" />

```

```
<xsl:apply-templates select="HourlyRate" />
<xsl:apply-templates select="NormalHours" />
</xsl:template>
  <xsl:template match="HEDNumber">
    <HEDNumber>
<xsl:apply-templates />
</HEDNumber>
</xsl:template>
  <xsl:template match="HourlyRate">
    <HourlyRate>
<xsl:apply-templates />
</HourlyRate>
</xsl:template>
  <xsl:template match="NormalHours">
    <NormalHours>
<xsl:apply-templates />
</NormalHours>
</xsl:template>
</xsl:stylesheet>
```

### 6. Save your stylesheet

Save your new map file with an XSL extension.



*For practice, save your xsl file as PayXsl.xsl*

## Editing ecp-views.xml

Perform the following steps to edit ecp-views.xml:

- 1. Open the view xml file**

Open the eCyborg delivered ecp-views.xml file using a text editor.

- 2. Create a new entry**

Position the cursor on the line after the last occurrence of '</ecp.view>' and insert the tags for the names of the view, the mapping file, and the stylesheet to produce the file eCyborg returns to the client application.



*For practice, enter the view name as 'PayMap' and the map name as 'PayMap.xml' and stylesheet as 'PayXsl.xsl'.*

Your ecp-views.xml file should look like the following example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ecp:views xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">
  <ecp:view name="PersonName">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PersonName.xsl</ecp:xsl>
    <ecp:allow-write value='false'>
  </ecp:view>

  <ecp:view name="PostalAddress">
    <ecp:map>PersonMap.xml</ecp:map>
    <ecp:xsl>PostalAddress.xsl</ecp:xsl>
    <ecp:allow-write value='false'>
  </ecp:view>

  <ecp:view name="PayMap">
    <ecp:map>PayMap.xml</ecp:map>
    <ecp:xsl>Payxsl.xsl</ecp:xsl>
    <ecp:allow-write value='false'>
  </ecp:view>
</ecp:views>
```

- 3. Save and close the file**

## Performing a Read side test

Perform the following steps to test the 'Read' side connection:

**1. Access the Collaborative Platform Web Services**

To run the custom application, open a browser and access the following URL:

`http://hostname/ecp/read.htm`

Replace 'hostname' with the name or IP address of the machine on which the Collaborative Platform (ecp.war) is installed. The default TCP/IP port used by http servers is 80. If your HTTP server is not configured to accept requests on the default port, you must include the port number in the URL. For example:

`http://host:8080/ecp/read.htm`

**2. Press Enter**

Your web browser displays an HTML form (Schema Read Test Page) and its fields with their default values.

**3. Add your desired environment details**

Before submitting the form, ensure that your user name and password are correct and enter the desired view name ('PayMap') you defined in the ecp-views.xml file. Next, enter valid company and employee numbers in the Parameter Name column and their corresponding values in the Value column for your Solution Series environment.

**4. Submit the form**

At the bottom of the form, click the Submit button to send the request to the Web Collaboration web service.

**5. Review the results**

The SOAP response appears as XML in the browser.

## CHAPTER 6

# Creating a schema - Write side

---

## In This Chapter

Introduction .....	66
Identifying the data source.....	67
Modifying the table definition file .....	68
Starting the eCyborg SQL tool .....	70
Testing the SQL statement.....	71
Adding field elements to the table definition file.....	72
Creating the map file .....	74
Editing ecp-views.xml .....	79
Performing a Write side test .....	80

## Introduction

The following directions provide information on configuring the Collaborative Platform to produce XML documents that conform to the Write side functionality of the collaborative relationship.

*Note: These exercises should be done in your test system environment.*

## Identifying the data source

A client has requested a change to the last name and address of a specific employee:

- June Meyer - employee number 1001 and company code 999999

The first and last name information is on the EMP\_NAME\_ADDRESS table (FF-SCR). Perform the following steps to find the form containing the information you want:

### 1. Review the table definition file

You should review the contents of the XML table definition file in a text editor to determine if the form/table you need is already included.



*For practice, use a text editor to open the default.xml file.*

### 2. Use the Find command

Use the text editor Find command, to locate the form containing the data you need to send to the requesting web services application.



*For practice, search for the FF-SCR using your text editor's Find/Search command.*

## Modifying the table definition file

You must add a screen to the default.xml because the FF-SCR is not defined in the table definition file. Ensure that the FF-SCR is available:

### 1. Position cursor at end of table entries



*For practice, move to the end of the last table entry in the file ('default.xml').*

### 2. Insert the table tag to start the new form/table entry

To maintain readability of the file, follow the indenting pattern shown for the eCyborg delivered entries for the FF-SCR. Use the tag to show the beginning of the new form/table entry.



*For practice, enter '<table>' and press Enter.*

### 3. Enter the screen element

Enter the beginning screen tag followed by the Solution Series form name and then close the screen tag.



*For practice, enter '<screen>FF-SCR</screen>'.*

### 4. Enter the name element

The table name is one you create and referenced by SQL. Use underscores (\_), not dashes (-), in the table name. Enter the beginning table name tag followed by the table name and then close the name tag.



*For practice, enter '<name>EMP\_NAME\_ADDRESS</name>'.*

### 5. Close the table tag

Your entry of the form (screen) and table name tags is now complete, close the table entry. Your complete sample should look like the following:



*For practice, enter '</tables>'.*

```
<table>
  <screen>FF-SCR</screen>
  <name>EMP_NAME_ADDRESS</name>
  <field>
    <name>EMPLOYEE_NAME_INQ</name>
    <alias>EMPLOYEE_NAME_INQ</alias>
  </field>
  <field>
    <name>EMP_NAME_CODE</name>
    <alias>EMP_NAME_CODE</alias>
  </field>
  <field>
    <name>EMP_NAME_PREFIX</name>
    <alias>EMP_NAME_PREFIX</alias>
    <options description="true" map="" />
  </field>
</table>
```

```

</field>
<field>
  <name>EMP_FIRST_NAME</name>
  <alias>EMP_FIRST_NAME</alias>
</field>
<field>
  <name>EMP_MIDDLE_NAME</name>
  <alias>EMP_MIDDLE_NAME</alias>
</field>
<field>
  <name>EMP_LAST_NAME</name>
  <alias>EMP_LAST_NAME</alias>
</field>
<field>
  <name>EMP_NAME_SUFFIX</name>
  <alias>EMP_NAME_SUFFIX</alias>
  <options description="true" map="" />
</field>
<field>
  <name>ADDRESS</name>
  <alias>ADDRESS</alias>
</field>
<field>
  <name>ADDRESS_2</name>
  <alias>ADDRESS_2</alias>
</field>
<field>
  <name>EMP_CITY_US</name>
  <alias>EMP_CITY_US</alias>
</field>
<field>
  <name>EMP_STATE</name>
  <alias>EMP_STATE</alias>
</field>
<field>
  <name>EMP_ZIP_CODE</name>
  <alias>EMP_ZIP_CODE</alias>
</field>
<field>
  <name>EMP_COUNTRY_CODE</name>
  <alias>EMP_COUNTRY_CODE</alias>
  <options description="true" map="" />
</field>
</table>

