

Guide  
June 1996



WORLD SOFTWARE

# C.A.S.E Computer Aided Software Engineering

Release  
A7.3

JD Edwards®



Item # A73CEACS



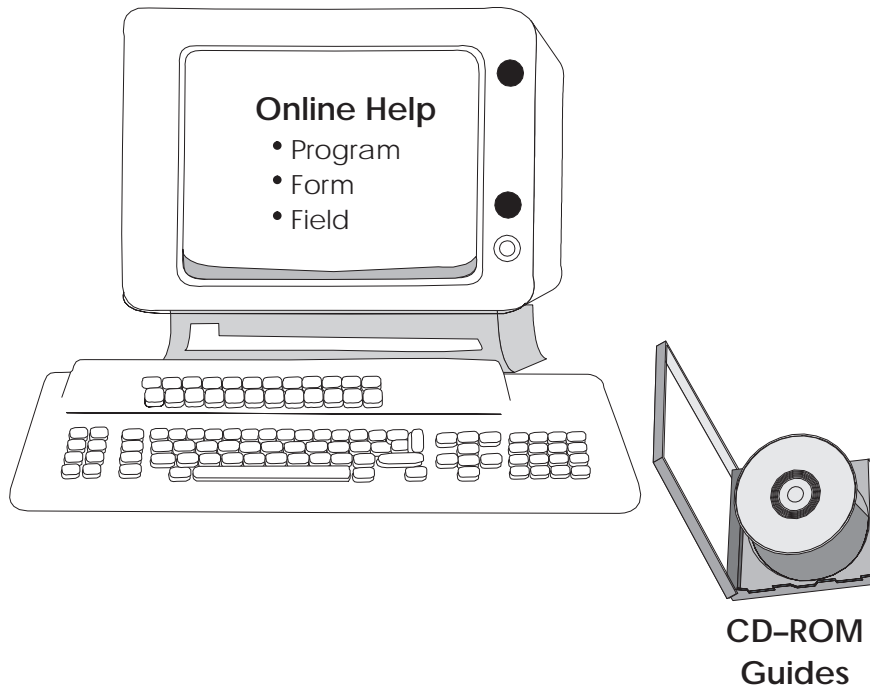
J.D. Edwards & Company  
8055 E. Tufts Avenue  
Denver, CO 80237

©J.D. Edwards & Company 1995, 1996  
All Rights Reserved

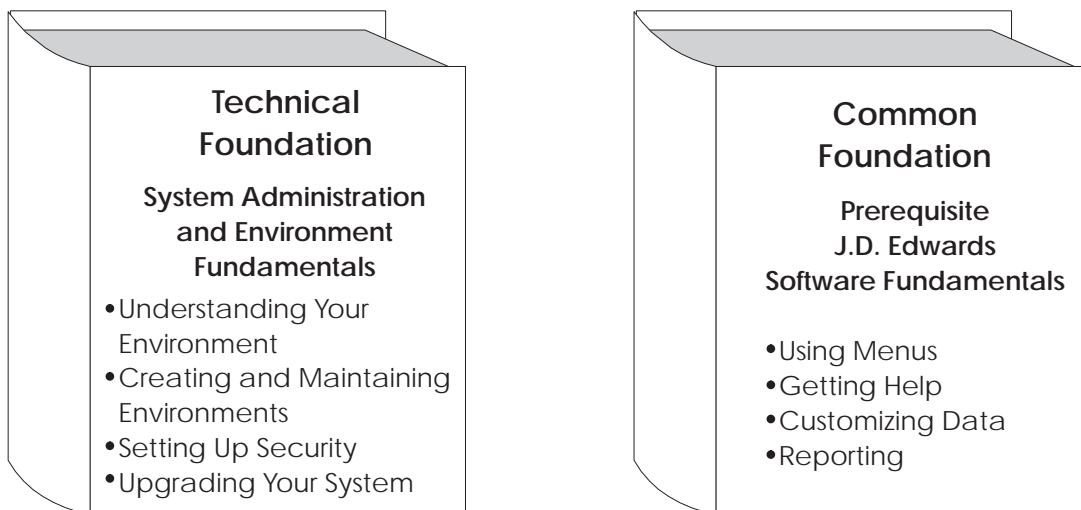
The information in this guide is confidential and a proprietary trade secret of J.D. Edwards & Company. It may not be copied, distributed, or disclosed without prior written permission. This guide is subject to change without notice and does not represent a commitment on the part of J.D. Edwards & Company. The software described in this guide is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. J.D. Edwards & Company uses an automatic software disabling routine to monitor the license agreement. For more details about this routine, please refer to the *Technical Foundation Guide*.



## Where Do I Look?



## Guides





## **Important Note for Students in Training Classes**

This guide is a source book for online helps, training classes, and user reference. Training classes may not cover all the topics contained here.





# Welcome

---

## About this Guide

This guide provides overviews, illustrations, procedures, and examples for release A7.3 of J.D. Edwards software. Forms (screens and windows) shown are only examples. If your company operates at a different software level, you might find discrepancies between what is shown in this guide and what you see on your screen.

This guide includes examples to help you understand how to use the system. You can access all of the information about a task using either the guide or the online help.

Before using this guide, you should have a fundamental understanding of the system, user defined codes, and category codes. You should also know how to:

- Use the menus
- Enter information in fields
- Add, change, and delete information
- Create and run report versions
- Access online documentation

## Audience

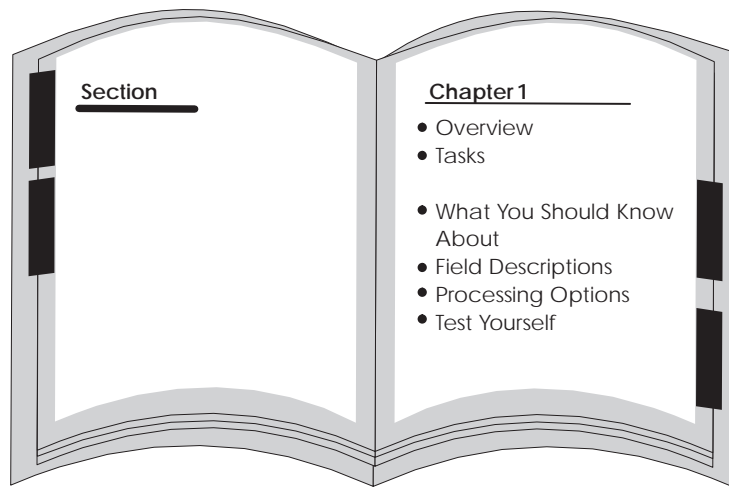
This guide is intended primarily for the following audiences:

- Users
- Classroom instructors
- Client Services personnel
- Consultants and implementation team members

## Organization

This guide is divided into sections for each major function. Sections contain chapters for each task or group of related tasks. Each chapter contains the information you need to accomplish the task, run the program, or print the

report. Chapters normally include an overview, form or report samples, and procedures.



When it is appropriate, chapters also might explain automatic accounting instructions, processing options, and warnings or error situations. Some chapters include self-tests for your use outside the classroom.

This guide has a detailed table of contents and an index to help you locate information quickly.

## Conventions Used in this Guide

The following terms have specific meanings when used in this guide:

- *Form* refers to a screen or a window.
- *Table* generally means “file.”

We assume an “implied completion” at the end of a series of steps. That is, to complete the procedure described in the series of steps, either press Enter or click OK, except where noted.

# Table of Contents

---

J.D. Edwards Overview .....	1-1
Signing On and Off .....	1-1
CASE Overview .....	1-5
System Integration .....	1-5
Specifications .....	1-6
Fundamentals .....	1-6
History of the Program Generator .....	1-6
Evolution of the Program Generator .....	1-6
Features .....	1-7
What are the Benefits of CASE? .....	1-8
Terms and Concepts .....	1-9
CASE, as an industry term .....	1-9
CASE, as a J.D. Edwards term .....	1-9
Detailed Information .....	1-9
About The Program Generator .....	1-10
About Program Types .....	1-11
About Master Source Code .....	1-11
About Program Specifications .....	1-11
Program Types .....	1-12
Library Naming Conventions .....	1-13
Menu Overview .....	1-14
J.D. Edwards Training Environment .....	1-16
The Student Library Setup .....	1-16
Signon Naming Conventions .....	1-16

## Foundation

About Foundation Information .....	2-1
Work with J.D. Edwards Provided Prerequisites .....	2-3
What Are the J.D. Edwards Provided Prerequisites? .....	2-4
Program Generator Files .....	2-4
Common User Defined Codes .....	2-5
File Server Copy Members .....	2-8
Source Code for Copy Modules .....	2-8
Source Code for J.D. Edwards Files .....	2-8
Work with User Provided Prerequisites .....	2-9
What are the User Provided Prerequisites? .....	2-10
Development Libraries .....	2-10
Creating the Multi-member Source File (JDESRC) .....	2-10
Job Queues .....	2-12
Project Management .....	2-13
What are CASE Profiles? .....	2-14
Accessing CASE Profiles .....	2-15
Object Authorities .....	2-21

## Program Generator

About Program Generator Steps .....	3-1
Define Program Generator Specifications .....	3-3
Accessing the Program Generator .....	3-4
Defining Program Generator Specifications .....	3-5
What are The Function Key Exits? .....	3-6
Define Program Purpose and Type .....	3-9
What Are the Function Key Exits? .....	3-17
Work with File Specifications .....	3-19
Accessing File Specifications .....	3-20
Understanding the File Specifications Form .....	3-21
What Are File Specifications? .....	3-23
Processing File Specifications .....	3-25
What Are the Function Key Exits? .....	3-26
Generating Source from Specifications .....	3-28
Define General Instructions .....	3-31
Accessing Define General Instructions .....	3-32
About the Edit Screen .....	3-33
About Special Characters .....	3-34
Special Characters for General Instructions .....	3-34
Special Characters within Help Instructions .....	3-35
Updating the Help File .....	3-36
Define Option and Function Key Exits .....	3-39
About Option and Function Key Exits .....	3-39
Defining Option and Function Key Exits .....	3-40
What Are the Function Key Exits? .....	3-42
Passing Parameters .....	3-42
Work with the Detailed Programming Facility .....	3-43
About the Detailed Programming Facility Form .....	3-45
What Are the Selection Exits? .....	3-46
What Are the Function Key Exits? .....	3-46
F6 – Repository Services .....	3-46
F10 – Select *PROC Fields On/Off .....	3-46
About Full Data Field Parameters .....	3-47
What Are the Primary Uses? .....	3-47
Accessing Full Data Field Parameters .....	3-48
What Are the Function Key Exits? .....	3-50
Loading VC0 Description Fields .....	3-51
Example – User Defined Code .....	3-53
Enabling the Database Update Function for Subfiles .....	3-54
Creating *ENTRY PLIST Entries .....	3-55
Protecting Fields from Being Cleared .....	3-57
Disabling Data Dictionary Edits .....	3-59
Creating a Partial KLIST for a File .....	3-60
Define Processing Options .....	3-61
Processing Options Setup Screen .....	3-63
What Are the Function Key Exits? .....	3-64
Creating Code for Processing Options .....	3-65
Example – Programs Using Processing Options .....	3-67

## Program Design Language

About Program Design Language .....	4-1
Work with Data Item Formula Revisions .....	4-3
Accessing Data Item Formula Revisions .....	4-3
Understanding the Data Item Formula Revisions Form .....	4-4
Understand PDL Statements and Syntax .....	4-5
Understanding PDL Statements .....	4-5
Understanding Blocks of Statements .....	4-8
Understanding Comments .....	4-9
Understanding Assignments .....	4-10
Understanding Database Operations .....	4-11
Understanding Calls .....	4-13
Understanding Loops .....	4-14
Understanding Conditions .....	4-15
Understanding Miscellaneous Keywords and Syntax .....	4-18
Understand Additional PDL Operations .....	4-21
Examples – PDL .....	4-22
Function Key Exits from PDL .....	4-25

## Source Modifications

About Source Modifications .....	5-1
Change Generated Source Code .....	5-3
About Changing Generated Source .....	5-3
Changing Generated Source .....	5-4
Regenerate Source Code .....	5-7
When to Regenerate Source Code .....	5-7
Changing CAP Status .....	5-8
Solving Generation Problems .....	5-10
Work with Model Control Language Programs .....	5-11
Copying a Model CL .....	5-12
Customizing a CL Model .....	5-13
J.D. Edwards Model CL Programs .....	5-14

## CASE Programs

About Creating CASE Programs .....	6-1
Create Subfile Inquiry Programs .....	6-3
Create Subfile Maintenance Programs .....	6-5
Create Report Programs .....	6-7
RDA Special Use Fields .....	6-8
Creating a Total Format .....	6-9
Defining a Subheading .....	6-14
DREAM Writer Considerations .....	6-16

## Additional Tools

About Additional Tools .....	7-1
Work with Quick Start CL Generator .....	7-3
About the Quick Start CL Generator .....	7-3
Accessing the Quick Start CL Generator .....	7-4
Compiling a CL Program .....	7-6
Work with the Quick Start Application Tool .....	7-7
About the Quick Start Application Tool .....	7-7
Steps of Quick Start .....	7-8
Selecting the Quick Start Application Definition .....	7-9
Selecting Data Fields .....	7-11
Accessing the Screen or Report You are Creating .....	7-12
Compiling the Screen or Report .....	7-13
Changing the Program Specifications .....	7-14
Submitting the Program to Compile .....	7-15
Accessing the Data Dictionary Glossary .....	7-16
Updating the Glossary .....	7-17
Completing Application Generation .....	7-18
Work with Action Diagramming .....	7-21
About Action Diagramming .....	7-21
Building an Action Diagram .....	7-22
Viewing an Action Diagram .....	7-23
What Are the Function Key Exits? .....	7-24
What Are the Cursor Sensitive Function Key Exits? .....	7-26
What Are the Selection Exits? .....	7-27
Accessing Logic Translation Feature .....	7-28

## Source Inventory and Database

About Using the Source Code Inventory and Database .....	8-1
Understand the Source Sequence Line Numbers .....	8-3
Create or Modify Program Types .....	8-7
About Creating or Modifying Program Types .....	8-7
Creating or Modifying Program Types .....	8-7
Create or Modify Logic Modules .....	8-13
About Logic Modules .....	8-13
What Are Primary Logic Modules? .....	8-13
What Are Detail Logic Modules? .....	8-14
Creating or Modifying Logic Modules .....	8-15
Accessing the Logic Module Index .....	8-16
Using Logic Module Cross Reference .....	8-17
Using Logic Module Op Codes .....	8-19
Maintaining the Logic Module File .....	8-20
Resequence Logic Module .....	8-20
Remove Logic Module .....	8-20
Creating or Modifying Formula Library Entry .....	8-21
Creating or Modifying Parameter Copy/Move .....	8-22
Printing Program Generator Specifications .....	8-23
Reviewing Source Modifications .....	8-24

Using Program Generator Updates .....	8-25
Using CASE Specifications Inquiry .....	8-26
Generation Options .....	8-27
Help Instructions Edit/Build .....	8-27
All Help Instructions .....	8-27
Global Program Regeneration .....	8-27
Understand Directives .....	8-29
Functional Directives .....	8-29
Substitution Directives .....	8-37
Exception Directives .....	8-39
Conditional Directives .....	8-40
Work with the Question and Answer System .....	8-43
Menu G9364, Option 3 — Simple Question & Answer .....	8-45
Reviewing Questions .....	8-45
Adding New Q & A Dialogue .....	8-46
Inquiring on a Dialogue .....	8-49
Changing a Dialogue .....	8-51
Copying a Dialogue .....	8-52
Rename a Dialogue .....	8-53
Running a Dialogue .....	8-54
Deleting a Dialogue .....	8-56
Running a Quiz .....	8-56
Create User Defined PDL .....	8-59
About User Defined PDL .....	8-59
Creating a User Defined PDL .....	8-60

## Appendices

Appendix A – Program Generator Checklist .....	A-1
Appendix B – Programming Standards .....	B-1
Appendix C – CASE Program Types .....	C-1
Appendix D – Source Listings .....	D-1
Appendix E – J.D. Edwards Subroutines and Flows .....	E-1
Appendix F – Putting It All Together .....	F-1
Appendix G – Functional Servers .....	G-1

## Glossary

## Index

## Exercises





# J.D. Edwards Overview

---

## Signing On and Off

```
Sign On

      System . . . . . :   JD ED
      Subsystem . . . . :   Qinter
      Display . . . . . :  V5251JI01

User . . . . . : _____
Password . . . . . :
```

### ► To sign on

---

From the Sign On menu:

1. Key your User ID in the User field
2. Key your Password in the Password field
3. Press Enter










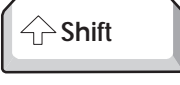





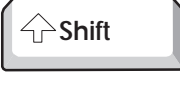


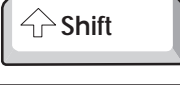


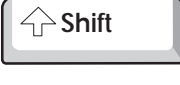

### ► To sign off

---

















On the *Selection* line:

1. Key a double period (. .) or a 90
2. Press Enter

## Standard Menu Function Keys

AS/400 Keyboard	PC Keyboard	Function
		Command Entry Prompt
		Access Menu Word Search
		Retrieve previous command
		Return to previous menu
	 	Fast Path Commands
	 	Menu Selection Detail
	 	Display Menu List window
	 	Access processing options Type desired menu selection and press F18
	 	List available Function Keys

## Standard Screen Function Keys

AS/400 Keyboard	PC Keyboard	Function
		Display J.D. Edwards field level help
		Exit
		Display Fold Area (more detailed information)
		View error message text
		Return to previous form
	 	Clear screen
	 	Display available functions window

## Additional Differences

AS/400	PC Keyboard
Field Exit	Enter
Enter	Ctrl
Reset	Alt
Roll-Up/Down	Page Up/Down
Help	Scroll Lock

## Frequently Used Hidden Selections

To access, key the desired Hidden Selection number on the Selection or Command line and press Enter.

### User Tools

Selection	Description
33	Display Submitted Jobs
34	Display User Messages
42	Display User Job Q
43	Display User Print Q
39	Change User Print Q
82	Hold Submitted Jobs
85	Display User Defaults
90	Sign Off

### Operator Tools

Selection	Description
27	Advanced Operations
29	Technical Operations
97	Install History Display

### Programming Tools

Selection	Description
25	Menu Specifications
40	File Field Description



Type HS on a Selection or Command line to display a list of available Hidden Selections.

# CASE Overview

---

## System Integration

CASE covers the entire spectrum of the application development life cycle, including:

- Design tools
- Code generation
- Automatic documentation generation
- Prototyping
- Repositories
- Other productivity improvement tools

These tools are designed for the development, operation, and maintenance of flexible, business application software.

## Application Development Cycle

There are three levels to the Application Development Cycle (A/D Cycle):

Level 1	The Application Platform, which represents the Technical Foundation course.
Level 2	The Design Platform, which represents the Advanced Programming Concepts and Skills course.
Level 3	The Development Platform, which represents the Program Generator class.

### Specifications

Various Program Generator specifications you use to define a program. You define program purpose and type, specify the files, create help text, define function keys and selection options, and add field-specific logic. You can also create processing options and document Automatic Accounting Instructions (AAIs). After you define the specifications, the Program Generator creates the program for you, adding in the correct validation files and servers to complete the program.

### Fundamentals

There are basic building blocks for a program. Program types are basic definitions of the programs. Using the Question and Answer facility, the system determines, based upon your answers, which program type to select. The program generator builds the program using primary and detail logic modules. Add AAIs to your programs and create CL programs to call completed programs from menu options.

### History of the Program Generator

- Development started in 1984
- First called Clone
- First program generation was in April, 1985
- Rewrite of all systems (World Systems) done through Clone I & II
- Became known as the KBG (Knowledge Based Generator) in 1991
- Became known as the Program Generator in 1992

### Evolution of the Program Generator

#### Clone II Programs

- Dynamic Data Dictionary
- Dynamic totaling and page skipping
  - Created 39 lines of code per field which caused large S002 subroutines

#### Clone II.5 Programs

- Started in 1989
- Dynamic totaling and page skipping
  - Creates 80 lines of code for ALL fields
- No more “?” code generated

- Cursor Sensitive Help (F1)
- F24 Window
- Code for subfile option processing generated

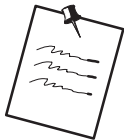
### Current Program Generator Programs

- More utilization of file servers
- Use of the Program Design Language for making user modifications instead of making changes through SEU

## Features

J.D. Edwards provides several tools to help create and customize your programs. Use precompiler commands to specialize your compile environment.

- Program Design Language (PDL) is available to add field-specific logic to you programs.
- Quick Start asks a few basic questions, then creates a basic RPG or CL program with little assistance from you.
- J.D. Edwards also provides many different specialized utilities to assist in the creation and maintenance of your code.
- This language enables you to add calculations or comparisons to specific fields within the program.



You cannot use the Program Generator to modify existing J.D. Edwards programs.

You will become familiar with the following areas:

- ☐ Foundation
- ☐ Program Generator
- ☐ Program Design Language
- ☐ Source Modifications
- ☐ CASE Programs
- ☐ Additional Tools
- ☐ Source Inventory and Database

### What are the Benefits of CASE?

#### J.D. Edwards Functionality

Every program created by the Program Generator automatically includes and uses J.D. Edwards functionality such as:

- Data Dictionary
- User Defined Codes
- Vocabulary Overrides
- Action Code Security
- Standard Function Keys
- Function Key and Selection Exit Security
- Cursor Sensitive Help
- Program Help
- DREAM Writer
- Processing Options

This functionality is consistent across all generated applications because it is built into the Program Generator and Master Source.

#### Quality

The Program Generator is the same tool that has been used to generate the J.D. Edwards application programs for many years. This is high quality code, which has stood the test of time.

#### Increased Productivity

There is a learning curve to become productive with the Program Generator, just as there is for any CASE product. Once the learning curve has been achieved, you will find that it is possible to create simple programs in a short period of time. Due to the standardization of the structure and subroutines of the generated programs, it becomes easier to incorporate complexities, because you know where they belong in either the Program Specifications or the source code.

#### Lasting Value

Since the RPG code is generated from Program Specifications, Program Types, and Master Source Code, it will always be possible to regenerate the source as J.D. Edwards enhances the functionality of its software. Since the enhanced functionality is found in the Master Source Code File, you would only need to regenerate the source code using the original Program Specifications.



## Terms and Concepts

### CASE, as an industry term

As an industry term, CASE is an acronym for computer-aided software engineering. There have been many tools created by different suppliers that implement various aspects of software engineering. These tools can be categorized as upper CASE or lower CASE tools.

Upper CASE tools focus on the business process and data models. Products that provide upper CASE capabilities include tools for organizational charts, decomposition diagrams, entity relationship diagrams, and data flow diagrams.

Lower CASE tools, on the other hand, focus on data models and generating source code. An example of a lower CASE product is J.D. Edward World CASE.

### CASE, as a J.D. Edwards term

As a J.D. Edwards term, CASE refers to a set of tools that are used in the software development process. Listed on the next page are the components of these CASE tools. All of these, except the Computer Assisted Programming tools, were covered in the Advanced Programming Concepts and Skills (APCS) class, which is a prerequisite for the CASE class.

## Detailed Information

### CASE Profiles

#### Computer Assisted Design (CAD)

- Data Dictionary
- User Defined Codes
- File Design Aid (FDA)
- Device Design
  - Screen Design Aid (SDA)
  - Report Design Aid (RDA)

### Computer Assisted Programming (CAP)

- Program Generator
  - Program Purpose and Type
  - File Specifications
  - General Instructions (Help)
  - Option and Function Key Exits
  - Detailed Programming Facility
  - Processing Options
- CL Generator
  - Model CL programs (J98MODEL1, etc.)
  - Quick Start CL Generator
- Quick Start Application Tool

### DREAM Writer

### Menu Design Aid

### J.D. Edwards Source Debugger

### About The Program Generator

The Program Generator is the J.D. Edwards tool that generates source code for both RPG programs and CL programs. In many respects the Program Generator is a very simplistic tool that combines three ingredients and produces the source code as a result of the mixing of the ingredients. The three ingredients are:

- Program Types
- Master Source Code
- Program Specifications

## About Program Types

The Program Generator builds software that can be classified in five categories:

- interactive
- window
- report
- batch
- conversion

These program types contain a list of individual definitions that, when combined, form a functional program. J.D. Edwards calls this a bill of materials. The individually defined parts within the bill of materials are called primary logic modules and are used to build the source code for the program type. Each primary logic module is stored in the Master Source Code File. These logic modules are the components of all J.D. Edwards defined program types.

## About Master Source Code

The Master Source Code File consists of over 11,000 lines of RPG source code. Some lines are pure RPG source code. Others contain some RPG code and some J.D. Edwards directives, which are interpreted by the Program Generator and replaced with pure RPG code. The interpretation of the directives is based upon the Program Specifications that you establish for generating a specific program type.

## About Program Specifications

To generate a program you must first complete the Program Generator Specifications. These specifications are the details of your program that are used to complete the RPG code being built from the master source directives. There are six specifications, two of which are required:

- Selecting a program type
- Identifying the files that are used by the program

After this information is specified, you can generate source code that compiles and executes a simple program.

### Program Types

Program types are defined in five categories:

#### Interactive

- Name is prefixed with an A, B, or D
- Can be either update or inquiry
- May or may not contain Action Codes
- May or may not contain a subfile

#### Window

- Name is prefixed with an E
- Normally used with cursor sensitive helps (F1)
- Sized to fit inside current interactive program

#### Report

- Name is prefixed with a C
- Provides for accumulated values (totals)
- Interfaces with DREAM Writer
- May or may not contain sub-headings

#### Batch

- Name is prefixed with an X
- Used to update master files
- May or may not contain a report

#### Conversion

- Name is prefixed with a Y
- Used to convert data from one file to another
- May or may not contain a report

J.D. Edwards currently provides 25 pre-defined program types with the Program Generator. You will create several of these types. Modification of existing program types and creation of your own program types will be covered in this manual.

## Library Naming Conventions

Your library name depends upon where you are located.

For Example: In the Denver Headquarters Office, we have several classroom numbers and those libraries are structured for that classroom. You will also have your own student library, and that library will take on the naming conventions of your student number. Other libraries that are contained in your library list are libraries which are standard to all J.D. Edwards class environments.

The library list appears as follows:

Name	Contents
Q Libraries	IBM library. Various IBM applications.
COMMON	Common library for training. Used for all J.D. Edwards Training Environments. Contains files that all training classes can share. For example: Help Files, Message Files, Field Reference Files.
STA401OBJ	Student's object library. Student's custom objects are compiled into this library. Will only contain programs that a student may have had to modify in a class exercise.
A4SHARE	Classroom shared library. Is shared for that particular classroom environment. Contains files that the students will all share. For example: Data Dictionary File.
STA401DTA	Student's data library. Used for the student's custom data files. Will only contain files that a student may have had to modify in a class exercise.
TRNSHARE	Shared library for all training. Used for all J.D. Edwards Training Environments. Contains files that all training classes can share. For example: Word Search Files.
JDFOBJ	Common object library for training. Contains all of J.D. Edwards execution programs. All J.D. Edwards training environments use this library.
STA401SRC	Student's Source Library. Used for the student to write custom source programs into. Will only contain programs that a student may have had to modify in a class exercise.
JDFSRC	Common Source Library for Training. Contains all of J.D. Edwards source code programs.

## Menu Overview

J.D. Edwards & Company systems are menu driven. System functions are organized according to their function and frequency of use. The options highlighted on these exhibits illustrate the flow to the functions explained by this guide.

```

G                                     J.D. Edwards & Company          JDED
                                     Master Directory

... GENERAL BUSINESS SYSTEMS          ... INDUSTRY SPECIFIC SYSTEMS
2. Address Book                      14. Distribution / Logistics
3. Electronic Mail                   15. Manufacturing Systems
4. Accounts Receivable               16. Construction Systems
5. Accounts Payable
6. General Accounting
7. Financial Reporting
8. Modeling, Planning,
9. Fixed Assets
10. Equipment/Plant Mana
11. Human Resources Mana

                                     Hidden Selections
                                     Operator Tools
- System Operator Msgs - Sel 41
- Display Active Jobs - Sel 44
- Display Print Writer - Sel 45
- EOJ w/o Sign Off - Sel 30
- Secondary Job - Sel 98
- IBM Q & A Data Base - Sel 84
- Adv/Tech Operations - Sel 27
- Setup Operations - Sel 29
- Install History Dsply - Sel 97

                                     more. . .
                                     Opt: 4=Sel Value F3=Return

Selection or command
=== HS

```

```

G9                                     J.D. Edwards & Company          JDED
                                     Advanced & Technical Operations

... ADVANCED                          ... TECHNICAL
2. Computer Assisted Design          14. Run Time Setup
3. Computer Assisted Programming     15. Documentation Services
                                     16. Computer Operations
                                     17. Project Management
                                     18. Security Officer

Selection or command
===
-
-

```

G93

J.D. Edwards & Company  
Computer Assisted Programming(CAP)

JDED

- |                                 |                           |
|---------------------------------|---------------------------|
| ... DAILY OPERATIONS            | ... ADV/TECH OPERATIONS   |
| 2. Software Versions Repository | 14. Model Program Design  |
| 3. Compile an Object            | 15. Developer's Workbench |
| 4. Quick Start Application Tool | 16. Action Diagramming    |
| 5. Quick Start CL Generator     | 17. Key List Maintenance  |

Selection or command

===

—  
—

G9361

J.D. Edwards & Company  
Model Program Design

JDED

- |                                  |                                  |
|----------------------------------|----------------------------------|
| ... PROGRAM TYPES:               | ... OTHER TOOLS:                 |
| 2. Create/Modify                 | 14. Parameter Copy/Move          |
| 3. Index                         | 15. Print Program Specifications |
| 4. Cross Reference               | 16. Review Source Modifications  |
| 5. Maintain Q/A                  | 17. Generator Updates            |
| 6. Program Search (w/logic type) | 18. CASE Specifications Inquiry  |
| ... LOGIC MODULES:               | ... GENERATION OPTIONS:          |
| 8. Create/Modify                 | 20. Help Instructions Edit/Build |
| 9. Index                         | 21. All Help Instructions        |
| 10. Cross Reference              | 22. Global Program Regeneration  |
| 11. Op Codes                     |                                  |
| 12. Formula Library Entry        |                                  |

Selection or command

===

—  
—

## J.D. Edwards Training Environment

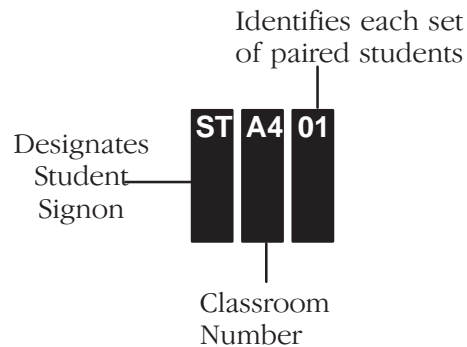
### The Student Library Setup

To help you understand the J.D. Edwards & Company training environment that has been set up for your learning experience, we have provided a list of signon naming conventions, library naming conventions, library content and what files are shared among you and your classmates.

### Signon Naming Conventions

Your signon depends upon where you are located.

For example: In the Denver Headquarters Office, we have several classroom numbers. The structure of signons is shown below.



Your instructor will assign what your User ID will be. Your password will be the same as your User ID.

User ID/Password: \_\_\_\_\_





# Foundation

## Objectives

- To verify the existence of J.D. Edwards provided prerequisites
- To provide user required prerequisites

## About Foundation Information

Before the Program Generator can successfully generate source code, a number of foundation items need to be in place. Some of these are provided by J.D. Edwards, and you must verify their existence. Other prerequisites must be performed by the user.

- ☐ Work with J.D. Edwards Provided Prerequisites
- ☐ Work with User Provided Prerequisites

### What Are the J.D. Edwards Provided Prerequisites?

- Program Generator Files
- Common User Defined Codes
- Source Code for Copy Modules
- Source Code for J.D. Edwards Files

### What Are the User Provided Prerequisites?

- Development Libraries
- Multi-member Source File (JDESRC)
- Job Queues
- Project Management
- CASE Profiles
- Object Authorities





# Work with J.D. Edwards Provided Prerequisites

---

## Working with J.D. Edwards Provided Prerequisites

There are several prerequisites supplied by J.D. Edwards. You must verify their existence.

- Program Generator Files
- Common User Defined Codes
- Source Code for Copy Modules
- Source Code for J.D. Edwards Files

## What Are the J.D. Edwards Provided Prerequisites?

### Program Generator Files

The files used by the Program Generator are categorized below. Each has a specific function when a program is generated. Some of these files are installed with data; others are installed with no data. You need to verify that the files exist in your CASE environment, and that they appropriately contain data or not.

#### Program Generator

These two files are database files and are installed with data.

- Program Types (F93000)
- Master Source (F93001)

#### Source Modifications/Helps

This file is a multi-member source file, and is installed with no members.

- Help/Modification Master (F93002)

#### Program Generator Specifications

These files are database files and are installed with no data.

- Program Purpose and Type (F93101)
- File Specifications (F93102)
- File Formats (F93103)
- Selection/Function Exits (F93104)
- Detail Field Definitions (F93105)
- Automatic Accounting Instructions (F93106)

This file is a database file and is installed with data.

- DREAM Writer Processing Options (F98301)

## Program Design Language (PDL)

These files are database files. F93108 is installed with data; the other two files are installed with no data.

- Generation Operation Codes (F93108)
- Data Item Formulas (F93109)
- Calculation Parameters (F93110)

## Q&A Dialogue

These files are database files and are installed with data.

- Dialogue Master (F00501)
- Dialogue Detail (F00502)
- Dialogue Questions (F00510)
- Dialogue Responses (F00511)
- Dialogue Text (F00512)

## Common User Defined Codes

The Program Generator requires that four User Defined Codes are used. The codes are:

- Logic Modules
  - 93/LM
- Common Subroutine Copy Members
  - 93//C
- Servers
  - 93//X
- Program Types
  - 93/PT

## Logic Modules

93, for Install System Code

LM, for User Defined Codes

- Identifies the pieces of code contained within Master Source Inventory file (F93001) that will be used to make up your RPG program.
- These pieces of code are called logic modules and are discussed in detail.