```

## 6. Save the file



*For practice, save the addition to the file before closing the file.*

## Starting the eCyborg SQL tool

1. **From the command prompt, type 'sqltool' to start the SQL Tool in interactive mode**

For example: C:\apps\jdbc>sqltool



*For practice, type 'sqltool' at the command prompt and press Enter.*

2. **Review the system response**

After pressing Enter, you should see the following display:

```
C:\apps\jdbc>sqltool
Cyborg SQL Tool v1.1
Connected
=>
```

## Testing the SQL statement

Perform the following steps to enter a Select statement in SQL and check the results:

**1. Enter a SELECT statement**

Enter a SQL SELECT statement to determine you can retrieve data from the FF-SCR. Enter a basic SELECT statement to test for data retrieval.



*For practice, at the '=>' prompt, type: 'SELECT \* FROM EMP\_NAME\_ADDRESS' and press Enter.*

**2. Review the results**

If you cannot retrieve data from a Solution Series form directly, you will see various error responses.



*For practice, review the output from your query.*

**3. List the field names**

By using the 'Fields' command, you can list the field names in the table (EMP\_NAME\_ADDRESS) you created for the FF-SCR.



*For practice, at the prompt enter 'fields EMP\_NAME\_ADDRESS' and press Enter.*

**4. Exit the eCyborg SQL tool**

To exit the eCyborg SQL tool, type 'Quit' and press Enter.

## Adding field elements to the table definition file

The client application extracts data from the table you created in the table definition file (default.xml) for the Employee Name and Address (FF-SCR). To extract the data, add fields to the table definition file for the FF-SCR entry:

### 1. Open the table definition file

Using a text editor, open the default.xml file.



*For practice, use your preferred text editor to open the default.xml file.*

### 2. Locate the table entry to modify

Using the text editor's Find/Search command, locate the entry previously made for the FF-SCR form.



*For practice, locate the FF-SCR entry.*

### 3. Add the field names

Follow the indenting pattern shown for the delivered entries for the FF-SCR. Position the cursor beneath the table name element and enter the field tag and the field name for each piece of data that will be extracted.



*For practice, enter these field name elements 'EMP\_FIRST\_NAME', 'EMP\_MIDDLE\_NAME', 'EMP\_LAST\_NAME' and 'ADDRESS'.*

### 4. (Optional) Create an alias for the field name elements

The client application may specify that data returned be labeled in a particular way. You can modify labels by using the 'Alias' option. The example below is not a complete sample of code, however, your current entry should look similar to the following:

```
<table>
  <screen>FF-SCR</screen>
  <name>EMP_NAME_ADDRESS</name>
  <field>
    <name>EMP_FIRST_NAME</name>
    <alias>EMP_FIRST_NAME</alias>
  </field>
  <field>
    <name>EMP_MIDDLE_NAME</name>
    <alias>EMP_MIDDLE_NAME</alias>
  </field>
  <field>
    <name>EMP_LAST_NAME</name>
    <alias>EMP_LAST_NAME</alias>
  </field>
  <field>
    <name>ADDRESS</name>
    <alias>ADDRESS</alias>
  </field>
</table>
```

**5. Save and close the table definition file**



*For practice, save and close the default.xml file.*

## Creating the map file

Using a text editor, open a new file in the 'conf' directory located in the ecp.war file:

- ServletExec data/default/ecp/war/web.inf/conf

The new file will become the XML map file containing your query.

### 1. Enter the XML declaration

The XML declaration specifies the version of XML you are using and the encoding information. You can enter this information based on your system or copy the declaration from one of the eCyborg delivered mapping files.



*For practice, copy the declaration from the eCyborg delivered ContactMap.xml into your newly opened map file.*

Your entry should look similar to the following example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

### 2. Identify the database element

The 'database' element of the map file defines the eCyborg SQL driver and the URL used to connect to the Solution Series data store.



*For practice, enter the mapping tag and the database element for the eCyborg SQL driver and the URL used to connect to the Solution Series data store on your system.*

Your entry should appear similar to the following example:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<mapping>
  <database url="jdbc:cyborg:localhost/default"
    driver="com.cyborg.jdbc.client.XmlDriver" >
  </database>
```

### 3. Enter the Insert element

Type your SQL-like statement specifying the columns you want to access.

```
<insert sql="INSERT INTO EMP_NAME_ADDRESS VALUES(?, ?, 1, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?)">
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/CompanyName/text()/</expr>
    </parameter>
    <parameter ordinal="2">
      <name>EMPLOYEE_NUMBER</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmployeeNumber/text()/</expr>
    </parameter>
```

```

<parameter ordinal="3">
  <name>EMP_NAME_PREFIX</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpNamePrefix/text()/</expr>
</parameter>
<parameter ordinal="4">
  <name>EMP_FIRST_NAME</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpFirstName/text()/</expr>
</parameter>
<parameter ordinal="5">
  <name>EMP_MIDDLE_NAME</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpMiddleName/text()/</expr>
</parameter>
<parameter ordinal="6">
  <name>EMP_LAST_NAME</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpMiddleName/text()/</expr>
</parameter>
<parameter ordinal="7">
  <name>EMP_NAME_SUFFIX</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpNameSuffix/text()/</expr>
</parameter>
<parameter ordinal="8">
  <name>ADDRESS</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/Address/text()/</expr>
</parameter>
<parameter ordinal="9">
  <name>ADDRESS_2</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/AddressTwo/text()/</expr>
</parameter>
<parameter ordinal="10">
  <name>EMP_CITY_US</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpCityUS/text()/</expr>
</parameter>
<parameter ordinal="11">
  <name>EMP_STATE</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpState/text()/</expr>
</parameter>
<parameter ordinal="12">
  <name>EMP_ZIP_CODE</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpZipCode/text()/</expr>
</parameter>
<parameter ordinal="13">
  <name>EMP_COUNTRY_CODE</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmpCountryCode/text()/</expr>
</parameter>
</parameters>
</insert>

```

#### 4. Enter the Update element

Update any of the following information.