## Common Subroutine Copy Members

93, for Install System Code

/C, for User Defined Codes

```
00051                      User Defined Code Revisions
                                Install System Code. . . . 93
                                User Defined Codes . . . . LM
Action Code. . . . . I      Skip To Code . . . .
                                Primary Logic Modules
                                _____
10 Character
  Code          Description          Description-2
FILEDEFN01     File Specification
FILEEXTN0      Tables & Arrays - STD Video
FILEEXTN1      Tables & Arrays - SFL Video
FILEEXTN2      Tables & Arrays - STD Rpt
FILEEXTN3      Tables & Arrays - 2F - Conv
FILEEXTN4      Tables & Arrays - Batch
FILEEXTN5      Tables & Arrays - Windows
INPUT1         Data Structures - STD Video
INPUT2         Data Structures - STD Rpt
INPUT3         Data Structures - 2F - Conv
INPUT4         Data Structures - Batch
INPUT5         Data Structures - Windows
INPUT6         Data Structures - Inquiry
MAINLINE       Mainline - Video
```

F5=Code Types F14=Memo F15=Where Used F18=Language F21=Print F24=More

```

00051                                User Defined Code Revisions
                                Install System Code. . . . 93
                                User Defined Codes . . . . /C
Action Code. . . . . I            Skip To Code . . . .
                                Common Subroutine Copy Members

10 Character
   Code      Description          Description-2
C00RSC      Soft Coding Server - Reports C,I
C00SC       Soft Coding Server - Videos C,I
C0000       Business Unit Security Check C
C0001       Edit Action Code         D,E,C
C0001A      Edit Action Code - Req Inquiry D,E,C
C0010       Next Numbering - Automatic E,I,C
C0011       Center Descriptive Titles E,C
C0012       Right Justify Numeric Fields E,C
C0012N      Right Justify Numeric Fields - C
C0015       Currency - Translate Video Fie C
C00151      Currency - Translate Video Fie C
C0016       Format Numeric Fields for Outp E,C
C00161      Format Numeric Fields for Outp C
C00161OLD   Old full RPG version of C00161 E,C

```

F5=Code Types F14=Memo F15=Where Used F18=Language F21=Print F24=More

- Lists all of the copy modules on the system
- Description-2 lists any additional copy modules that are needed to make the common subroutine function properly.
  - For example, C0012 requires copy module E0012

D bring in the file copy modules  
(F specifications)

E bring in the extension copy modules  
(E specifications)

I bring in the input copy modules  
(I specifications)

C bring in the calculation copy modules  
(C specifications)

## File Server Copy Members

```
00051                               User Defined Codes
```

```
Action Code . . . . . I           Install System Code. . . . 93  
See Memo                          User Defined Codes . . . . /X  
10 Character                      Skip To Code . . . _____  
Code                             File Server Copy Members
```

Code	Description	Description-2
X0005	User Defined Code Server	I0005U
X0006	Cost Center Scrub/Validation	I000661
X0901	Account Number Scrub/Edit	I090161
X9203	Data Dictionary Desc Server	I9203A
X9800E	Data Dictionary Server	I9800E

```
F5=Code Types   F14=Memo   F15=Where Used   F18=Language   F21=Print   F24=More
```

## Source Code for Copy Modules

The program generator requires that the source code for the Common Subroutine Copy Members be in the CASE environment.

- Copy modules are stored in file JDECPY in library JDFSRC.

## Source Code for J.D. Edwards Files

It is also required that the source code for J.D. Edwards database files be in the CASE environment.

- File source is stored in file JDESRC in library JDFSRC.



# Work with User Provided Prerequisites

---

## Working with User Provided Prerequisites

There are several prerequisites that must be provided by the user. These prerequisites include:

- Development Libraries
- Multi-member Source File (JDESRC)
- Job Queues
- Project Management
- CASE Profiles
- Object Authorities

Perform the following tasks:

- ☐ Create the multi-member source file (JDRSRC)
- ☐ Access CASE Profiles

## What are the User Provided Prerequisites?

### Development Libraries

There are three types of libraries that are required for CASE generated development.

- Source, which will contain the Development Source File.
- Object, which will contain the CASE generated programs and device files, as well as your non-CASE developed programs and device files.
- Data, which will contain any CASE generated database files.

While it may be customary to create three different libraries for these purposes, it is not required. Either of the following scenarios is acceptable.

#### Unique Libraries

- Source = DEVSRC
- Object = DEVOBJ
- Data = DEVDTA

#### Common Libraries

- Source = DEVLIB
- Object = DEVLIB
- Data = DEVLIB

### Creating the Multi-member Source File (JDESRC)

To use the Program Generator to develop application software, the program source file must:

- Be 142 bytes long (to allow for the Program Generator serial number).
- Contain eight specific fields.



#### **To create the multi-member source file JDESRC**

---

1. Use the CPYF Command

```
CPYF FROMFILE(F93002) TOFILE(LIBRARY/JDESRC) MBROPT(*NONE)
CRTFILE(*YES)
```

```

                                Copy File (CPYF)

Type choices, press Enter.

From file . . . . . > F93002      Name
Library . . . . .      *LIBL      Name, *LIBL, *CURLIB
To file . . . . .      > JDESRC      Name, *PRINT
Library . . . . .      > YOURSRLIB Name, *LIBL, *CURLIB
From member . . . . .      *FIRST   Name, generic*, *FIRST, *ALL
To member or label . . . . .      *FIRST Name, *FIRST, *FROMMBR
Replace or add records . . . . . > *NONE   *NONE, *ADD, *REPLACE
Create file . . . . .      > *YES      *NO, *YES
Print format . . . . .      *CHAR     *CHAR, *HEX

```

```

                                Bottom
F3=Exit   F4=Prompt   F5=Refresh   F10=Additional parameters   F12=Cancel
F13=How to use this display   F24=More keys

```

- The F93002 file is used because it is already in the correct format for program generation.
- The To file may be any name; it is not required to be JDESRC.



CRTSRCPF will not work because it will have only three fields in it, Date, Time, and Data, and the Program Generator requires extra fields.

## 2. Use the RMVM Command

```
RMVM FILE(LIBRARY/JDESRC) MBR(F93002)
```

Remove Member (RMVM)

Type choices, press Enter.

Data base file . . . . .	>	JDESRC	Name	
Library . . . . .	>	YOURSRLIB	Name, *LIBL, *CURLIB	
Member . . . . .	>	F93002	Name, generic*, *ALL	

F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display

F24=More keys

Bottom

After you have created the JDESRC file, you can remove the empty member that was added during the CPYF step.



If you receive the error message CPD3105 for incorrect source file format, your JDESRC file has been created incorrectly, due to either:

- The wrong length
- Improper formatting



### Exercises

See the exercises for this chapter.

## Job Queues

By default, program generation jobs are submitted to the job queue CLONE, and program compile jobs are submitted to the job queue COMPILE. If you want to use these default job queues, then you have to create them and attach them to an existing subsystem.

If you want to use different job queues, or existing job queues, then the defaults have to be overridden, either in the CASE Profile for \*PUBLIC or the CASE Profile for specific users.

## Project Management

### What Are Initial Decisions?

Two decisions need to be made concerning Project Management.

1. The first decision is whether CASE generated programs (or any development work) are going to be managed using the J.D. Edwards SAR System (Software Action Request), which is shipped as part of System 00, General Back Office, under the name of Work Order Processing.
  - Refer to the APCS Manual for more information about the J.D. Edwards SAR System.
  - If you are going to use the J.D. Edwards SAR System for managing software development, you will need to create a SAR before starting the development or have the number of an existing SAR that can be used for development.
  - If you are not going to use the J.D. Edwards SAR system for managing software development, you can disable the SAR number validation by entering \*NONE in the SAR number field of CASE profiles.
2. If you decide to use the J.D. Edwards SAR System to manage software development, then the second decision is whether to use SAR logging.

### SAR Logging

SAR Logging is a process that allows you to associate a SAR number with all of the components of the software development project (e.g., Data Dictionary, UDCs, Files, Programs, Vocabulary Overrides, DREAM Writer, Menus). The purpose of SAR Logging is that it allows you to identify what pieces need to be moved from your development environment to a testing environment and/or a production environment.

If you are going to use SAR Logging, you must decide what method of association will be used to link a SAR number with each piece of the development work. There are two ways of associating a SAR number with development.

- The first method is to use a default SAR number, which is used with all development work until the default number is changed.
- The second method is to be prompted to enter the SAR number as the development work is performed.

### Implementation in CASE Profiles

The results of your decisions are implemented in CASE Profiles.

## What are CASE Profiles?

CASE profiles are user defined values that can pertain to individual users or to one \*PUBLIC user profile.

- Information is stored in the CASE Profiles File (F98009).
- These profiles are used to define the overall CASE operating environment.

Various processing control parameters are defined by the user including:

- Default development libraries
- Compile job queue
- Program Generator source generation job queue
- Compile print options
- SAR logging options
- Immediately update the record for User ID \*PUBLIC.
- When entering information for \*PUBLIC, all fields are required.



- Default CASE Profile values are maintained in a record with the User ID \*PUBLIC. CASE Profile values for individual users should be entered only if overrides to the \*PUBLIC values are needed.
- When entering values for individual users, all fields may be left blank except for the specific values being overridden.

## Accessing CASE Profiles

There are two ways to access CASE Profiles.

### ► To access CASE profiles

Select one of the following methods:

- Select Case Profiles from the Computer Assisted Design menu

```
G92                      J.D. Edwards & Company          JDED
Programmers              Computer Assisted Design (CAD)

... SYSTEM DESIGN AIDS    ... PROGRAM DESIGN AIDS
 2. Software Versions Repository 14. Processing Options
 3. Menus                  15. Help Instructions
 4. Data Dictionary        16. Universal File Converter
 5. Model Relations
 6. CASE Profiles
 7. Function Key Definitions
 8. Vocabulary Overrides

Selection or command
====> _____
_____
```

- From the Repository Services menu in the Software Versions Repository

```

9801                               Software Versions Repository

Action Code. . . . . -
Member ID. . . . . _____
Description. . . . . _____
Function Code. . . . . _____
Function Use . . . . . _____
System Code. . . . . _____
Reporting System _____
Base Member Name _____
Maint/RSTDSP . . . . . - Omit Opt
Copy Data (Y/N). . . . . - Optional

O Source Object Source
P Library Library File
_____
_____
_____
_____
_____
_____
_____
_____

98500-----Repository Services-----
"1" Available Services
- Data Dictionary
- Menus
- Vocabulary Overrides
- Function Key Definitions
- Dream Writer Versions
- Processing Options
- User Defined Codes
- Edit System Helps
1 CASE Profiles
- SAR Log Inquiry
- Copy DD,VO,DW,UDC,SVR,Menus
-Sel:--"1"=Select-----F12=Previous-----

Opt: 1=Browse 2=Edit 3=Copy 5=SAR 8=Print 9=Dlt 10=Design 14=Crt F24=More

```

Select the CASE Profiles option

The new CASE Profiles screen appears. The program will attempt to automatically inquire on your User ID. If your ID is not set up, an error will occur and you can then inquire on \*PUBLIC.

```

98009                               CASE Profiles

Action Code. . . . . I
User ID. . . . . *PUBLIC
Default Development Environment
Source File . . . . . JDESRC
Source Library. . . . . PGFSRC71
Object Library. . . . . PGFOBJ71
CL Source File. . . . . JDECLSRC
Data File Library . . . . PGFDTA71
SAR Number. . . . .
Version ID. . . . . A71
Status Code . . . . . 2
SAR Options
SAR File Library. . . . DDPDATA
SAR Delivery Type . . . *LOG
Miscellaneous
Source Gen Opt (Future) - SEU
Helps Maint Opt(Future) -
F24=More Keys

Program Creation Options
Compile Job Queue . . . COMPILE
Prog Gen Job Queue. . . CLONE
Compile Target Release. *CURRENT
Print Option . . . . . 1
Cross-Reference Listing N
A7.1 Base
Custom

```



## Default Development Environment

Field	Explanation
Source File	The default file name where source is to be stored within the source library. Must reside within the specified library.
Source Library	The default library where source will be stored. The source file specified above must reside within this library.
Object Library	The default library where compiled objects will be stored.
CL Source File	The default file where source for CL programs will be stored. This file must reside within the specified source library.
Data File Library	Used to specify the test (or development) library for physical and logical files. Used as the default object library for the Software Versions Repository when copying source code for physical or logical files.
SAR Number	<p>An abbreviation for software action request (SAR).</p> <ul style="list-style-type: none"> <li>• *NONE = the SAR number will not be validated in any of the CAD/CAP programs and can be left blank.</li> <li>• If a SAR number is entered, it is used in conjunction with the SAR Delivery Type of *DFT (default).</li> </ul>
Version ID	The software release number to be defaulted in the Software Versions Repository file.
Status Code	Determines the status of the software as well as where it resides in production. It indicates that the software is in production, development, or a release.



If the SAR Number is left blank, you are required to enter a valid SAR number when using the CAD/CAP tools.

## Program Creation Options

Field	Explanation
Compile Job Queue	Specifies which job queue will be used when submitting programs to compile. Used for programs with a function code of RPG, CBL, PLI, C, and SYSC
Prog. Gen Job Queue	Specifies which job queue will be used when submitting jobs for the Program Generator. These jobs include the source code generation and the source code monitor from SEU.
Compile Target Release	<p>Used by various AS/400 compilers (RPG, CLP, COBOL, C) to compile an object compatible with a specified target release.</p> <p>A value of *CURRENT compiles an object to the release of the machine at compile time.</p> <p>A value of *PRV compiles an object compatible with one release back.</p>
Print Option	<p>Used to designate whether or not a report will be generated when an object is compiled.</p> <p>0 = no print. 1 = print 2 = print and hold spool file. 3 = print only — does not generate an execution object (applies to COBOL and RPG only). 4 = print when creation or compile fails.</p>
Cross-Reference Listing	<p>Used to designate whether or not a report will be generated when an object is compiled.</p> <p>Specifies whether a cross-reference listing will be generated for variables and fields in a program's compile listing.</p>

## SAR Options

Field	Explanation
SAR File Library	<p>Specifies the library where the Software Action Request (SAR) file being used for software development exists.</p> <ul style="list-style-type: none"> <li>• If left blank, the user's library list will be used.</li> <li>• Specify *NONE in the SAR number field if you do not want any SAR number editing SAR</li> </ul>
SAR Delivery Type	<p>Associated with SAR logging, which tracks all modifications to J.D. Edwards software. For example, it will track when User Defined Codes are modified.</p> <p>*NONE = no logging.</p> <p>*LOG = log to SAR #00000000 (no SAR number is used for logging).</p> <p>*DFT = log to a default SAR number (specified in the SAR Number field).</p> <p>*PROMPT = log and prompt the user for the SAR number to be used and allow the user to enter the revision notes.</p>

## Miscellaneous Options

Field	Explanation
Source Gen Opt	For future use.
Helps Maint Opt	For future use.

## What Are the Function Key Exits?



### F9 – Previous Profile

- Allows the user to inquire again on the last record updated.

### Summary of CASE Profiles

- Update the \*PUBLIC record as well as add any additional individual records.
- You cannot delete the \*PUBLIC record.
- When you enter information for the \*PUBLIC record, all fields are required.
- The record for User ID \*PUBLIC contains the values that will be used as the defaults for all users unless individual user profiles have been set up.

- When you enter values for individual profiles, all fields are left blank except for the specific values being overridden on the \*PUBLIC profile.
- SAR Number and SAR Delivery type work together to determine what type of SAR logging should occur.

*NONE	no SAR logging at all.
*LOG	no SAR number will be included as part of the SAR logging.
*DFT	the SAR number specified will be used for the SAR logging.
*PROMPT	the user will be prompted for a SAR number and revision notes when an entry is about to be made to the SAR log.



### Exercises

See the exercises for this chapter.

## Object Authorities

The user's authorities to some objects are checked at different steps in the generation of programs using CASE. Therefore, it is necessary that these authorities be reviewed initially.

### Job Control Authority

On the user's IBM User Profile, it is necessary that the Special Authority parameter be set to \*JOBCTL. This authority is necessary when entering the CASE Specifications.

### Source Library

It is necessary that the user have Object Management authority to the Source Library that is used for software development.

### Source File

It is necessary that the user have Object Management authority to the Source File that is used for software development.

### Job Queues

It is necessary that the user be authorized to use the job queues for generating source code and compiling programs.





# Program Generator

## Objectives

- To Define the Program Generator Specifications
- To Define the Program Purpose and Type
- To Define the File Specifications
- To Define General Instructions
- To Define Option and Function Keys
- To Work with the Detailed Programming Facility
- To Define Processing Options

## About Program Generator Steps

The Program Generator uses a series of steps to create a program. Perform the following tasks:

- ☐ Define Program Generator Specifications
- ☐ Define Program Purpose and Type
- ☐ Work with File Specifications
- ☐ Define General Instructions
- ☐ Define Option and Function Key Exits
- ☐ Work with the Detailed Programming Facility
- ☐ Define Processing Options







# Define Program Generator Specifications

---

## Defining Program Generator Specifications

The Program Generator uses specification forms to create a program.