```
<update sql="UPDATE EMP_NAME_ADDRESS SET EMP_FIRST_NAME = ?, EMP_MIDDLE_NAME = ?,
EMP_LAST_NAME = ?,
ADDRESS = ?, ADDRESS_2 = ?, EMP_CITY_US = ?, EMP_STATE = ?, EMP_ZIP_CODE = ?,
EMP_NAME_SUFFIX = ?
WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER = ? AND EMP_NAME_CODE = 1">
  <parameters>
    <parameter ordinal="1">
      <name>EMP_FIRST_NAME</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpFirstName/text()/</expr>
    </parameter>
    <parameter ordinal="2">
      <name>EMP_MIDDLE_NAME</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpMiddleName/text()/</expr>
    </parameter>
    <parameter ordinal="3">
      <name>EMP_LAST_NAME</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpLastName/text()/</expr>
    </parameter>
    <parameter ordinal="4">
      <name>ADDRESS</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/Address/text()/</expr>
    </parameter>
    <parameter ordinal="5">
      <name>ADDRESS_2</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/AddressTwo/text()/</expr>
    </parameter>
    <parameter ordinal="6">
      <name>EMP_CITY_US</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpCityUS/text()/</expr>
    </parameter>
    <parameter ordinal="7">
      <name>EMP_STATE</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpState/text()/</expr>
    </parameter>
    <parameter ordinal="8">
      <name>EMP_ZIP_CODE</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpZipCode/text()/</expr>
    </parameter>
    <parameter ordinal="9">
      <name>EMP_NAME_SUFFIX</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/EmpNameSuffix/text()/</expr>
    </parameter>
    <parameter ordinal="10">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
      <expr>/ContactInfo/Contact/CompanyNumber/text()/</expr>
    </parameter>
  </parameters>
</update>
```

```

<parameter ordinal="11">
  <name>EMPLOYEE_NUMBER</name>
  <type>STRING</type>
  <expr>/ContactInfo/Contact/EmployeeNumber/text()</expr>
</parameter>
</parameters>
</update>

```

## 5. Enter the query element

Type your SQL-like statement specifying the columns you want retrieve. Remember that the query element has a sub-element: parameters.



*For practice, enter the query sql tag followed by: `"SELECT * FROM EMP_NAME_ADDRESS WHERE COMPANY_NUMBER = ? AND EMPLOYEE_NUMBER = ? AND EMP_NAME_CODE = 1"` within the query element.*

```

<query sql="SELECT * FROM EMP_NAME_ADDRESS WHERE COMPANY_NUMBER = ? AND
EMPLOYEE_NUMBER = ? AND
EMP_NAME_CODE = 1">
  <parameters>
    <parameter ordinal="1">
      <name>COMPANY_NUMBER</name>
      <type>STRING</type>
    </parameter>
    <parameter ordinal="2">
      <name>EMPLOYEE_NUMBER</name>
      <type>STRING</type>
    </parameter>
  </parameters>
</query>

```

## 6. Enter the root and element elements

The 'root' element provides the name of the destination root element and the name of the elements that will represent either Solution Series records or database rows.



*For practice, enter the root name as 'ContactInfo' and the rowName as 'Contact'.*

```

<root name="ContactInfo" rowName="Contact">
  <element name="CompanyNumber">
    <content>COMPANY_NUMBER</content>
  </element>
  <element name="EmployeeNumber">
    <content>EMPLOYEE_NUMBER</content>
  </element>
  <element name="EmpNameCode">
    <content>EMP_NAME_CODE</content>
  </element>
  <element name="EmpNamePrefix">
    <content>EMP_NAME_PREFIX</content>
  </element>
  <element name="EmpFirstName">
    <content>EMP_FIRST_NAME</content>
  </element>
  <element name="EmpMiddleName">
    <content>EMP_MIDDLE_NAME</content>
  </element>

```

```
</element>
<element name="EmpLastName">
  <content>EMP_LAST_NAME</content>
</element>
<element name="EmpNameSuffix">
  <content>EMP_NAME_SUFFIX</content>
</element>
<element name="Address">
  <content>ADDRESS</content>
</element>
<element name="AddressTwo">
  <content>ADDRESS_2</content>
</element>
<element name="EmpCityUS">
  <content>EMP_CITY_US</content>
</element>
<element name="EmpState">
  <content>EMP_STATE</content>
</element>
<element name="EmpZipCode">
  <content>EMP_ZIP_CODE</content>
</element>
<element name="EmpCountryCode">
  <content>EMP_COUNTRY_CODE</content>
</element>
</root>
```

### 7. Save your map file

Save your newly created map file with an XML extension. Remember, map files are case sensitive.



*For practice, save your xml map file as WriteNameAddressMap.xml.*

## Editing ecp-views.xml

Perform the following steps to edit ecp-views.xml:

- 1. Open the view xml file**

Open the eCyborg delivered ecp-views.xml file using a text editor.

- 2. Create a new entry**

Position the cursor on the line after the last occurrence of '</ecp.view>' and insert the tags for the names of the view, the mapping file, and the stylesheet to produce the file eCyborg returns to the client application.



*For practice, enter the view name as 'WriteNameAddress' and the map name as 'WriteNameAddressMap.xml'.*

Your ecp-views.xml file should look like the following example:

```
<ecp:view name="WriteNameAddress">
  <ecp:map>WriteNameAddressMap.xml</ecp:map>
  <ecp:xsl></ecp:xsl>
  <ecp:allow-write value='true'>
    <ecp:transform/>
  </ecp:allow-write>
</ecp:view>
```

- 3. Save and close the file**

## Performing a Write side test

Perform the following steps to test the 'Write' side connection:

**1. Access the Collaborative Platform Web Services**

To run the custom application, open a browser and access the following URL:

`http://hostname/ecp/read.htm`

Your web browser displays an HTML form (Schema Write Test Page) and its fields.

**2. Add your desired environment details**

Before submitting the form, ensure that your user name and password are correct and enter the desired view name you defined in the `ecp-views.xml` file:

`WriteNameAddress`

Add your employee and company information in the view query parameters fields.

**3. Submit the form**

Press the submit button to send the request to the Collaborative Platform web service.

**4. Review the results**

The SOAP response appears as XML in the browser.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header/>
  <soap-env:Body>
    <ContactInfo>
      <Contact>
        <CompanyNumber>999999</CompanyNumber>
        <EmployeeNumber>1001</EmployeeNumber>
        <EmpNameCode>1</EmpNameCode>
        <EmpNamePrefix>Ms</EmpNamePrefix>
        <EmpFirstName>june</EmpFirstName>
        <EmpMiddleName></EmpMiddleName>
        <EmpLastName>MEYER</EmpLastName>
        <EmpNameSuffix>Jr.</EmpNameSuffix>
        <Address>4444 MISTY LANE</Address>
        <AddressTwo>UNIT 55</AddressTwo>
        <EmpCityUS>EL SEGUNDO</EmpCityUS>
        <EmpState>IL</EmpState>
        <EmpZipCode>93101</EmpZipCode>
        <EmpCountryCode>USA</EmpCountryCode>
      </Contact>
    </ContactInfo>
  </soap-env:Body>
</soap-env:Envelope>
```

**5. Save the response as an XML file**

Remove all the SOAP tags and XML tags and save the file. Consistent with the example above, your output should look similar to the following:

```

<ContactInfo>
  <Contact>
    <CompanyName>999999</CompanyName>
    <EmployeeNumber>1001</EmployeeNumber>
    <EmpNameCode>1</EmpNameCode>
    <EmpNamePrefix>Ms</EmpNamePrefix>
    <EmpFirstName>june</EmpFirstName>
    <EmpMiddleName></EmpMiddleName>
    <EmpLastName>MEYER</EmpLastName>
    <EmpNameSuffix>Jr.</EmpNameSuffix>
    <Address>4444 MISTY LANE</Address>
    <AddressTwo>UNIT 55</AddressTwo>
    <EmpCityUS>EL SEGUNDO</EmpCityUS>
    <EmpState>IL</EmpState>
    <EmpZipCode>93101</EmpZipCode>
    <EmpCountryCode>USA</EmpCountryCode>
  </Contact>
</ContactInfo>

```

## 6. Make your updates to the file

Review the following for the example last name and address change:

```

<ContactInfo>
  <Contact>
    <CompanyName>999999</CompanyName>
    <EmployeeNumber>1001</EmployeeNumber>
    <EmpNameCode>1</EmpNameCode>
    <EmpNamePrefix>Ms</EmpNamePrefix>
    <EmpFirstName>june</EmpFirstName>
    <EmpMiddleName></EmpMiddleName>
    <EmpLastName>JONES</EmpLastName>
    <EmpNameSuffix>Jr.</EmpNameSuffix>
    <Address>1200 Wacker Dr</Address>
    <AddressTwo>UNIT 55</AddressTwo>
    <EmpCityUS>EL SEGUNDO</EmpCityUS>
    <EmpState>IL</EmpState>
    <EmpZipCode>93101</EmpZipCode>
    <EmpCountryCode>USA</EmpCountryCode>
  </Contact>
</ContactInfo>

```

## 7. Save the file

Save your new XML file to an appropriate directory.

## 8. Map to the following Schema Write Test Page

Access the following URL to update Solution Series:

<http://hostname/ecp/write.htm>

## 9. Enter required Write side details

Enter your username, password, view name, and browse to the XML data file you just saved in step 7.

**10. Press the Submit button**

Upon successful update you will see the following message:

```
<?xml version="1.0" encoding="UTF-8" ?>
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header />
  <soap-env:Body>
    <ecp:Acknowledgement xmlns:ecp="http://www.hewitt.com/eCyborg/ecp">
      <ecp:RecordsAffected>1</ecp:RecordsAffected>
    </ecp:Acknowledgement>
  </soap-env:Body>
</soap-env:Envelope>
```

PART 3

## Appendices

---

### In This Section

Collaborative Platform Included Solution Series Forms.....	85
Error handling and codes .....	87
eCyborg SQL Tool Syntax .....	91

A P P E N D I X A

# Collaborative Platform Included Solution Series Forms

---

## In This Appendix

Solution Series forms.....86

## Solution Series forms

The following is a list of Solution Series forms that are included with the delivered default.XML file:

Organization Name and Address	AA-SCR
Employee Information	EF-SCR
Payroll Home Location/Pay Allocations	GG-SCR
Employee Tax Record Maintenance	JJ-SCR
Additional Personal and ID Information	02-SCR
Emergency Contact/Physician	16-SCR
Company Personal Property	22-SCR
Driver License Information	25-SCR
Salary Assignment/Changes	40-SCR
Absences	93-SCR

A P P E N D I X B

## Error handling and codes

---

### In This Appendix

Error handling.....	88
---------------------	----

## Error handling

The JDBC client classes implement standard eCyborg SQL driver interfaces and throw exceptions of type `SQLException` in an error. The `SQLException` class constructor takes three parameters:

- String - error message
- String - SQL State
- Int - vendor code

The SQL driver only uses the error message and vendor code not the SQL state parameter. Java applications using the eCyborg SQL driver can access the error message and error code using the following methods:

- `getMessage()`
- `getErrorCode()`

Below is a list of error codes and a description of when the error might occur. The "-debug" option puts the SQL server into debug mode and the server creates the file "sqlserver.log" containing debug information. Typically, the "sqlserver.log" file contains limited information and messages are always appended. For a better explanation on bugs, you must use the "-trace" option. The Solution Series administrator maintains the "sqlserver.log" file and must archive it or delete it as necessary.