- Only two specifications are required to be entered by the user:
  - Program Purpose and Type
  - File Specifications
- A third specification is required but is automatically created after you enter the File Specifications
  - Detailed Programming Facility
- Optional specifications include:
  - General Instructions
  - Option and Function Keys
  - Processing Options
  - Automatic Accounting Instructions

Only source for files and common copy modules are required during the specifications and generation steps. Objects will not be required until you compile the program.

Perform the following tasks:

- ☐ Access the Program Generator
- ☐ Define Program Generator specifications

## Accessing the Program Generator

### ► To access the Program Generator

---

From the Computer Assisted Programming (CAP) menu

```
G93                      J.D. Edwards & Company          JDED
Programmers              Computer Assisted Programming(CAP)

... DAILY OPERATIONS          ... ADV/TECH OPERATIONS
2.  Software Versions Repository 14.  Model Program Design
3.  Compile an Object           15.  Developer's Workbench
4.  Quick Start Application Tool 16.  Action Diagramming
5.  Quick Start CL Generator     17.  Key List Maintenance
```

```
Selection or command
====> _____
_____
```

1. Select Software Versions Repository.
  - The form that appears serves as the front end to all J.D. Edwards Design tools including the Program Generator.
2. Inquire on the 'P' member from the Software Versions Repository.  
(Class example will be P92801)
3. Enter option 10, Design, next to the selected environment to enter Program Design Aid.
  - The Program Generator definition screen appears.

## Defining Program Generator Specifications

### ► To define Program Generator specifications

Choose the appropriate option from the Define Generator Specifications form.

```

93100M                                Define Generator Specifications

Member ID. . . . . P92801             File ID. . . . . JDESRC
SAR Number . . . . . 834451           Src Library. . . . . JDFSRC71

Type 1 next to desired option(s) and press ENTER.
Press F21 to select all.
">" identifies functions already defined.

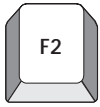
Opt   Program Generator Definition Option
-     > Program Purpose and Type
-     > File Specifications
-     > Define General Instructions
-     > Define Option and Function Key Exits
-     > Detailed Programming Facility
-     > Define Processing Options

Opt: 1=Define  F2=Monitor  F6=Repository  F9=Search  F21=Sel All  F24=More

```

Field	Explanation
Program Purpose and Type	Defines what kind of program you are designing and the status of the program generation (CAP status).
File Specifications	Allows the user to enter the data base files to be used by the program you are designing.
Define General Instructions	Allows the user to enter program-specific help instructions.
Define Option and Function Key Exits	Allows the user to define special program exits.
Detailed Programming Facility	Allows the user to specify data field definition parameters for fields included in the screen, the report, and the master file(s).
Define Processing Options	Allows the user to define processing options the program can use.

## What are The Function Key Exits?



### F2 – Monitor

After the user enters all of the Program Generator specifications, they can press F2 to see if the monitor program can detect any pre-defined errors.

- This program checks for important features that are pertinent to the generation of source code by the Program Generator
- This program will not check for things such as forgetting to regenerate the file specifications after you have changed your video file

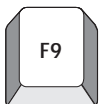
The following is a list of items checked by the monitor program. This list is subject to change as the monitor program is enhanced. It checks:

- For \$\$ fields specified in the TOTAL formats of the report file for the Program Generator totaling feature
- That the field SH#RRN is defined for programs processing by relative record number
- For a file information data structure being defined for programs processing by relative record number
- For a keyed master file for programs processing by relative record number
- For a field being defined as mandatory entry *N* for transaction processor programs (subfiles)
- For a hidden field being defined for transaction processor programs
- To see if the master file key fields are defined as output
- Fields that are set up to use next numbering have a validation file attached



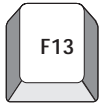
### F6 – Repository Services

- Displays a window of J.D. Edwards repositories



### F9 – Search

- Exits to the Software Search facility
- The user can enter a program name to show all programs that equal or are greater than the search criteria



## F13 – Automatic Accounting Instructions

- For documentation only. No code is generated.

```

93106                      Automatic Accounting Instr          Action Code : C
Program Name: P92801      Item Maintenance

-----
Key . . . . . GLG11
Purpose . . . . . Item revenue
-----
Use: Cost Center. . . N          (R = Required O = Optional N = Not Used)
    Object . . . . . R
    Subsidiary . . . N
Cost Cntr Dflt Fld. . .
-----
Key . . . . . GLG13
Purpose . . . . . Item Expense
-----
Use: Cost Center. . . N          (R = Required O = Optional N = Not Used)
    Object . . . . . R
    Subsidiary . . . N
Cost Cntr Dflt Fld. . .

```

Field	Explanation
Key	Organizes AAIs into groupings. <ul style="list-style-type: none"> <li>• Numbers, that are set up, group the accounts according to the systems that use the AAIs</li> </ul>
Purpose	Explains purpose of the AAI used in this program.
Cost Center	Specifies whether the cost center in the AAI parameter as used by this program is required (R), optional (O) or not applicable (N).
Object	Specifies whether the object in the AAI parameter as used by the program is required (R), optional (O), or not applicable (N).
Subsidiary	Specifies whether the subsidiary in the AAI parameter as used by this program is required (R), optional (O), or not applicable (N).
Cost Cntr Dflt Fld	Specifies the alternate assignment of the cost center when using AAI accounts.



### F21 – Select All

- Prompts the user through all of the Program Generator definition screens.



### F23 – Delete All Specifications

- Deletes all of the Program Generator specifications for the program displayed.
- Removes the Pxxxxx and Hxxxxx members from the F93002 file.



If definitions have already been entered for a program, a > symbol will show up next to the specification that has been defined. The field will also be highlighted.

# Define Program Purpose and Type

---

## Defining Program Purpose and Type

Program purpose and type is a required specification. Defining the program purpose and the program type is the first step in the creation of an RPG program through the program generator.

Program types specify the basic function of the program. There are program types for:

- Interactive maintenance programs
- Programs with subfiles
- Report programs
- Conversion programs

The Program Purpose and Type screen also includes information about regenerating the program, the SAR associated with the program, and the install system.

To define the program purpose and type complete the following tasks:

- ☐ Define the program purpose and type
- ☐ Identify the program type

► **To define program purpose and type**

---

From the Define Generator Specification form

```
93100M                      Define Generator Specification

Member ID. . . . . P92801      File ID. . . . . JDESRC
SAR Number . . . . . 834451    Src Library. . . . . JDFSRC71

Type 1 next to desired option(s) and press ENTER.
Press F21 to select all.
">" identifies functions already defined.

Opt   Program Generator Definition Option
_     > Program Purpose and Type
_     > File Specifications
_     > Define General Instructions
_     > Define Option and Function Key Exits
_     > Detailed Programming Facility
_     > Define Processing Options
```

Field	Explanation
Program Purpose and Type	Defines what kind of program you are designing and the status of the program generation (CAP status). <ul style="list-style-type: none"><li>• This is a required definition</li><li>• Information is stored in F93101</li><li>• Creates the Pxxxxx member in F93002</li><li>• Creates a data item in Data Dictionary (F9200)</li></ul>

1. Select the Program Purpose and Type Option



93100 Program Purpose and Type

Action Code. . C

Program ID . . P92801

Title. . . . . Item Maintenance

Purpose

To allow for the addition, revision, deletion, and inquiry of items based upon their own business unit.

Install System 92

CAP Status . . Y

SAR Number . . 834451

Program Type . D0040

SFL/T/F - w/Act - w/Sel - Keys

Lockout Act. . \_ \_ \_ \_ \_

F11=Pgm Type Selection F2=Program Type X-Ref

## 2. Complete the Program Purpose and Type Form

- Allows user to define the purpose and type of program being created
- Additional information defaults from Software Version Repository

Field	Explanation
Program ID	RPG program name specified in the Software Versions Repository. System adds a data item by this name, with a glossary group of <i>P</i> to the data dictionary as part of this program definition.
Title	The title defaults to the description in the Software Versions Repository and should not be changed.  When help instructions are generated, this title appears as the Help program title.  Serves as the alpha description for the data item previously mentioned.
Purpose	Full glossary definition that is stored in the data dictionary.  This becomes the summary statement in the help instructions.
Install System	Defaults to the system specified in the Software Versions Repository.
SAR Number	Defaults to the SAR entered in the Software Versions Repository.

Field	Explanation
CAP Status	<p>Indicates whether the source code can be generated or not.</p> <p>Default is Y for Yes.</p> <p>Should only be set to N if the Program Generator is not used to generate the program source or if the source generation process is complete and the program has moved into production and your production source library is <i>not</i> 142 bytes long.</p>
Program Type	<p>Identifies the basic function or type of program that is being created.</p> <p>When creating a program, the <i>Program Type</i> field is determined from the Q&amp;A Dialogue.</p> <p>To change a program type, press F11 which will display the "Dialogue text".</p> <p>If you change program types, the program generator will prompt for the deletion of all prior source modifications.</p>
Lockout Act (Action)	<p>Allows the user to specify which action codes they do not want included in the program.</p> <p>Any codes listed will not be allowed. That is, the program will not allow the indicator associated with the action code being locked out to ever be turned on. The source to process the <i>Action Code</i> will still be included but the associated indicator will never be allowed to be turned on.</p> <p>Utilizes array @NAC in the programs.</p>

## ► To identify program type

Complete the Program Type Dialogue Selection form

- Series of questions to determine Program Type

98533
Dialogue Selection
\*DEFAULT  
LC

The following tutorial is designed to help you choose a program type for KBG generation processing.

Question:
Opt

Of what general type is the program?  
OR  
If you know the correct logic type enter the desired value where indicated.

Responses:

An interactive program.....	—
An interactive window program.....	—
Print a report.....	—
Conversion program.....	—
Batch update program.....	—

Opt: X=Select
F5=Review Selections
More...



If the user knows the program type, they can page down to an entry field to enter the program type name.

The following pages illustrate the flow used in selecting the proper program type.

Of what general type  
is the program?

*Interactive*

A

*Interactive window*

E0010

*Print a report*

B

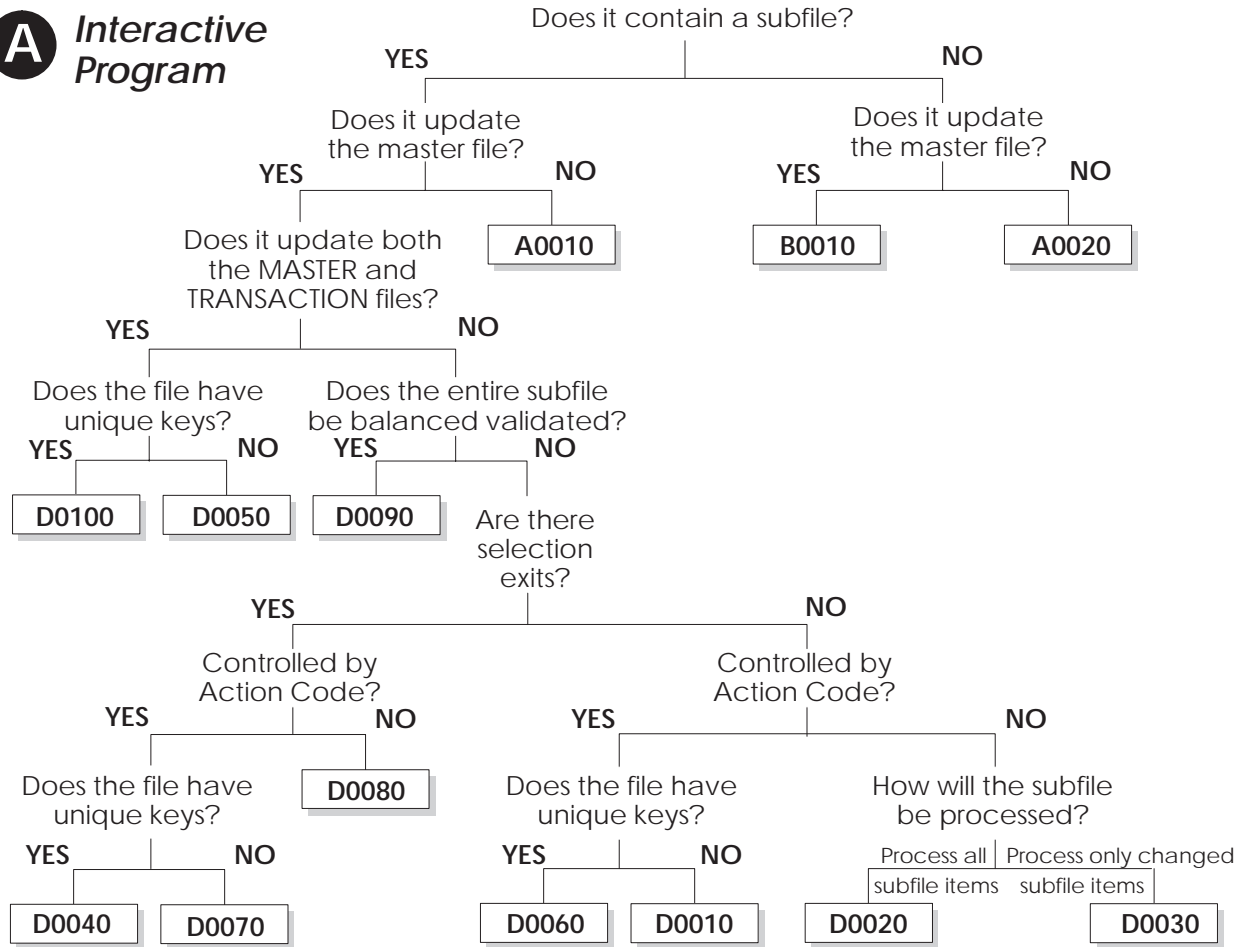
*Conversion program*

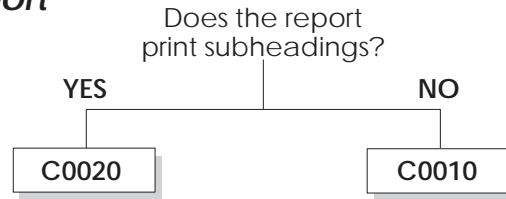
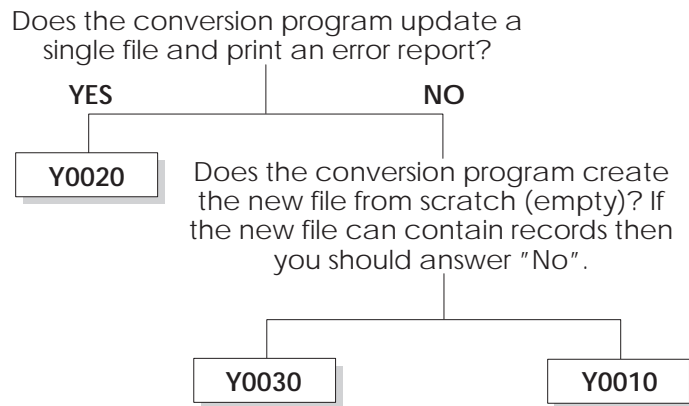
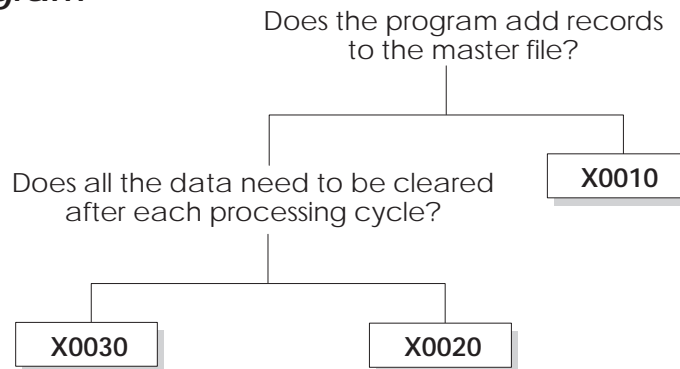
C

*Batch update program*

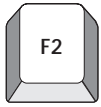
D

**A** *Interactive Program*



**B** *Print a Report***C** *Conversion Program***D** *Batch Update Program*

## What Are the Function Key Exits?



### **F2 – Program Type Cross-Reference**

Allows the user to view all the programs that are defined the same as the selected program type.



### **F11 – Program Type Selection**

When you copy a program that already has its specifications determined, you will not need to go through the question and answer process, which is used to determine the class of logic or program type.

If there is no program type specified or the user presses F11 for Program Type Selection, the first dialogue screen will appear.





# Work with File Specifications

---

## Working with File Specifications

The program generator requires that you specify the files for your program. The program generator adds any necessary validation files and servers.

File specifications allow you to enter the database files your program uses. After you select your specific program type, continue by completing the File Specifications screen.