## Codes and explanation

The following is a list of Collaborative Platform error codes and their explanation:

Error Codes	Explanation
10000	The JDBC client is unable to connect to the JDBC server. An invalid connection property like 'host' or 'port' may cause a failure to connect. Also, the JDBC server may be unavailable or setup incorrectly.
10004	An open request did not contain an environment name. The connection to the JDBC server was successful but the JDBC server does not have an environment name and therefore cannot connect to CBSV and FILE01.
10005	An open request did not contain a Solution Series username. A username is required to open a connection to CBSV.
10006	An open request did not contain a Solution Series password. A password is required to open a connection to CBSV.
10007	The Solution Series username and password are invalid or have expired.

10008	An error occurred while attempting to establish a connection to CBSV. The error could be the result of invalid connection properties or the server is unavailable.
10009	An error occurred while attempting to establish a connection to CYBIO. This could be the result of invalid connection properties or the server is unavailable.
10010	The table name specified does not exist.
10014	An error occurred while loading file default.xml which is usually caused by an invalid XML file. Verify if the XML file is invalid by opening the default.xml in Internet Explorer.
10016	A communication error occurred while attempting to access FILE01 using program CYBIO. The error could be the result of network problems or the server was taken down during an active JDBC connection.
10017	An update or delete of a record failed because the record no longer exists. This error is rare because the eCyborg SQL driver will select records and update them in one pass. During the short processing period, it is unusual to have deleted a record selected for an update.
10018	An attempt has been made to insert a row that already exists.
10019	An error occurred while attempting to read an option list from FILE01.
10020	The Solution Series form does not exist. This error might occur if you did not enter a form name correctly in file default.xml.
10021	You do not have security rights to view the requested company or employee record.
10022	You do not have security rights to run the specified Solution Series form.
10023	The company number or employee number you specified does not exist.
10024	Form rejects occurred while you attempted to view or send data to a Solution Series form. The exception message will contain the Solution Series reject error.

10025	A communication error occurred while you attempted to access a Solution Series form. The error could be the result of network problems or the server was taken down during an active JDBC connection.
10030	An error occurred because you attempted to insert or update an option list field by description and the description does not exist. When updating fields by description, Solution Series makes a reverse lookup to find the code value by reading each option list record matching the description.
10031	An error occurred because you attempted to insert or update a position admin field by description (job, position or org unit) and it does not exist.
10032	An error occurred because you attempted to update a table (usually a virtual table) that does not support updating. This type of table is read-only and any attempt to delete, update, or insert records will result in this error.
20000	A syntax error raised by the SQL parser usually caused by a problem with the SQL statement but it could be any number of reasons. For example, the syntax could be incorrect or a table or column name is invalid.

APPENDIX C

## eCyborg SQL Tool Syntax

---

### In This Appendix

SELECT statement .....	92
INSERT statement .....	93
UPDATE statement .....	94
GROUP BY clause .....	95
ORDER BY clause .....	96
TOP keyword.....	97
CONTAINS function.....	98
String function .....	99
Date function .....	100
WHERE clause .....	101

## SELECT statement

The SELECT statement retrieves data from a table.

**Basic usage:**

```
SELECT <FIELD_LIST> FROM <TABLE_NAME>
```

TABLE\_NAME is the name of the table. FIELD\_LIST is a comma separated list of field names, or a wildcard "\*" character which selects all the fields. Fields can be given a different name in the results using the AS keyword. Table names, field names and keywords are case insensitive.

**Example:**

The following example selects all the fields from the table EMP\_BASIC\_DETAILS:

- ```
SELECT * FROM EMP_BASIC_DETAILS
```

The following example selects the named fields from the table:

- ```
SELECT FIRST_NAME, MIDDLE_NAME, LAST_NAME FROM  
EMP_BASIC_DETAILS
```

The following example selects the same fields as the previous example but gives each of them a different name in the results:

- ```
SELECT FIRST_NAME AS first, MIDDLE_NAME AS middle,  
LAST_NAME AS last FROM EMP_BASIC_DETAILS
```

## INSERT statement

The SQL INSERT statement allows the SQL driver to add a new record to a table.

**Example:**

In the following example, TABLE\_NAME is the name of the table and DATA is a comma separated list of values entered into the new row. A value must be provided for each field, except for read only fields, which must not be included:

- INSERT INTO <TABLE\_NAME> VALUES ( <DATA> )

For date fields, the current date can be inserted using either of the keywords TODAY or GETDATE, and null values can be inserted using NULL or UNDEFINED.

## UPDATE statement

The UPDATE statement is used to change one or more fields in one or more rows.

**Example:**

In the following example, TABLE\_NAME is the name of the table:

- UPDATE <TABLE\_NAME> SET <DATA> [ WHERE <CONDITIONS> ]

DATA is a comma separated list of field name and value pairs which specifies the fields to be updated:

- FIELD1=VALUE1, FIELD2=VALUE2 ...

Key fields cannot be updated. CONDITIONS is a list of conditions that specify which rows are to be updated. The format of the WHERE clause is the same as for the SELECT statement. If no WHERE clause is specified, all rows are updated. Date fields can be set to the current date using the either of the keywords TODAY or GETDATE, and can be set to null using NULL or UNDEFINED.

## GROUP BY clause

Use the GROUP BY clause to group the results of a query based on the value of a specific field or list of fields. It must be the last part of the statement and you cannot use it in conjunction with ORDER BY.

eCyborg represents each group created by a GROUP BY clause as one result row. If more than one field is specified in the GROUP BY clause, eCyborg forms the groups by rows with the same specified fields. If a row contains a null value in any of the specified fields, it will form its own group.

If eCyborg uses a GROUP BY clause, only the fields grouped can be selected. The other fields may have different values for different members of the same group. For numeric fields, whether included in the GROUP BY clause or not, a sum may be calculated and included as a field in the results using the SUM keyword. You can use the keywords MIN, MAX and AVG to include the minimum, maximum, and average values of a column.

You can use the keyword COUNT to give a count of how many rows in a group have a non-NULL value in a particular column, or if used with a wildcard "\*", how many rows made up the group. You can give these extra fields an alias name using the AS keyword. By using a HAVING clause, you can restrict the groups returned to those matching one or more conditions.

The syntax for the conditions is exactly the same as for the WHERE clause. However, the conditions can only refer to the visible columns in the grouped results. That is, the column used for grouping and the columns produced by SUM and COUNT must be referred to by the names they are given in the grouped results, not the original column names.