Perform the following tasks:

- ☐ Access file specifications
- ☐ Understand file specifications
- ☐ Process file specifications
- ☐ Generate source from file specifications

## Accessing File Specifications

### ► To access file specifications

---

From Define Generator Specification select File Specifications

```
93100M                      Define Generator Specification

Member ID. . . . . P92801      File ID. . . . . JDESRC
SAR Number . . . . . 834451    Src Library. . . . . JDFSRC71

Type 1 next to desired option(s) and press ENTER.
Press F21 to select all.
">" identifies functions already defined.

Opt  Program Generator Definition Option
-    > Program Purpose and Type
-    > File Specifications
-    > Define General Instructions
-    > Define Option and Function Key Exits
-    > Detailed Programming Facility
-    > Define Processing Options
```

---

Field	Explanation
File Specifications	Allows the user to enter the data base files to be used by the program you are designing. <ul style="list-style-type: none"><li>• This is a required definition</li><li>• Information is stored in F93102 and F93103</li><li>• Creates F93105 records</li></ul>

---

## Understanding the File Specifications Form

The File Specifications form

- Allows the user to list the files necessary for the program
- Defines the file usage for each file, such as input, output, or update

93102
File Specifications
Action Code. C

Name: P92801
Item Maintenance

File	Input	Output	Update	Add	CC	Sec	
F0001	X	-	-	-	-	-	Business Unit Security
F92801	X	-	-	-	-	-	SDM Item Master File
F92801LA	-	-	M	X	-	-	LF - Cost Center, Item ID
V92801	-	-	-	-	-	-	Item Maintenance
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	
	-	-	-	-	-	-	

F3=Exit w/o Field Generation
F4=Xtnd Parms
F5=Data Model
F9=Search

Field	Explanation
File	The member ID for the file.
Input	Identifies a file as an input-only file. <ul style="list-style-type: none"> <li>• M or 1 through 9 = Master input file</li> <li>• X = General input file</li> </ul>
Output	Identifies a file as an output-only file. <ul style="list-style-type: none"> <li>• X = General output file</li> </ul>

Field	Explanation
Update	Identifies the file as an update-capable file. <ul style="list-style-type: none"><li>• M or 1 through 9 = Update master file</li><li>• T = Transaction (detail) update file</li></ul>
Add	Identifies the file as an add-capable file. <ul style="list-style-type: none"><li>• X = Add-capable file</li><li>• Can be left blank for an update file, and the X will automatically be added</li></ul>
CC Sec	Obsolete field. Was used in previous releases to invoke Business Unit security.



- A significant feature of the Program Generator is its interpretive ability to include secondary editing and referencing files.
- If update is specified for a file, the Program Generator examines all fields in that file and includes any other files required to edit those fields during an update.

## What Are File Specifications?

A key step to the successful generation of source code is the correct specification of the master file(s) for a program. All of the database I/O operations for the master file(s) are based on being specified with the correct value in the correct column.

PROGRAM TYPE	DESCRIPTION	SPECIFICATION
A0010	SFL Inquiry	The master file is specified with an M or 1 in the Input column.
A0020	Single Record Inquiry	
C0010	Standard Report	
C0020	Standard Report — Subheading	
C0025	Standard Report — Subheading above Columns	
E0010	Window	
B0010	Single Record Maintenance	The master file is specified with an M or 1 in the Update column
D0040	SFL Maintenance — KEY	
D0045	SFL Maintenance — KEY, No Action Code	
D0060	SFL Maintenance — KEY	
X0010	Batch Update — 1 File	
Y0020	File Conversion — 1 File	
Y0030	File Conversion — 1 File	
D0100	SFL Maintenance — KEY, 2 Update Files	The master file maintained in the SFL Control format is specified with a 1 in the Update column, and the transaction file maintained in the SFL format is specified with a 2 in the Update column.
D0010	SFL Maintenance — RRN	The logical file that is used to fill the subfile is specified with a 1 in the Input column. Also, a File Information Data Structure name is entered for the logical file in the fold area.  The physical file that is updated is specified with a 2 in the Update column. Also, the physical file is identified as a non-keyed file in the fold area.
D0020	SFL Maintenance — RRN No Act	
D0030	SFL Maintenance — RRN No Act	
D0070	SFL Maintenance — RRN	
D0070	SFL Maintenance — RRN	
D0080	SFL Maintenance — RRN No Act	
D0090	SFL Maintenance — RRN	

PROGRAM TYPE	DESCRIPTION	SPECIFICATION
D0050	SFL Maintenance — RRN, 2 Update Files	<p>The master file maintained in the SFL Control format is specified with a 1 in the Update column.</p> <p>The logical file that is used to fill the subfile is specified with a 3 in the Input column. Also, a File Information Data Structure name is entered for the logical file in the fold area.</p> <p>The physical file that is updated is specified with a 2 in the Update column. Also, the physical file is identified as a non-keyed file in the fold area.</p>
X0020	Batch Update, 2 Files	The input file is specified with a 1 in the Input column. The output file is specified with a 2 in the Update column.
X0030	Batch Update, 2 Files	
Y0010	File Conversion, 2 Files	



### Notes about File Specifications

- You must have one master file specified (M or 1).
- Do not specify one file with an M and another file with a 1.
- The function code must be correct on the Software Versions Repository for the video/report specified or no moves to the video/report will be generated by the Program Generator.
- The user does not have to enter anything in the columns for a video or report. The Program Generator knows what to do with them automatically.
- Users may use non-J.D. Edwards files with the Program Generator, but they must enter the file in the Software Versions Repository.

## Processing File Specifications

### ► To process file specifications

---

Complete the File Specifications form and press enter.

- A job runs interactively that analyzes the File Specifications and creates records for three Program Generator files. As this work is being performed, you will see messages flashing at the bottom of their screen as the job progresses.

#### **F93102**

Updated with one record for each file named in the File Specification.

- If the master file includes a Business Unit field, then the Business Unit security file F0001 is added to the File Specifications.
- If the master file is for an interactive program and contains a field that uses a validation file, then that file is added to the File Specifications. If this validation file is not needed, it can be deleted by going back into the File Specifications screen and pressing Field Exit through the file name, and then pressing Enter.

#### **F93103**

Updated with one record for each format in each file.

- If the file is a database file, then the F93103 record contains the name of the Key List that the Program Generator will use, and the name(s) of the Key Field(s).

#### **F93105**

Updated with one record for each field in each file. These records are used in the Detailed Programming Facility.

- If the file is a master file or device file, then all fields are included.
- If the file is a database file used only for input purposes, then only the key fields are included.



## Notes about Processing File Specifications

If changes are made to the fields in any of the files used by the program, you must rerun the File Specifications step. If a field has been deleted from a file, you will need to delete that field from the Detailed Programming Facility manually. Rerunning the File Specifications step will not remove records from the F93105 file.

## What Are the Function Key Exits?



### F4 – Extended Parameters

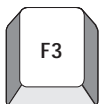
- A fold area is displayed which contains the library names where the source for this file is located. Default library names come from the Software Versions Repository and your library list.

```

93102                               File Specifications                               Action Code. C
Name: P92801      Item Maintenance
   File   Input Output Update Add CC Sec
F0001      X      -      -      -      -      -      Business Unit Security
Src Lib/File . . JDFSRC      / JDESRC      Keyed(Y/N) . Y  File Info DS . _____
PF Src Lib/File. _____ / _____      External(Y/N). Y
F92801      X      -      -      -      -      -      SDM Item Master File
Src Lib/File . . PGFSRC      / JDESRC      Keyed(Y/N) . Y  File Info DS . _____
PF Src Lib/File. _____ / _____      External(Y/N). Y
F92801LA     -      -      M      X      -      -      LF - Business Unit, Item ID
Src Lib/File . . PGFSRC      / JDESRC      Keyed(Y/N) . Y  File Info DS . _____
PF Src Lib/File. JDFSRC      / JDESRC      External(Y/N). Y
V92801      -      -      -      -      -      -      Item Maintenance
Src Lib/File . . PGFSRC61    / JDESRC      Keyed(Y/N) . Y  File Info DS . _____
PF Src Lib/File. _____ / _____      External(Y/N). Y
Src Lib/File . . _____ / _____      Keyed(Y/N) . _  File Info DS . _____
PF Src Lib/File. _____ / _____      External(Y/N). _
F3=Exit w/o Field Generation  F4=Xtnd Parms  F5=Data Model  F9=Search
  
```

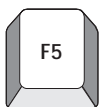


Field	Explanation
Src Lib/File	File and library that contains the file source.
Keyed (Y/N)	Y indicates the data file being accessed is keyed or accessed by a relative record number if N. <ul style="list-style-type: none"> <li>If processing by RRN, the physical file that is being updated must be specified as keyed = N.</li> </ul>
File Info DS	Name assigned to an RPG III file information data structure if needed for an associated data file. <ul style="list-style-type: none"> <li>If processing by RRN, the logical file that is used to retrieve database records must have a file information data structure name. In addition, you must define SH#RRN as a hidden field on the SFL format of the video, since the Program Generator uses this field to store the relative record number.</li> <li>Suggested naming conventions are INFDS1, INFDS2, etc.</li> <li>Used with a keyed data file that does not have UNIQUE keys.</li> <li>If you use the POST operation code from IBM for a file information data structure, you <i>must</i> do it for every file information data structure in the program.</li> </ul>
PF Src Lib/File	File and library that contains the physical file linked to the logical file.  Used when a logical file is in a development library but the physical is in a production library.
External (Y/N)	Strictly reserved for files designated in multi-file access utility functions that deal with source files or with other types of system software. Otherwise, it is always set to Y.



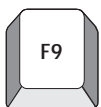
### F3 – Exit

- Allows the user to exit the program without actually updating the file specifications



### F5 – Data Model

- Exits to the Data Modeling facility
- Must rebuild a data model before you can view it



### F9 – Search

- Exits to the Software Search facility
- The user can enter a program name to show all programs that meet or are greater than the search criteria

## Generating Source from Specifications

After you define the Program Purpose and Type and enter all File Specifications, you may generate the source for your program.

Notice that the Detailed Programming Facility has been updated from your File Specifications.



### To generate source from specifications

From the Software Versions Repository

```

9801                               Software Versions Repository

Action Code. . . I
Member ID. . . . P92801
Description. . . Item Maintenance
Function Code. . RPG   RPG Programs
Function Use . . 198   Model Source Member
Install System . 92    Computer Assisted Design
Reporting System 92    Computer Assisted Design
Base Member Name P92801
Maint/RSTDSP . . 1    Omit Option. . . _
Copy Data (Y/N). N    Optional File. . N Common File. . . N

                                DREAM Writer Form Exists
O Source   Object   Source   SAR   Version   S D   User   Date
P Library  Library  File     Number  ID       C P   ID    Modified
-- JDFSRC71 JDFOBJ71 JDESRC   8344551 A71      1 _   QUARLES 07/01/94
--
--
--
--
--
--
--
--
                                KBG Generation Submitted
Opt: 1=Browse 2=Edit 3=Copy 5=SAR 8=Print 9=Dlt 10=Design 14=Crt

```

1. Inquire on P member.
2. Enter option 15 to generate your source.
  - A batch job will be submitted to process your program specifications.
  - This job has a naming convention which is your member ID prefixed with a G.
  - This job is submitted to the generation job queue defined in your CASE Profile.
3. After completion of your generation, select option 14 to compile your program.
4. Review your compile for errors and correct any errors.
  - Repeat steps two and three if necessary.



## Exercises

See the exercises for this chapter.



# Define General Instructions

---

## Defining General Instructions

General Instructions lets you develop program-specific help text for programs you create. To work with define general instructions you should:

- Understand the edit screen
- Understand the use of special characters
- Know how to update the help file

Perform the following tasks:

- ☐ Access define general instructions
- ☐ Update the help instructions

## Accessing Define General Instructions

### ► To access Define General Instructions

---

From Define Generator Specification select Define General Instructions

```
93100M                      Define Generator Specification

Member ID. . . . . P92801      File ID. . . . . JDESRC
SAR Number . . . . . 834451    Src Library. . . . . JDFSRC71
```

```

Type 1 next to desired option(s) and press ENTER.
Press F21 to select all.
">" identifies functions already defined.
```

```

Opt      Program Generator Definition Option
_          > Program Purpose and Type
_          > File Specifications
_          > Define General Instructions
_          > Define Option and Function Key Exits
_          > Detailed Programming Facility
_          > Define Processing Options
```

---

Field	Explanation
Define General Instructions	Allows the user to enter program-specific help instructions. <ul style="list-style-type: none"><li>• Information is stored in the <i>Hxxxxx</i> member in F93002</li></ul>

---

## About the Edit Screen

- Allows user to change or enter general instructions for a program
- User should stay between columns 5 and 70 or text will be truncated

```

Columns . . . :    1  71                Edit                JDFCLONE6/F93002
SEU==>                                     H92801
FMT **  ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
          ***** Beginning of data *****
0001.00  ~Item Master Maintenance (P92801)~
0002.00
0003.00
0004.00  ¢OVERVIEW¢
0005.00
0006.00  The user inquires on a cost center. All item master records
0007.00  with that cost center are displayed in a subfile. The user can
0008.00  then add, update, and delete the item master records that are
0009.00  displayed
0010.00
0011.00  This screen controls the following:
0012.00
0013.00      o The action performed. You can add, change, delete or
0014.00      inquire.
0015.00
0016.00      o The files updated. The system will update the Item Master
F3=Exit  F4=Prompt  F5=Refresh  F9=Retrieve  F10=Cursor
F16=Repeat find  F17=Repeat change  F24=More keys
(C) COPYRIGHT IBM CORP. 1981, 1992.

```

## About Special Characters

### Special Characters for General Instructions

Character	Explanation
**	Must be in positions 1 & 2. Causes a page skip when printed.
++	Must be in positions 1 & 2 followed by a data item. Causes the most current data dictionary information to be pulled in.
>>	Copy in all help instruction records for the program listed after the >>. Only shows when printed.
//BYPASS	Marks the beginning of help information that should be ignored. Beginning of comment lines.
//END	Marks the end of help information that should be ignored. End of comment lines.
	Underline.
¢	Underline and high intensity.
~	High intensity (press Upper Shift + Tilde, then press the Space Bar). (ALT + HEX + A1 if you do not have a ~ on your keyboard)



## Special Characters within Help Instructions

```

Columns . . . :   1  71           Edit           JDFCLONE/F93002
SEU==>                H55100
***** Beginning of data *****
0001.00      This help instruction illustrates the use of the special
0002.00      characters available for on-line helps.
0003.00
0004.00      |General Use|
0005.00      By using the double bar, the field(s) that are enclosed will
0006.00      have an underline.
0007.00
0008.00      ~Additional features~
0009.00      By using the tilde, the field(s) that are enclosed will
0010.00      will have highlighting.
0011.00
0012.00      ¢Special Considerations¢
0013.00      By using the cent sign, the field(s) that are enclosed will
0014.00      have an underline and will be highlighted.
0015.00
***** End of data *****

```

(C) COPYRIGHT IBM CORP. 1981, 1992.

For example: If you want to use the special characters to display fields with specific attributes, begin and end the selected words as illustrated above.

|General Use| will appear as General Use

~Additional Features~ will appear as **Additional Features**

¢Special Considerations¢ will appear as **Special Considerations**

## Updating the Help File

Once the General Instructions have been defined, the help instructions for the program must be rebuilt in order to include them. There are two ways to do this.

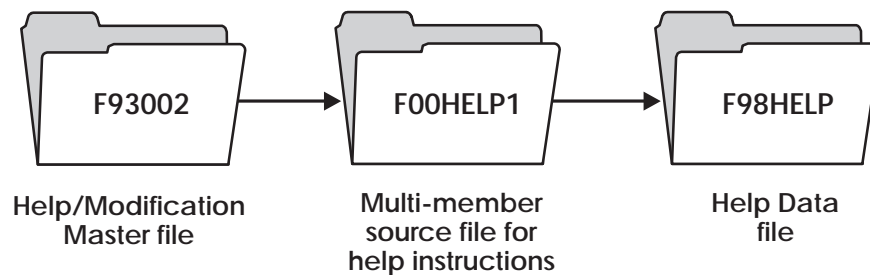
### ► To update the help instructions

---

Regenerate the program with option 15 from the Software Versions Repository

OR

Rebuild the help instructions with option 18 from the Software Versions Repository



If you review the F00HELP1 file via menu G92 – Help Instructions, you will notice that some directional statements have been added to the general instructions you created. They will be formatted as follows:

++ HELP100 , ++ HELP200 , and so on.

The rebuild automatically generates these statements in order to categorize the help instructions, and to include additional help. For example, it will automatically create field explanations and a list of functions and selections defined for the program. HELP100, 200, and so on, are defined in the Data Dictionary.

The end result:

92801

Sample Program - Item Mtc

Action Code. . I  
Cost Center. . 310

O	Item	Item	Quantity	Ship	
P	Number	Description	On Hand	UM	Date Ty

: P92801

Sample Program - Item Mtc

Skip to. . . .

See Memo

Help Task List

General Instructions

Print Request Submitted to Batch

Opt: 1=View 8=Print Task F21=Print F24=More

Opt: 1=Item Master F4=More Detail F24=More Keys

Select General Instructions to view the help instructions created.

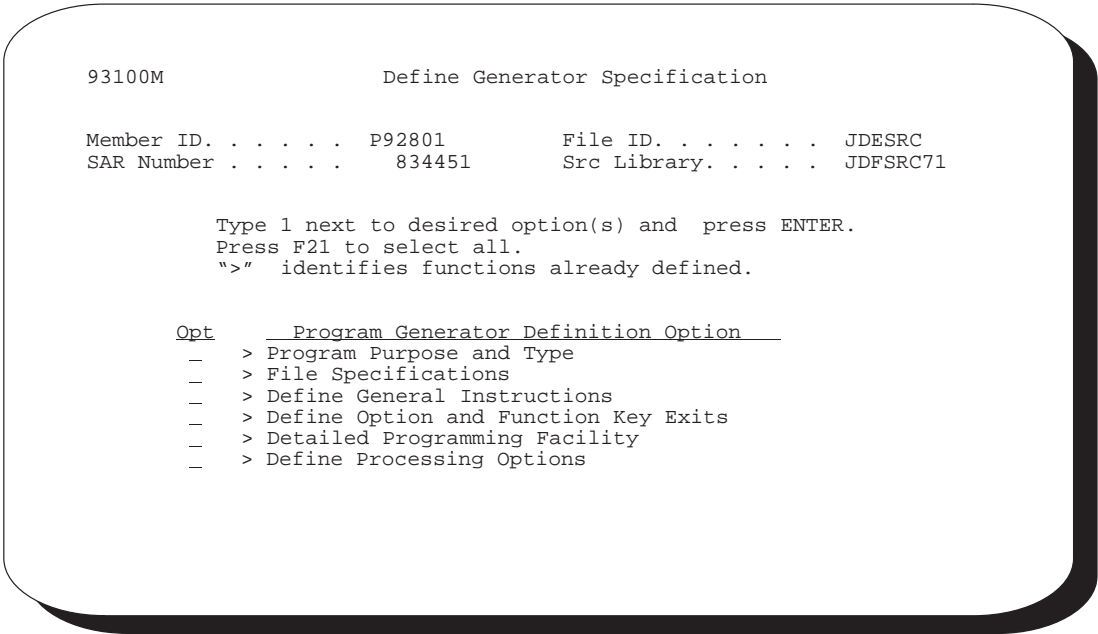


# Define Option and Function Key Exits

## About Option and Function Key Exits

The program generator automatically adds the standard function keys to your program. You can document the keys you want to display on line 24 of the screen through Screen Design Aid (SDA). The program generator creates a list of function keys and options for the program. Pressing F24 on any screen displays the list of function keys and selection exits. To add additional function keys and selection exits to your program, use Define Option and Function Key Exits.

From Define Generator Specification select Define Option and Function Key Exits



Field	Explanation
Define Option and Function Key Exits	Allows the user to define special program exits. <ul style="list-style-type: none"><li>Information is stored in F93104</li></ul>

## Defining Option and Function Key Exits

Allows for the creation of user defined function keys and subfile selection exits for the program being created. Standard function keys such as F3 to exit a program will automatically be created.

### ► To define option and function key exits

---

Complete the Option and Function Key Exits form

- The Function Key Definitions table for the form will automatically be created/updated.

```
93104                                Option & Function Key Exits          Action Code. C
Name: P92801      Item Maintenance

Field  Program Id  Key  Parm 1  Parm 2  Parm 3  Parm 4  Parm 5  Parm 6  Parm 7  Parm 8
#S01   P928011     01  SFXIT  _____  _____  _____  _____  _____  _____
Purpose of Exit : Item Master Information
Returned Key Fld . _____ Returned Desc Fld. _____

Purpose of Exit : _____
Returned Key Fld . _____ Returned Desc Fld. _____

Purpose of Exit : _____
Returned Key Fld . _____ Returned Desc Fld. _____

Purpose of Exit : _____
Returned Key Fld . _____ Returned Desc Fld. _____

Purpose of Exit : _____
Returned Key Fld . _____ Returned Desc Fld. _____

F9=Search
```

Field	Explanation
Field	<p>The internal field name assigned to each option and function key.</p> <p>Correlation exists between this field and the Function Key Definitions repository.</p> <p>Maintained in the soft coding server data structure (I00SC).</p> <ul style="list-style-type: none"> <li>• This is a required field.</li> </ul>
Program ID	<p>The program to be called when the specified function key or selection option is chosen.</p> <ul style="list-style-type: none"> <li>• If the program is not in Software Versions Repository, a warning message is displayed.</li> <li>• This is a required field</li> </ul>
Key	<p>Designates the function key or option you want to assign.</p> <p>Enter the full two digits (01, 02, etc...)</p> <ul style="list-style-type: none"> <li>• This is a required field</li> </ul>
Parm 1–8	<p>Designates up to 8 passed parameters required for a program.</p>
Purpose of Exit	<p>Describes the selection</p> <p>Creates comment lines in the RPG program.</p> <p>Updates the Function Key Definitions table for the form.</p> <ul style="list-style-type: none"> <li>• Function key will show up in the F24 window for the program</li> <li>• Selection exits will show</li> <li>• This is a required field</li> </ul>
Returned Key Fld	<p>Causes logic generation to let a returned key pass through the local data area and loads the value in the specified key field.</p> <ul style="list-style-type: none"> <li>• Only valid with the CL program J98LDAKY.</li> </ul>
Returned Desc Fld	<p>Causes logic generation to let a returned description pass through the local data area and loads the value to the designated description field.</p> <ul style="list-style-type: none"> <li>• Only valid with the CL program J98LDAKY.</li> <li>• For more information on using the Returned Key and Returned Desc Fld, see the program level Helps for P93104.</li> </ul>

## What Are the Function Key Exits?



### F9 – Search

- Exits to the Software Search feature.
- The user can enter a program name to show all programs that meet or are greater than the search criteria.

## Passing Parameters



Users should be cautioned when passing form or subfile fields to other programs. The fields passed may be changed by the called program.

If you are not concerned about passing form or subfile fields, use these fields as parameters in the option and function key exit definitions.

If you are concerned about passing form or subfile fields, alternate options include:

- Pass PSxxxx instead of VDxxxx or SFxxxx.
  - This will require a manual source change to the program in order to properly load the PSxxxx field with the form/subfile field, or load by using Program Design Language
- Pass SHxxx instead of VDxxx or SFxxx
  - The user can define the SHxxxx fields as hidden fields on their form and then load them with the proper information through the Detailed Programming facility



# Work with the Detailed Programming Facility

---

## Working with the Detailed Programming Facility

The Detailed Programming Facility allows you to specify data field definition parameters. It lists the files and the fields for the shell program the program generator creates. To work with the Detailed Programming Facility you should be familiar with:

- The Detailed Programming Facility form
- Selection and function key exits
- Full Data Field Parameters
- Accessing Full Data Field Parameters
- Loading VC0 description fields
- Enabling the Data Base Update Function for Subfiles
- Creating \*Entry PLIST Entries
- Protecting fields from being cleared
- Data Dictionary edits
- Creating a partial KLIST for a file

From Define Generator Specification select Detailed Programming Facility

```
93100M                      Define Generator Specification

Member ID. . . . . P92801      File ID. . . . . JDESRC
SAR Number . . . . . 834451    Src Library. . . . . JDFSRC71

Type 1 next to desired option(s) and press ENTER.
Press F21 to select all.
">" identifies functions already defined.

Opt  Program Generator Definition Option
  _  > Program Purpose and Type
  _  > File Specifications
  _  > Define General Instructions
  _  > Define Option and Function Key Exits
  _  > Detailed Programming Facility
  _  > Define Processing Options
```

Field	Explanation
Detailed Programming Facility	Allows the user to specify data field definition parameters for fields included in the screen, the report, and the master file(s). <ul style="list-style-type: none"><li>• This is a required definition</li><li>• Information is stored in F93105</li><li>• Created from File Specifications</li></ul>

## About the Detailed Programming Facility Form

- Data field specific information
- Listed in file order and then in field order within the files
- Provides access to Field Detail and Program Design Language

```

Program Name: P92801
              Item Maintenance
Locate
File Name   : F0001           Business Unit Security
Field Name  : MSFILE         File Name
O
P Purpose   : Business Unit Security
F0001       :
- File Name           MSFILE      02 - Y
- CC - Thru           MSMCUT      03 - Y
- User ID             MSUSER      01 - Y
F92801      : SDM Item Master File
- Item ID             QXXIT       01 Y Y
F92801LA    : LF - Business Unit, Item ID
- Business Unit       VDXCC       01 C Y
- Description - Compressed QXXDC   - - Y
- Description         SFXDS       - - Y
- Date Last Ship      SFXDT       - Y Y
- Item ID             SFXIT       02 Y Y
Opt:  2=Data Dic  4=Field Dtl  6=*PROC  9=Dlt Fld      F24=More

```

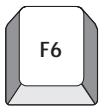
Field	Explanation
Op (Option)	Allows for selection exits for each field.
Purpose	Specifies the file that contains the specified field and the description of the fields the file contains.
Read From	Provides the data field name that contains the data to be placed in the data field referenced in the <i>Write To</i> field.  When *SKIP is specified or when the field is left blank, no logic will be created for that field. Also known as the <i>Source of Data</i> field.
Write To	Specifies the field name as it is identified in the file.
KY PS	Designates the relative position of the field in a key list.  Used in generating key lists (KLIST) definitions.  Used to create a partial key list.

Field	Explanation
RA (Right Adjust)	Designates special scrubbing routines for entry fields. <ul style="list-style-type: none"><li>• Y = Field should be right adjusted</li><li>• N = Field should not be right adjusted</li><li>• C = The field is a business unit and should be left filled with blanks instead of zeros</li><li>• A = The field is an account number and the account number edit routine will be used for editing</li></ul> Can only be used when the <i>Read From</i> field is a video field and the <i>Write To</i> field is a data base field.
DD (Data Dictionary Edit)	Used to override the editing functions from the data dictionary and allow entering custom editing logic.

## What Are the Selection Exits?

Selection	Explanation
2 – Data Dictionary	Exits the user to the Data Dictionary Repository for the data item.
4 – Field Detail	Exits the user to the Full Data Field Parameters screen for more detail on the field.
6 – Program Design Language (*PROC)	Exits the user to the Data Item Formula Revisions screen where the user enters PDL code.
9 – Delete Field	Allows the user to delete a field from the Detailed Programming Facility.

## What Are the Function Key Exits?



### F6 – Repository Services

- Displays a window of J.D. Edwards repositories.



### F10 – Select \*PROC Fields On/Off

- A toggle switch which either displays all fields in the Detailed Programming Facility or just the fields with PDL attached.

## About Full Data Field Parameters

### What Are the Primary Uses?

Loading VC0 description fields

- Utilizes the Field Name, Source of Data, Description File, Description File Key fields

Enabling the database update function for subfiles

- The subfile field that controls data base updates is specified with Entry Optional set to N

Creating the \*ENTRY PLIST code for a program

- Utilizes the PLIST Sequence field

Protecting a field from being cleared every time S001 is executed

- Specifying N in the Clear After (Y/N) field
- Needed for output only fields that do not have a VC0 prefix
- Needed for key fields in RRN program types

Adding user created error messages

Suppressing edits in S005 for audit fields

Creating a partial KLIST for an input file

## Accessing Full Data Field Parameters

- Allows for the creation of additional source code

### ► To access the Full Data Field Parameters

Select Full Data Field Parameters from Detailed Programming Facility

```

93125                               Full Data Field Parameters

Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . F92801      SDM Item Master File
Field Name . . . . QXXIT      Item ID

General Information:
Source of Data . . _____ Dictionary Name. . . XIT
Field Type . . . . _      Data Field Use . . _
Key Position . . . 01      PLIST Sequence . . _
Entry Optional . . Y      Clear After (Y/N). N
Right Adj (Y/N). . Y      Center (Y/N) . . . _
Description File . _____ Descr. File Key. . _____

Editing Information:
Dictionary Edit. . Y
Error Msg No . . . _____ Validation File. . _____
Error Indicator. . _____ Error Index. . . . _
Data Item Type . . S      Data Item Size . . . 8
Decimal Pos. . . . 0      Edit Code. . . . . Z

F3=Return to Subfile / Next Option      F16=File Field Descrip. Window

```

Field	Explanation
Source of Data	What data is to be loaded into the result field. The <i>Read From</i> field on the previous screen.
Dictionary Name	The data item name for the field.
Field Type	Designates the field as a field for a master file, a subfile control record, a subfile record, a report detail, a report heading line, or a report total line.
Data Field Use	How the field is used on a display or report.
Key Position	Designates the relative position of the field in a key list. Used in generating key lists (KLIST) definitions. Used to create a partial key list.

Field	Explanation
PLIST Sequence	<p>The relative position of passed parameters to a program.</p> <ul style="list-style-type: none"> <li>• Used to generate a *ENTRY parameter list</li> <li>• 01–32 are valid</li> <li>• Must enter as <i>01</i> and <b>not 1</b></li> <li>• If the first parameter is passed a non-blank value, an auto-inquiry will be performed.</li> </ul>
Entry Optional	<p>Used with subfile maintenance programs to identify the field that controls database updates.</p> <ul style="list-style-type: none"> <li>• One field needs to be designated as Entry Optional: N</li> <li>• Defaults to a <i>blank</i></li> </ul>
Clear After (Y/N)	<p>Designates whether a field is always cleared at the end of each transaction entry or is only cleared when the user presses the specific function key to clear the screen.</p>
Right Adj (Y/N)	<p>Designates special scrubbing routines for entry fields.</p> <ul style="list-style-type: none"> <li>• Y = Field should be right adjusted</li> <li>• N = Field should not be right adjusted</li> <li>• C = The field is a cost center and should be left filled with blanks instead of zeros</li> <li>• A = The field is an account number and the account number edit routine will be used for editing</li> </ul> <p>Can only be used when the <i>Read From</i> field is a video field and the <i>Write To</i> field is a data base field.</p>
Center (Y/N)	<p>Designates the data should be centered within the field.</p>
Description File	<p>Used in conjunction with loading a VC0 description field.</p> <ul style="list-style-type: none"> <li>• Identifies the file that contains the description</li> </ul>
Descr. File Key	<p>Used in conjunction with loading a VC0 description field</p> <p>Identifies the field name that contains the value whose description is to be retrieved</p> <p>Can also be a key list name</p>
Dictionary Edit	<p>Controls the generation of data dictionary editing for fields in the master file.</p> <ul style="list-style-type: none"> <li>• Defaults to Y</li> <li>• Specifying N will result in no data dictionary editing for the value that is moved to a master file field</li> <li>• Is useful for audit fields such as User ID that can be loaded from the Program Status Data Structure and need no editing</li> </ul>
Error Msg No	<p>Identifies a custom error message to use when errors are detected on a video field.</p> <ul style="list-style-type: none"> <li>• Loads the value in array EMK of subroutine S999</li> </ul>

Field	Explanation
Validation File	Specifies the file name to use for validating the current data field contents.  File name is automatically provided from the data dictionary if it exists.
Error Indicator	Specifies the indicator assigned to the reverse-image, high intensity attributes with the screen definition.  Automatically loaded from the data field generation program.
Error Index	Identifies the specific array index in the EMK array that contains the custom error number.
Data Item Type	Specifies the field type, such as alpha or numeric data.
Data Item Size	Specifies the data field size.
Decimal Pos	Specifies the number of decimal positions contained in numeric data.
Edit Code	Specifies the output formatting options to use for the current data field.

## What Are the Function Key Exits?



### F16 – File Field Description Window

- Pressing F16 will display the File Field Description Window.
- This function key is field sensitive. If your cursor is not on the description file key, the window will pre-load the fields from the description file. The returned value (Opt 4 = Sel) will be placed in Source of Data.
- If your cursor is on the description file key, the window will pre-load the fields from the description file and the returned value (Opt 4 = Sel) will be placed in the Description File Key.



## Loading VC0 Description Fields

The Detailed Programming Facility allows you to specify what file to use to access a description for a video or report description field whose prefix is VC0 (VC0xxx).

```

93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . VC0001
General Information:
Source of Data . . MCDL01      Dictionary Name. . . .
Field Type . . . . P          Data Field Use . . . 0
Key Position . . . .          PLIST Sequence . . .
Entry Optional . . Y          Clear After (Y/N). N
Right Adj (Y/N). . .          Center (Y/N) . . .
Description File . F0006      Descr. File Key. . QXXCC
Editing Information:
Dictionary Edit. . N
Error Msg No . . . .          Validation File. . .
Error Indicator. . .          Error Index. . . .
Data Item Type . . A          Data Item Size . . 30
Decimal Pos. . . . 0          Edit Code. . . . .
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window

```

### ► To load the VCO Description fields

Complete the following three fields:

Field	Explanation
Source of Data	The field name in the description file that holds the description which will be loaded to the VC0 field.
Description File	The file containing the description.

Field	Explanation
Descr. File Key	<p>The field used to CHAIN to the Description File to retrieve the description.</p> <ul style="list-style-type: none"><li>• Should be the field from the master file whose description is to be retrieved (i.e., QXXCC, not MCMCU).</li><li>• For description files that have more than one key field, can be the name of a composite key that will be loaded by the program.</li><li>• If you use the composite key, you will have to add the code to the program to load those fields. You can do this using PDL or SEU.</li><li>• If your description file is F0005, User Defined Codes, the description file key should be the code field. Since the System and Record Type are known to the Data Dictionary, a key list is not necessary.</li></ul>

### Example

In the previous example, QXXCC is a field in the master file that holds a business unit value. F0006 is the file that holds descriptions of business units. MCDL01 is the field in F0006 that holds the business unit description and this description is loaded to VC0001.