### Examples:

The following are scenarios and explanations:

- `SELECT CITY, COUNT(*) AS NUMBER FROM EMP_BASIC_DETAILS [WHERE Clause] GROUP BY CITY`  
The statement returns a row for each city. Each row has a CITY field and a NUMBER field which is a count of the number of employees in that city.
- `SELECT <FIELD1>, SUM(<FIELD2>), AVG(FIELD3), COUNT(*) FROM <TABLE> GROUP BY FIELD1`  
The statement groups the results by the value of FIELD1 and includes in the results for each group the sum of the values of FIELD2 and the average of the values of FIELD3 and a count of the number of rows that made up each group.
- `SELECT <FIELD1>, SUM(<FIELD2>) AS sum_2, AVG(FIELD3) AS avg_3, COUNT(*) AS total FROM <TABLE> GROUP BY FIELD1 HAVING total > 3`  
The statement groups the results the same way as the previous example but gives each result column a name and only selects the groups that were made up from more than three rows.

## ORDER BY clause

eCyborg uses the ORDER BY clause to sort the results of a query based on the value of a specified field or list of fields. It must be the last part of the statement.

By default the rows are sorted in increasing order but you can use the keyword DESC to sort in descending order. The sort only applies to the field that it follows, so it is possible to mix ascending and descending order when ordering by more than one field.

### Examples:

The following are scenarios and explanations:

- `SELECT * FROM EMP_BASIC_DETAILS [WHERE clause] ORDER BY LAST_NAME`  
The statement orders the results according to the value of the LAST\_NAME field.
- `SELECT * FROM EMP_BASIC_DETAILS [WHERE clause] ORDER BY LAST_NAME, FIRST_NAME`  
The statement is the same as above, but rows where the LAST\_NAME field is identical will be further sorted by the value of FIRST\_NAME.
- `ORDER BY <FIELD1> DESC, <FIELD2>`  
The results of this statement orders the results by FIELD1 in descending order, with rows having the same value of FIELD1 further sorted by FIELD2 in ascending order.

## TOP keyword

You use the TOP keyword to limit the number of rows returned by a query. It must immediately follow the SELECT keyword.

**Example:**

The following example selects the first three rows from the table:

- `SELECT TOP 3 * FROM EMP_BASIC_DETAILS`

## CONTAINS function

The CONTAINS function performs full-text search matches in the WHERE clause. The matches are not case sensitive and CONTAINS can only be used on string fields. The following is the general form of the CONTAINS function:

- CONTAINS(<FIELD>, '<SEARCH\_CONDITIONS>')

The FIELD must be the name of a string field and SEARCH\_CONDITIONS can take one of two forms:

- <SEARCH\_TERM> [<COMBINE\_TERM> <SEARCH\_TERM> ...]

If SEARCH\_TERM contains spaces, you must use double quotes. If SEARCH\_TERM ends with a wildcard (\*) then it matches if it matches the start of a word. If it contains spaces and ends with a wildcard, then it matches if each part matches the start of successive words. The COMBINE\_TERM function can be any of AND, OR, AND NOT, NEAR.

You can use the tilde character (~) as an alias for NEAR. A NEAR condition matches if both search terms are found and there are less than 50 characters between them. An alternative number of characters can be specified within brackets: NEAR(20) gives a maximum distance of 20 characters. Brackets may be used to group the search terms with the exception that NEAR cannot be applied to a group.

### Examples:

The following examples provide scenarios and explanations:

- CONTAINS(FIRST\_NAME, 'john OR fred\*')  
The expression matches 'John' or 'Fred' or 'Frederick' but not 'Johnny' or 'Alfred'.
- CONTAINS(FULL\_NAME, 'john AND NOT (smith OR jones)')  
The expression matches any name containing 'John' and not containing 'Smith' or 'Jones'.
- CONTAINS(FULL\_NAME, '"paul gray\*")'  
The expression matches 'Paul Gray', 'Pauline Grayson' etc.
- ISABOUT(<SEARCH\_TERM> [, <SEARCH\_TERM> ...])  
The second form uses the ISABOUT keyword
- SEARCH\_TERM is the same as in the first form.  
If there is more than one SEARCH\_TERM, a row is considered to match if any one of the terms matches.

### Example:

The following examples provide scenarios and explanations:

- CONTAINS(CITY, 'ISABOUT(Chicago, "San Diego", Detroit)')  
The expression matches any of the named cities.

## String function

You can use the following functions in the WHERE clause to modify or return information about a string. In each case, the FIELD argument must be either the name of a string field or a literal string.

**UPPER(<FIELD>)**

The expression returns the value of FIELD converted to upper case.

**LOWER(<FIELD>)**

The expression returns the value of FIELD converted to lower case.

**CHAR\_LENGTH(<FIELD>)**

**CHARACTER\_LENGTH(<FIELD>)**

These expressions return the length of the value of FIELD.

**POSITION(<STRING> IN <FIELD>)**

The expression returns the position of the start of STRING in the value of FIELD, counting the first character as position 1. If the value of FIELD does not contain STRING, it returns 0.

**SUBSTRING(<FIELD> FROM <START> [FOR <LENGTH>])**

This returns the substring of the value of FIELD, starting from START, with a length of LENGTH. If LENGTH is not given, or if the sum of START plus LENGTH is greater than the length of the value of FIELD, the statement returns the substring from START to the end of the value of FIELD. The first character is counted as position 1.

**TRIM([LEADING|TRAILING|BOTH] [CHARACTER] FROM <FIELD>)**

This trims CHARACTER from the leading, trailing, or both ends of the value of FIELD.

If CHARACTER is not specified, the space character is used.

If LEADING, TRAILING, or BOTH are not specified, both ends are trimmed.

## Date function

You can use the following functions in the WHERE clause to modify or return a specific part of a date. In each case, FIELD must be either the name of a date field or a literal date.

**YEAR(<FIELD>)**

The expression returns the year of the given date as a number.

**MONTH(<FIELD>)**

The expression returns the month of the given date as a number.

**DAY(<FIELD>)**

The expression returns the day of month of the given date as a number.

**DATEADD(<PART>, <AMOUNT>, <FIELD>)**

This adds AMOUNT to the specified PART of a date. PART must be one of 'YEAR', 'MONTH' or 'DAY'. The AMOUNT must be a literal number and the FIELD must be the name of a date field or a literal date.