In the A52 (or earlier) release, F0006 is a file that is not accessed by a server program, so it is necessary to include it in the File Specifications. Beginning with the A61 release, F0006 is accessed by a server program.

## Example – User Defined Code

If you are accessing a description for a field that is a User Defined Code, the Description File will be F0005 and the Descr. File Key will be the field for which the description is being accessed.

The Program Generator retrieves the Install System Code and User Defined Code Type from the Data Dictionary and builds the composite key to access the User Defined Code file.

```

93125                               Full Data Field Parameters

Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . VC0002

General Information:
Source of Data . . DRDL01      Dictionary Name. . . .
Field Type . . . . P          Data Field Use . . . Q
Key Position . . . .          PLIST Sequence . . . .
Entry Optional . . N          Clear After (Y/N). N
Right Adj (Y/N). . .          Center (Y/N) . . . .
Description File . F0005      Descr. File Key. . QXXTY

Editing Information:
Dictionary Edit. . N          Validation File. . . .
Error Msg No . . . .          Error Index. . . .
Error Indicator. . . .        Data Item Size . . 30
Data Item Type . . A          Edit Code. . . .
Decimal Pos. . . . Q

F3=Return to Subfile / Next Option      F16=File Field Descrip. Window

```

In the above example, QXXTY is a field in the master file that is a User Defined Code (UDC). F0005 is the file that holds descriptions of UDCs. DRDL01 is the field in F0005 that holds the UDC description and this description is loaded to VC0002.

Because F0005 is a file that is accessed with a server program, it is not necessary to include it in the File Specifications.



- Loading description fields using this approach only works if the description field being loaded is a VC0 field.
- Specifying a file does not guarantee that the file you specify will be brought into the File Specifications. You will need to check the File Specifications to make sure the files from which you want to retrieve descriptions are present. The exception would be for files that are accessed with a server program.

## Enabling the Database Update Function for Subfiles

If you are designing a subfile maintenance program, define at least one field in your subfile as Entry Optional *N*.

### ► To enable the database update function for subfiles

On the Full Data Field Parameters form enter "N" in the Entry Optional Field

```

93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . SFXIT      Item ID
General Information:
Source of Data . . . .
Field Type . . . . S
Key Position . . . .
Entry Optional . . N
Right Adj (Y/N). . -
Description File . . . .
Editing Information:
Dictionary Edit. . N
Error Msg No . . . .
Error Indicator. . 43
Data Item Type . . A
Decimal Pos. . . . 0
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window
Dictionary Name. . XIT
Data Field Use . . B
PLIST Sequence . .
Clear After (Y/N). Y
Center (Y/N) . . . -
Descr. File Key. . .
Validation File. . .
Error Index. . . .
Data Item Size . . 8
Edit Code. . . . . Z

```

In the above example, Item ID (SFXIT) is identified as the field that controls database updates.

- If Item ID is blank, but there is a database record for the subfile record, then the database record is deleted.
- If Item ID is not blank, then the database update is either a write or update depending on whether the database record existed in the subfile.
- You must define one or more hidden fields in the subfile record if:
  - The program type uses key processing for the subfile. The hidden field must be SHxxxx, where xxxx is the data dictionary item. There must be an SHxxxx field for each key field that is in the subfile record.
  - The program type uses RRN processing for the subfile. The hidden field must be SH#RRN.

## Creating \*ENTRY PLIST Entries

PLIST entries are used to define which data items are entries in a parameter list.

- Maximum of 32 parms



### To create \*ENTRY PLIST entries

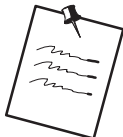
On the Full Data Field Parameters form

```

93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . VDXCC      Cost Center
General Information:
Source of Data . . QXXCC      Dictionary Name. . XCC
Field Type . . . . P          Data Field Use . . B
Key Position . . . .          PLIST Sequence . . 01
Entry Optional . . Y          Clear After (Y/N). Y
Right Adj (Y/N). . -          Center (Y/N) . . . -
Description File . .          Descr. File Key. . .
Editing Information:
Dictionary Edit. . N
Error Msg No . . . .          Validation File. . .
Error Indicator. . 41          Error Index. . .
Data Item Type . . A          Data Item Size . . 12
Decimal Pos. . . . 0          Edit Code. . . .
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window
  
```

Enter a two digit number corresponding to the sequence of the parameter in the PLIST sequence field.

The data item VDXCC will be used as the first parameter in the entry list of Subroutine S999. The program generator will create a field name which is the same data dictionary item prefixed with ##. This parameter field will be moved to VDXCC from the parameter field.



The data item in the display file, not the database file, must be used for creating PLIST parameters.

```

Columns . . . :   1  71          Browse          JDFSRC61/JDESRC
SEU==>          P92801
FMT C .....CL0N01N02N03Factor1+++OpdcFactor2+++ResultLenDHHiLoEqComments++++
2549.00      CSR          S999          BEGSR
2550.00      C*          -----
2551.00      C*
2552.00      C*      Required program parameters.
2553.00      C*
2554.00      CSR          *ENTRY      PLIST
2555.00      C*
2556.00      C*      Passed Parameter - Business Unit
2557.00      C*
2558.00      CSR          PARM          ##XCC  12
2559.00      C*
2560.00      C*      Move to internal reference - Cost Center
2561.00      C*
2562.00      CSR          MOVE ##XCC      VDXCC
2563.00      C*
2564.00      C*      Test for auto inquiry function.
2565.00      C*

```

If the parameter value is not blank, the variable \$AUTO will be updated with a “1”. This informs the program to perform an automatic inquiry (S003) when called.

```

Columns . . . :   1  71          Browse          JDFSRC61/JDESRC
SEU==>          P92801
FMT C .....CL0N01N02N03Factor1+++OpdcFactor2+++ResultLenDHHiLoEqComments++++
2554.00      CSR          *ENTRY      PLIST
2555.00      C*
2556.00      C*      Passed Parameter - Cost Center
2557.00      C*
2558.00      CSR          PARM          ##XCC  12
2559.00      C*
2560.00      C*      Move to internal reference - Business Unit
2561.00      C*
2562.00      CSR          MOVE ##XCC      VDXCC
2563.00      C*
2564.00      C*      Test for auto inquiry function.
2565.00      C*
2566.00      CSR          VDXCC      IFNE *BLANK
2567.00      CSR          MOVE '1'          $AUTO  1
2568.00      CSR          END
2569.00      C*-----
2570.00      C*

```



If you are passing parameters, the CL program calling this RPG program will need to pass a blank parameter.

## Protecting Fields from Being Cleared

This feature is useful when creating data entry programs with a repetitive data field. For example, when a date does not need to be keyed except for the first entry.

- All fields except those prefixed with VC0 will be cleared each cycle in Subroutine S001
- Default is Y
- Function Key 22 will clear all fields



### To protect fields from being cleared

Enter "N" in the Clear After field on the Full Data Field Parameters form

```

93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . VC0001
General Information:
Source of Data . . MCDL01      Dictionary Name. . . .
Field Type . . . . P          Data Field Use . . . . 0
Key Position . . . .          PLIST Sequence . . . .
Entry Optional . . Y          Clear After (Y/N). . . . N
Right Adj (Y/N). . .          Center (Y/N) . . . .
Description File . F0006      Descr. File Key. . . . QXXCC
Editing Information:
Dictionary Edit. . . N
Error Msg No . . . .
Error Indicator. . . .
Data Item Type . . A          Validation File. . . .
Decimal Pos. . . . 0          Error Index. . . .
                               Data Item Size . . . . 30
                               Edit Code. . . .
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window
  
```

The data item VC0001 will be cleared only when the clear command is issued.

## What You Should Know About

### User Error Message

Updating error message number and error index will add error to EMK array in Subroutine S999.

```
93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . SFXTY      Item Type
General Information:
Source of Data . . QXXTY      Dictionary Name. . XTY
Field Type . . . . S          Data Field Use . . B
Key Position . . . .         PLIST Sequence . .
Entry Optional . . Y          Clear After (Y/N). Y
Right Adj (Y/N). . -          Center (Y/N) . . . -
Description File . .          Descr. File Key. .
Editing Information:
Dictionary Edit. . N
Error Msg No . . . 1684      Validation File. .
Error Indicator. . .         Error Index. . . . 21
Data Item Type . . A         Data Item Size . . 2
Decimal Pos. . . . 0         Edit Code. . . . .
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window
```

For example: In Subroutine S999, error message 1684 will be loaded to element 21 of the EMK array. Reserved indexes 1 to 20 should not be used.



You will need to modify your source to use this error message index.



## Disabling Data Dictionary Edits

### ► To disable Data Dictionary edits

Enter “N” in the Dictionary Edit option

```

93125                               Full Data Field Parameters
Action Code. . . . I
Program ID . . . . P92801
                               Item Maintenance
File ID. . . . . V92801      Item Maintenance
Field Name . . . . SFXDS      Description
General Information:
Source of Data . . QXXDS      Dictionary Name. . XDS
Field Type . . . . S          Data Field Use . . B
Key Position . . . . _        PLIST Sequence . . _
Entry Optional . . Y          Clear After (Y/N). Y
Right Adj (Y/N). . _          Center (Y/N) . . . _
Description File . . _        Descr. File Key. . _
Editing Information:
Dictionary Edit. . N
Error Msg No . . . . _        Validation File. . _
Error Indicator. . 44         Error Index. . . . _
Data Item Type . . A          Data Item Size . . 30
Decimal Pos. . . . 0          Edit Code. . . . . _
F3=Return to Subfile / Next Option  F16=File Field Descrip. Window

```



This feature is useful if custom validation is added through SEU or PDL.

## Creating a Partial KLIST for a File

- Program Generator loads full key list
- Key position may be changed to exclude subordinate elements

### ► To create a partial KLIST for a file

On the Detailed Programming Facility erase the key position field, starting with the last element.

```

93105                      Detailed Programming Facility
Program Name: P92801
                      Item Maintenance
Locate
File Name   : V92801           Item Maintenance
Field Name  : SFXDS           Description
O
P  Purpose                                     . . . Data Flow . . . KY R D
   F0001 Business Unit Security             Read From Write To PS A D
-   File Name                               MSFILE      02 _ Y
-   CC - Thru                               MSMCUT       03 _ Y
-   User ID                                MSUSER       01 _ Y
   F92801 SDM Item Master File
-   Item ID                                QXXIT        01 Y Y
   F92801LA LF - Business Unit, Item ID
-   Business Unit                           VDXCC         QXXCC      01 C Y
-   Description - Compressed                 QXXDC         QXXDC      _ _ Y
-   Description                             SFXDS         QXXDS      _ _ Y
-   Date Last Ship                          SFXDT         QXXDT      _ _ Y
-   Item ID                                SFXIT         QXXIT      02 Y Y
Opt:  2=Data Dic  4=Field Dtl  6=*PROC  9=Dlt Fld  F24=More

```

For example: The key list sequence for the Business Unit security file is:

- User ID
- File Name
- Through Business Unit

If you wish to use a partial key to access this file, starting with the last element, Field Exit through the key position (KY PS).



If you need both the full key list and a partial key list, you will need to enter this through SEU.

# Define Processing Options

---

## Defining Processing Options

Processing options let individual programs perform in many different ways. They are analogous to mechanical switches that are set before the program is run.

Define Processing Options allows you to define processing options the program can use. After you define the processing options, you can access them through the DREAM Writer versions list and change the processing values for a specific version. To define processing options you must be familiar with the Processing Options Setup form. You must complete the following tasks:

- ☐ Access Define Processing Options
- ☐ Complete the Processing Options Setup form
- ☐ Create Code for the Processing Options.

## What You Should Know About

### **Program Generator**

The program generator does not automatically include code that handles interactive processing options. The program generator does handle some of the tasks for batch programs. The text you enter on the Processing Options Setup form displays prior to running the program using the processing option. This text does not pass instructions to the program generator.

► **To define processing options**

1. From Define Generator Specifications select Define Processing Options

93100M

Define Generator Specification

Member ID. . . . . P92801

File ID. . . . . JDESRC

SAR Number . . . . . 834451

Src Library. . . . . JDFSRC71

Type 1 next to desired option(s) and press ENTER.

Press F21 to select all.

">" identifies functions already defined.

Opt

Program Generator Definition Option

> Program Purpose and Type

> File Specifications

> Define General Instructions

> Define Option and Function Key Exits

> Detailed Programming Facility

> Define Processing Options

Field	Descriptions
Define Processing Options	Allows the user to define processing options the program can use. <ul style="list-style-type: none"><li>Information is stored in F98301</li></ul>

2. Complete the Processing Options Setup form

## Processing Options Setup Screen

- Allows the user to create run time parameters
- The processing options are retrieved in the Housekeeping subroutine (S999) and loaded into array @OP
- You are responsible for adding specifications and/or source code to use the processing options

```

98304                                Processing Options Setup
Action Code. . . I                  Form ID. . . . . P92801
                                   Item Maintenance

Seq      Text                      Opt Date R Text D O Field
      Nbr (0/1/ J Only L P Name
1  DISPLAY OPTIONS:                1  0  0  1  -  - 
2  -----                        1  0  0  1  -  - 
3  Enter a '1' to inhibit display of items 1  0  0  0  -  - 
4  whose Last Ship Date is after today's  1  0  0  1  -  - 
5  date. Default of blank will display  1  0  0  1  -  - 
6  all items regardless of their Last Ship 1  0  0  1  -  - 
7  Date.                            1  0  0  1  -  - 
008                                     -  -  -  -  -  - 
009                                     -  -  -  -  -  - 
010                                     -  -  -  -  -  - 
011                                     -  -  -  -  -  - 
012                                     -  -  -  -  -  - 
013                                     -  -  -  -  -  - 
014                                     -  -  -  -  -  - 
015                                     -  -  -  -  -  - 

Opt:  1=Insert Blank Lines  2=Resequence  9=Delete Line  F18=Lang Text

```

Field	Explanation
Seq	Specifies how the processing option text lines should be ordered on the screen. Not input capable.
Text	The narrative text for the processing option line.
Opt Nbr	Specifies to the program the position for each processing option.  After the number is assigned, you should never change it.  This field is not input capable for existing lines of text.
Date (0/1)	Specifies whether the option refers to a date. 0 Not a date 1 Date is to be stored as a Gregorian date in standard M/D/Y 2 Date is to be stored as a Julian date 3 Date is to be stored in YYYY/MM/DD (allows entry of full year to accommodate new century)

Field	Explanation
R J	Determines if the entry field is right-justified. blank or 0 Information is not right-justified 1 Information to be entered is numeric and should be right-justified 2 Information to be entered is to be right-justified and left-filled with blanks
Text Only	There may be multiple lines of text for any given processing option. 0 Line is a valid processing option entry line 1 Line is only for text  For each processing option there must be one and only one line with a value of 0.
O P (Optional)	Allows the user to delete or resequence existing lines or add blank lines. 1 Insert blank line 2 Resequence all lines 9 Delete line
Field Name	Points to the Data Dictionary definition and is used during data entry to edit field size and other field attributes.  User must enter the full field name or indent two spaces and then enter the data item because the first two characters are stripped off to determine the data item.  Permits F1 to be functional on the processing option field.
Display Level	Used to secure processing options from view.

## What Are the Function Key Exits?



### F6 – Repository Services

- Displays a window of J.D. Edwards repositories



### F18 – Language Preference Text

- Displays a screen that is used to enter language specific processing options

## Creating Code for Processing Options

### **To create code for the processing options**

---

1. From Software Versions Repository locate the program for which you are adding processing options. View the RPG code.
2. Enter “1” next to the line in the subfile for the program.
  - The code for the program appears.
3. Scan for the following instances within the code:
  - Where you instruct the compiler to bring in the required source for the Extension Specification related to the C9803 subroutines.
  - Where you interpret and act upon the values entered in the processing options.
  - Where you instruct the compiler to copy the source for the calculation specifications related to the C9803 subroutine.

## What Happens When You Define Processing Options?

1. Defining processing options will automatically make the following changes to the RPG source code:
  - Create an O record type in file F98301.
  - Bring in /COPY statement for E81DRPT.
  - Bring in /COPY statement for C81DRPT.
  - Bring in EXSR C81DRPT statement in the Housekeeping subroutine (S999).
  - Processing options are loaded to array @OP, which has 99 elements of 25A.
2. To use the processing options in a program, you must add code manually via SEU or PDL.
  - Source code must be added in the Housekeeping subroutine (S999) to move the processing option into a program work field.

Example:

```
        MOVEL @OP,1    $PO1    2
```

- The program work field can then be used with PDL, or you can manually add source code to the program via SEU.

PDL example:

```
\ If document type is blank,      \  
\ use Processing Option as default \  
begin  
    If VDDCTO = ' ' Then  
        VDDCTO := $PO1;  
    end
```

3. To retrieve the processing options, your program needs to have values defined for PSPID and PSVERS.
  - If your program is a report program, PSPID and PSVERS are automatically generated as PLIST parameters. Therefore, you do not need to do anything.
  - If your program is an interactive program, you will need to add PSPID(10) and PSVERS(10) as PLIST parameters. Remember to modify any programs that call this program so that these two parameters are passed.