## WHERE clause

The WHERE clause qualifies the rows (records) returned by creating one or more conditions and it must precede the table name. The following is the general form of a condition:

- <FIELD> <OPERATOR> <VALUE>

Conditions are separated by either of the keywords AND or OR and may be grouped by brackets. Place the keyword NOT before the condition or group of conditions to establish the opposite of the condition or group.

| Example                                                                     | Description                                                                                                                                                                                                          |
|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SELECT * FROM <TABLE> WHERE <CONDITION>                                     | A single condition must be met.                                                                                                                                                                                      |
| SELECT * FROM <TABLE> WHERE <CONDITION1> AND <CONDITION2>                   | Two conditions joined by a logical AND must be met.                                                                                                                                                                  |
| SELECT * FROM <TABLE> WHERE <CONDITION1> AND (CONDITION2> OR < CONDITION3>) | A group of conditions joined by a logical OR that is joined to a single condition by a logical AND must be met.<br><br>This will select rows where CONDITION 1 and either or both CONDITION2 and CONDITION3 are true |
| SELECT * FROM <TABLE> WHERE NOT (<CONDITION1> AND <CONDITION2>)             | A group of two conditions with its meaning reversed by use of the NOT keyword must be met.<br><br>This statement will select all rows except where both conditions are true.                                         |

Typically, FIELD is the name of a field and VALUE is a literal value, but both may be a field or either may be an expression.

### Literal values

Identify literal string values by using single quotes. Literal date values must be in the following SQL date format:

- DATE 'yyyy-mm-dd' or {d 'yyyy-mm-dd'}

You can use the words 'TODAY' or 'GETDATE' to refer to the current date.

### OPERATORS

The following table lists the valid OPERATORS in a condition:

| Operator | Description                                                                                                                                                                          |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| =        | Equal To                                                                                                                                                                             |
| !=       | Not Equal To                                                                                                                                                                         |
| <>       | Not Equal To                                                                                                                                                                         |
| >        | Greater Than                                                                                                                                                                         |
| <        | Less Than                                                                                                                                                                            |
| >=       | Greater Than or Equal To                                                                                                                                                             |
| <=       | Less Than or Equal To                                                                                                                                                                |
| GREATEST | Matches the row(s) where FIELD is greatest. VALUE is not used with this operator.                                                                                                    |
| >>       | An alias for GREATEST                                                                                                                                                                |
| LEAST    | Matches the row(s) where FIELD is least. VALUE is not used with this operator                                                                                                        |
| <<       | An alias for LEAST                                                                                                                                                                   |
| NEAREST  | Matches the row(s) with the greatest value of FIELD that is less than VALUE                                                                                                          |
| ~=       | An alias for NEAREST                                                                                                                                                                 |
| LIKE     | For string fields, VALUE is treated as a pattern where '_' matches any character and '%' matches zero or more of any character.                                                      |
| IN       | VALUE is a comma-separated list of values enclosed in brackets '()'. Rows matching any of the listed values are returned. FIELD must be a field name.                                |
| BETWEEN  | VALUE is a lower and upper value separated by AND. Rows where FIELD is between the two values (inclusive) are returned. The lower value must come first. FIELD must be a field name. |
| IS NULL  | Matches rows where FIELD contains a null value. FIELD must be a field name. VALUE is not used with this operator.                                                                    |

| Examples                                                          | Description                                                                                            |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| SELECT * FROM <TABLE> WHERE <NUMERIC_FIELD> = 100                 | This statement selects only rows where NUMERIC_FIELD is equal to 100.                                  |
| SELECT * FROM <TABLE> WHERE <STRING_FIELD> LIKE 'a%'              | This statement selects only rows where STRING_FIELD begins with an 'a'.                                |
| SELECT * FROM <TABLE> WHERE <FIELD> GREATEST                      | This statement selects only the row(s) where FIELD has the greatest value.                             |
| SELECT * FROM <TABLE> WHERE <DATE_FIELD> NEAREST {d '2002-08-16'} | Only the row(s) with the latest value of DATE_FIELD that is still earlier than 2002-08-16 is selected. |
| SELECT * FROM <TABLE> WHERE <FIELD1> = <FIELD2>                   | This Select statement returns only rows where FIELD1 and FIELD2 have the same value.                   |

| Examples                                                        | Description                                                                                       |
|-----------------------------------------------------------------|---------------------------------------------------------------------------------------------------|
| SELECT * FROM <TABLE> WHERE <NUMERIC_FIELD> IN (100, 120, 160)  | This statement selects only rows where NUMERIC_FIELD has one of the given values.                 |
| SELECT * FROM <TABLE> WHERE <NUMERIC_FIELD> BETWEEN 100 AND 200 | This statement selects only rows where NUMERIC_FIELD has a value between 100 and 200 (inclusive). |

**FIELD or VALUE**

Either FIELD or VALUE may be an expression instead of a single value. If you use numeric values, the mathematical operators '+', '-', '\*' and '/' may be used grouped inside brackets '()'. For date values, the only valid mathematical operation is to add or subtract a numeric value. For string values, you can use the '+' operator to concatenate two strings (either literals or the values of string fields).

| Example                                              | Description                                                                                                              |
|------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------|
| WHERE <NUMERIC_FIELD1> > 3 * (<NUMERIC_FIELD2> - 10) | Only rows where FIELD1 is greater than 3 times FIELD2 minus are selected.                                                |
| WHERE <DATE_FIELD> BETWEEN TODAY - 3*365 and TODAY   | Only rows where the DATE_FIELD value is between a range of today's date minus 3 times 365 and today's date are selected. |

**Use of COMPANY\_NUMBER and EMPLOYEE\_NUMBER**

If the WHERE clause uses the COMPANY\_NUMBER field, it must be the first condition and the operator must be '='. If the WHERE clause uses the EMPLOYEE\_NUMBER field, it must use the '=' operator and it must be preceded by a COMPANY\_NUMBER condition. Also, the two conditions must be joined by AND.

If neither COMPANY\_NUMBER nor EMPLOYEE\_NUMBER are specified in the WHERE clause, the query will be run for the employees in the delivered 999999 test company. For example:

- WHERE COMPANY\_NUMBER = '999999' AND EMPLOYEE\_NUMBER = '1001'

SQL access to some Solution Series forms requires only one parameter, COMPANY\_NUMBER, in forms such as the Flex Master/Group Plan Components form (TP-SCR) or the Flex Benefits Master Plan Rules form (TKFSCR). The SQL query will not work if both parameters—company and employee numbers—are specified for non-employee related forms.



Refer to eCyborg SQL Tool Syntax (on page 91) to learn how to build more complex statements.

## Glossary of Terms

---

### **Data store**

Repository where data is stored. Client applications request information from a Solution Series data store.

### **HR-XML**

The HR-XML Consortium is an independent, non-profit organization dedicated to the development and promotion of a standard suite of XML specifications to enable e-business and the automation of human resources-related data exchanges.

### **SOAP**

Simple Object Access Protocol. This is an XML-based protocol that exchanges structured and typed information on the Web.

### **SQL**

Structured Query Language retrieves information from a relational database so it can populate Impromptu reports.

### **XML**

Extensible Mark-up Language is a language for documents containing structured information.

### **XML Schema Definition Language**

A language for describing and constraining the content of XML documents.

### **XPath**

A syntax used to describe parts of an XML document.

### **XSLT**

Extensible Stylesheet Language Transformations. A language for transforming XML documents into other XML documents.



# Index

---

- A**
- Adding field elements to the table definition file 55, 72
  - Adding Solution Series records ..... 37
  - Appendices ..... 83
  - Audience ..... 4
  - Authentication ..... 9
- B**
- Basic usage ..... 28
- C**
- Codes and explanation ..... 88
  - Collaborative Platform Included Solution Series
    - Forms ..... 85
  - Command line ..... 26
  - CONTAINS function ..... 98
  - Creating
    - Collaborative Schema ..... 20
    - Creating a schema - Read side ..... 49
    - Creating a schema - Write side ..... 65
    - Creating a stylesheet ..... 60
    - Creating Collaborative Schema ..... 47
    - Creating the map file ..... 57, 74
- D**
- Data mapping ..... 10
  - Data sources ..... 21
  - Data store ..... 105
  - Database information ..... 36
  - Date function ..... 100
  - Default.xml ..... 21
  - Defining a Web Service View ..... 41
- E**
- eCyborg Collaborative Platform ..... 8
  - eCyborg SQL Tool Syntax ..... 91
  - Editing ecp-views.xml ..... 63, 79
  - elements of the mapping file ..... 10
  - Error content defined ..... 44
  - Error handling ..... 88
  - Error handling and codes ..... 87
  - Exit codes ..... 30
- F**
- Field alias element ..... 23
  - Field elements ..... 22
- G**
- GROUP BY clause ..... 95
- H**
- How to get additional help ..... 5
  - HR-XML ..... 105
  - HR-XML consortium ..... 9
- I**
- Identifying the data source ..... 51, 67
  - INSERT statement ..... 93
  - Interactive mode ..... 27
  - Interactive usage ..... 28
  - Introduction ..... 1, 3, 20, 50, 66
  - Invoking the Web Service ..... 42
- L**
- List fields ..... 28
  - List fields - non-interactive usage ..... 30
  - List tables ..... 28
  - List tables - non-interactive usage ..... 30
- M**
- map file ..... 10

|                                             |        |                                                         |        |
|---------------------------------------------|--------|---------------------------------------------------------|--------|
| Map option .....                            | 23     | Test harness.....                                       | 16     |
| Mapping file example.....                   | 11     | Test the Read functionality .....                       | 33     |
| Mapping file output example .....           | 13     | Test the Write functionality .....                      | 45     |
| Modifying Solution Series records .....     | 38     | Testing the SQL statement.....                          | 54, 71 |
| Modifying the table definition file .....   | 52, 68 | Testing the SQL statement - non-interactive usage ..... | 29     |
| <b>N</b>                                    |        | Testing your SQL statement.....                         | 29     |
| Non-interactive usage.....                  | 29     | TOP keyword.....                                        | 97     |
| <b>O</b>                                    |        | Transforming an incoming XML document .....             | 40     |
| ORDER BY clause.....                        | 96     | <b>U</b>                                                |        |
| <b>P</b>                                    |        | Unsupported forms.....                                  | 14     |
| Performing a Read side test.....            | 64     | UPDATE statement.....                                   | 94     |
| Performing a Write side test.....           | 80     | Using XSLT .....                                        | 13     |
| Processing the request .....                | 42     | <b>V</b>                                                |        |
| <b>Q</b>                                    |        | Validate the SQL statement .....                        | 25     |
| Query map file.....                         | 31     | View definition file .....                              | 14     |
| <b>R</b>                                    |        | <b>W</b>                                                |        |
| Read functionality .....                    | 19     | Welcome .....                                           | 4      |
| Read side functionality .....               | 16     | What is a schema?.....                                  | 9      |
| Receiving a request .....                   | 15     | WHERE clause .....                                      | 101    |
| Request process .....                       | 15     | Working with the Collaborative Platform.....            | 7      |
| <b>S</b>                                    |        | Write functionality .....                               | 35     |
| schemas .....                               | 10     | Write side functionality.....                           | 17     |
| SELECT statement.....                       | 92     | <b>X</b>                                                |        |
| SOAP.....                                   | 105    | XML.....                                                | 105    |
| SOAP and XML.....                           | 9      | XML and the Collaborative Platform .....                | 10     |
| Solution Series forms .....                 | 86     | XML document transformation.....                        | 13     |
| SQL .....                                   | 105    | XML Schema Definition Language .....                    | 105    |
| SQL-like syntax.....                        | 27     | XPath .....                                             | 105    |
| Starting the eCyborg SQL tool.....          | 53, 70 | XSLT .....                                              | 105    |
| Starting the SQL tool.....                  | 25     | XSLT applied.....                                       | 40     |
| String function.....                        | 99     |                                                         |        |
| Structure of the incoming XML document..... | 39     |                                                         |        |
| Structure of the mapping file.....          | 10, 36 |                                                         |        |
| Suggestions and feedback .....              | 5      |                                                         |        |
| <b>T</b>                                    |        |                                                         |        |
| Table definition .....                      | 21     |                                                         |        |