## Example – Programs Using Processing Options

The following is an example of an interactive program containing processing options.

1. Using the Software Version Repository, inquire on P928200
2. Using the search option in SEU, search for string C9803. The following screen displays.

```
Columns . . . :    1  71          Browse          DEVSRC/JDESRC
SEU==>>          P928200
0060.00          E*
0061.00          E*      Copy Member for Composite Common Subroutine - C9803
0062.00          E*
0063.00          E/COPY JDECPY,E9803
0064.00          E*****
0065.00          I*****
0066.00          I*      PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES
0067.00          I*
0068.00          I*
0069.00          I*      Data Structure to Load Video Screen Text
0070.00          I*
0071.00          IDSTXT      DS                      640
0072.00          I              1  14 VTX001
0073.00          I              41  41 VTX002
0074.00          I              81  81 VTX003
0075.00          I             121 134 VTX004
0076.00          I             161 168 VTX005
0077.00          I             201 230 VTX006
0078.00          I             241 248 VTX007
0079.00          I             281 297 VTX008
0080.00          I             321 322 VTX009
String c9803 found.
```

This code copies the E Specs related to the common subroutine. This code defines all necessary arrays and tables for the copy module C9803. Typically, you add copy statements such as this at the end of the E Specs.

3. The next section of code in this program which relates to processing options is the following:

```

Columns . . . : 1 71          Browse          DEVSRC/JDESRC
SEU==>>          P928200
1423.02          CSR          MOVE '001'      PSVERS 3
1424.00          CSR          EXSR C9803
1425.00          C*
1425.01          CSR          MOVE@OP,1      $DSPSD 1
1425.02          C*
1425.03          C*          Default for processing option for display of records with
1425.04          C*          a last ship date after today's date
1425.05          C*
1425.06          CSR          $DSPSD      IFNE '1'
1425.07          CSR          $DSPSD      ANDNE ' '
1425.08          CSR          MOVE@OP,1      $DSPSD
1425.09          CSR          END
1425.10          C*
1426.00          C*
1427.00          C*
1428.00          C*          Key list for - Business Unit Security
1429.00          C*
1430.00          CSR          MSKY01      KLIST
1431.00          CSR          KFLD          MSUSER
1432.00          CSR          KFLD          MSFILE
1433.00          CSR          KFLD          MSMCUT
String c9803 found.

```

@OP is the array of returned values for the processing options. @OP,1 contains the first processing option value, which is the value you give when you select the first processing option. It is then moved into another program field for usage.

4. The next section of code which relates to processing options is as follows:

```

Columns . . . : 1 71          Browse          JDFSRC/JDESRC
SEU==>>          P928200
1514.00          C*
1515.00          C/COPY JDECPY,C9803
1516.00          C*****
1517.00          C*****
          ***** End of data *****

```

This code copies the C Specs for C9803. This code is the actual subroutine C9803 that performs the retrieval of processing option values for the DREAM Writer version you execute.

## Example – Report Program Using Processing Options

The program generator builds segments of code that are required for handling processing options. The code related to report processing options exists in two locations in the example program. The locations are:

- Where the program generator copies the Extension Specifications related to the common subroutine for retrieving processing options.
- Where the program generator copies the Calculation Specifications related to the common subroutine for retrieving processing options.

While in SEU for program 928400, search for a string which includes the common subroutine C81DRPT.

```
Columns . . . : 1 71      Browse      DEVSRC/JDESRC
SEU==>>      P928400
0053.00      E*
0054.00      E*      Copy Member for Composite Common Subroutine - C81DRPT
0055.00      E*
0056.00      E/COPY JDECPY,E81DRPT
0057.00      E*****
0058.00      I*****
0059.00      I*      PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES
0060.00      I*
0061.00      I*
0062.00      I*      Data Structure to Load Video Screen Text
0063.00      I*
0064.00      IDSTXT      DS      680
0065.00      I      1 12 VTX001
0066.00      I      41 52 VTX002
0067.00      I      81 92 VTX003
0068.00      I      121 132 VTX004
0069.00      I      161 190 VTX005
0070.00      I      201 202 VTX006
0071.00      I      241 242 VTX007
0072.00      I      281 310 VTX008
0073.00      I      321 328 VTX009
String c81drpt found.
```

This code copies the E Specs related to the common subroutine and defines all necessary arrays and tables for the copy module C81DRPT.

The following screen shows the next example of code related to processing options in reports.

```

Columns . . . : 1 71          Browse          DEVSRC/JDESCR
SEU==>>          P928400
0934.00          C*
0935.00          C/COPY JDECPY,C81DRPT
0936.00          C*
0937.00          CSR          SETON          OF
0938.00          CSR          MOVE ' '          $$PAGE 1
0939.00          CSR          GOTO END999
0940.00          C*
0941.00          C*
0942.00          C*
0943.00          C*          Process file open errors.
0944.00          C*
0945.00          CSR          T999FE          TAG
0946.00          C*
0947.00          CSR          SETON          LR
0948.00          CSR          MOVE 'JDE9901' #@MSG 7
0949.00          CSR          CALL 'J98CMSG'          81
0950.00          C*
0951.00          CSR          PARM          #@MSG
0952.00          CSR          PARM          #@MDTA
0953.00          CSR          END999          ENDSR
0954.00          C*****
String c81drpt found.

```

This code copies the C Specs related to the C81DRPT subroutine and instructs the program to retrieve all pertinent DREAM Writer information in processing options, level breaks, and totaling for a given version of a Form ID. Any code needed for storing and interpreting the processing option values is added manually after this point. As in the interactive example, the processing option values are loaded into the array @OP by the common subroutine.



# Program Design Language

## Objectives

- To work with Data Item Formula Revisions
- To understand Program Design Language (PDL) statements and syntax
- To understand PDL editing, parsing, and source generation

## About Program Design Language

Program Design Language (PDL) lets you create specifications within the Detailed Programming Facility that causes specialized source code to generate. Use PDL for calculations or comparisons. When the program generates, the program generator converts the code into RPG.

Any code written in PDL comes before the standard code that the program generator creates. If you want the PDL code to come after the standard code for a field, place the PDL code on the field immediately following the field it is associated with. The program generator creates all source code for fields in alphabetical order.

PDL is stored in the F93109 file with one record per formula. File F93109 divides into statements in the F93110 file. The F93110 file contains multiple records for each formula.

PDL checks variable definitions as follows:

- Checks the variable to see if it is a keyword
- Checks for the variable in the RPG program
- If not in the program, checks to see if it exists in the Data Dictionary Repository
- If not in the Data Dictionary Repository, the user must define the variable



The Program Design Language covers many areas including:

- The Data Item Formula Revisions form
- PDL Statements
- Blocks of Statements
- Comments
- Assignments
- Database Operations
- Calls
- Loops
- Conditions
- Miscellaneous Keywords and Syntax

Perform the following tasks:

- ☐ Work with Data Item Formula Revisions
- ☐ Understand PDL Statements and Syntax
- ☐ Understand Additional PDL Operations

---

```

93105                               Detailed Programming Facility
Program Name: P92801
                               Item Maintenance

Locate
File Name      : F0001                Business Unit Security
Field Name     : MSFILE                File Name

O                               . . . Data Flow . . . KY R D
P   Purpose                               Read From   Write To   PS  A  D
-   F0001                Business Unit Security
-   File Name                _____   MSFILE       02  _  Y
-   CC - Thru                _____   MSMCUT        03  _  Y
-   User ID                  _____   MSUSER        01  _  Y
-   F92801                SDM Item Master File
-   Item ID                  _____   QXXIT          01  Y  Y
-   F92801LA    LF - Business Unit, Item Id
-   Business Unit            _____   VDXCC          01  C  Y
-   Description - Compressed _____   QXXDC          _  _  Y
-   Description              _____   SFXDS          _  _  Y
-   Date Last Ship          _____   SFXDT          _  Y  Y
-   Item ID                  _____   SFXIT          02  Y  Y
Opt:  2=Data Dic   4=Field Dtl   6=*PROC   9=Dlt Flg   F24=More

```

Option	Explanation
6 – Program Design Language (*PROC)	Exits the user to the Data Item Formula Revisions screen where the user enters PDL code.

## Understanding the Data Item Formula Revisions Form

93109

Data Item Formula Revisions

Action Code. . . . . A  
Program ID . . . . . P92801  
File ID. . . . . F92801LA  
Field Name . . . . . QXXCC

Data Item Formula

F5=Variables

F6=Repository Services

F24=More

Field	Explanation
Program ID	The RPG program name defined in the Software Versions Repository Master file.
File ID	The name of a given file.
Field Name	Specifies the field name as it is identified in the file.
Data Item Formula	A set of Program Design Language (PDL) statements, which are then translated into RPG code.



# Understand PDL Statements and Syntax

---

## Understanding PDL Statements and Syntax

A Data Item Formula consists of PDL statements. PDL statements perform the following types of operations.

- Blocks of statements
- Comments
- Assignments
- Database operations
- Program calls
- LOOPS
- Conditions
- Miscellaneous

You must use specific syntax when you work with PDL statements.

## Understanding PDL Statements

A PDL statement is formed by combining one or more of the following elements:

- Keywords
- Variables
- Database File name
- Assignment operator
- Arithmetic operator
- Constants
- Punctuation

### Keywords

Keywords make up the “vocabulary” of PDL. They identify the type of operation that is performed by the statement. The keywords, their syntax and rules, and some examples are presented in the next section.

### Variables

The following are valid variable names in PDL statements:

- Database field names

Examples: ABAN8, MCDL01

- Screen and report field names

Examples: VDDOCO, SFTRDJ, VC0001, RR#CLS

- Data Dictionary

Data Dictionary fields may be used in PDL. Their data type and size will be used as defined.

- Indicators

Indicators are referenced by using the names IN01 to IN99. INLR may also be used. They may be used in PDL assignment statements to set on or off, and in conditional expressions to test for on or off.

Example:

```
in98 := '0'  
If in98 = '0' Then
```

Note that with PDL the \* is not used with indicators; that is indicator 01 is specified as in01 and not \*in01.

- Program Workfields

Any name that PDL recognizes as a variable, but isn't a database field name, Data Dictionary field, screen or report field name, or indicator is considered a program work field. PDL will prompt you to define its data type.

Examples: \$#am1, \$#xtp, \$po1

Since source code has not been generated, PDL is not able to search the source code to find a definition.

### Database Files

A database file name used in one of the database I/O statements MUST have been defined in the File Specifications. PDL does NOT add file names to the specifications.

### Operators

The valid assignment and arithmetic operators are defined in the Keyword section.

### Constants

Alpha constants are specified by enclosing them in single quotes. Numeric constants are specified without quotes.

Examples:

```
vc0001 := 'Proof Mode';  
$#am1 := 0;
```

PDL does NOT recognize the RPG constants such as \*BLANK or \*ZERO.

### Punctuation

- The basic PDL punctuation is a semi-colon (;).
- PDL statements must be separated by the semi-colon.

## Understanding Blocks of Statements

### Keywords and Syntax

Field	Explanation
Begin	Initiates a block of statements. The syntax is: Begin
End	Terminates a block of statements initiated by the Begin statement. The syntax is: End

### Rules

1. All Data Item Formulas must be contained within a Begin...End block. A comment statement may precede the Begin statement.
2. All statements within a Begin...End block must be separated by a semicolon.
3. Begin...End may be nested to a maximum of 50 levels.

For example: **Rule 1**

```
\ Use system date as default. \  
Begin  
    If vdtrdj = ' ' Then  
        vdtrdj := $$edt;  
End
```

For example: **Rule 2**

```
\ Load A/B name to vc0 field. \  
Begin  
    aban8 := q3an8;  
    chain f0101la;  
    If in98 = '0' Then  
        vc0003 := abalph;  
End
```

For example: **Rule 3**

```
\ Computer counter. \  
Begin  
  If zac1st = '900' Then  
    Begin  
      rr#nin := ' 1';  
      $#nin := 1;  
    End;  
  If zac1st < '900' Then  
    Begin  
      rr#nin := ' 0';  
      $#nin := 0;  
    End  
  End  
End
```

## Understanding Comments

### Keywords and Syntax

Field	Explanation
\ (backslash)	Initiates and terminates a comment.  The syntax is: \ text \  All comments must be enclosed within a pair of backslashes.

### Rules

Comment lines must not exceed 50 characters.

For example: **Initial Comment**

```
\ Compute extended amount. \  
Begin  
  $#xtp := q2xqt * q2uncs;  
End
```

For example: **Embedded Comment**

```
Begin  
  $#am1 := 0;          \ Order Total \  
  $#xtp := 0;          \ Extended Amount \  
End
```

## Understanding Assignments

### Operator and Syntax

Operator	Explanation
:=	The assignment operator. The syntax is: variable := expression;
+	Add
-	Subtract
*	Multiply
/	Divide
	Concatenate
>	Blank and Concatenate
<	Truncate and Concatenate
SST	Substring The syntax is: variable := SST (field,n1,n2) n1 = start position n2 = length of string

### Rules

Standard notation using parentheses is allowed for arithmetic operations.

For example:

```
in98 := '0';  
vdremk := 'NOT DEFINED';  
sfttdj := $$edt;  
$#am1 := $#am1 + (qzqty * qzcst);  
$#wrk := 100;  
abalph := vd#fnm |> vd#lnm;  
$cc := SST (qxxcc,3,10)
```

## Understanding Database Operations

### Keywords and Syntax

Keyword	Explanation
Chain	Provides for random data base processing. The syntax is: CHAIN file;
Delete	Provides the ability to delete a specific data base record. The syntax is: DELETE file;
Poseq	Provides for the positioning of a pointer to a specific data base record that is equal to the key value or greater than the key value specified. The syntax is: POSEQ file;
Posgt	Provides for the positioning of a pointer to a specific data base record that is greater than the key value that is specified. The syntax is: POSGT file;
Read	Provides for sequential data base processing by reading the next record in the designated file. The syntax is: READ file;
Readc	Provides for processing of externally described workstation files to obtain the next changed record in a subfile. They syntax is: READC file;
Reade	Provides for keyed sequential data base processing of the designated file. The syntax is: READE file;
Readp	Provides for sequential data base processing of the previous record in the designated file. They syntax is: READP file;
Update	Provides the ability to update specific data base records. The syntax is: UPDATE file;
Write	Provides the ability to write specific data base records. The syntax is: WRITE file;

### Rules

1. The file specified on the statement must be defined to the program in the File Specifications.
2. The Chain, Poseq, Posgt, and Reade statements use the default KLIST name that is generated for the specified file.
  - Each field of the KLIST should be assigned prior to performing the statement.
3. Indicator 98 is specified on the statements to signal that no record was returned to the program.
4. Indicator 99 is specified on the statements to signal that an error occurred on the database operation. For example:

```
\ Load A/B name to vc0 field. \  
Begin  
    aban8 := q3an8;  
    chain f0101la;  
    If in98 = '0' Then  
        vc0003 := abalph;  
End
```



## Understanding Calls

### Keywords and Syntax

Keyword	Explanation
Call	Allows you to execute another program. The syntax is: CALL variable;
Parm	Provides for passing parameters to a program being executed by the Call statement. The syntax is: PARM variable;

### Rules

1. Neither the Call statement nor the Parm statement allow the use of constants.
  - Prior to the Call statement you must enter an assignment statement to load a variable with the name of the program to be called, and load one or more variables with the values of the parameters.
2. The Parm statements must immediately follow the Call statement.

For example:

```
Begin
    ##pid := 'P1540    ';
    ##vers := 'ZJDE001';
    ##doco := nrdoco;
    Call ##pid;
    Parm ##pid;
    Parm ##vers;
    Parm ##doco;
End
```

## Understanding Loops

### Keywords and Syntax

Keyword	Syntax
Until	<p>Provides for loop processing where a condition is evaluated at the bottom of the loop.</p> <ul style="list-style-type: none"><li>• Will translate to DOU in the RPG code.</li></ul> <p>The syntax is: UNTIL (condition) DO (Statement)</p>
While	<p>Provides for loop processing where a condition is evaluated at the top of the loop.</p> <ul style="list-style-type: none"><li>• Will translate to DOW in the RPG code.</li></ul> <p>The syntax is: WHILE (condition) DO (Statement)</p>

### Rules

1. The Do keyword is an integral part of the loop statement.
2. The statement following Do may be a single statement, or a block of statements contained in a Begin...End pair.
3. The condition is simply two expressions separated in a relationship.

For example:

```
Begin
  While in98 = '0' Do
    Begin
      $#xtp := q2xqt * q2uncs;
      $#am1 := $#am1 + $#xtp;
      reade f59422;
    End
  End
```

## Understanding Conditions

### Keywords and Syntax

Keyword	Explanation
If	<p>Provides for conditional processing.</p> <ul style="list-style-type: none"> <li>Remember, the condition is two expressions separated by a relationship</li> <li>The data types of the expressions have to match — alpha to alpha, numeric to numeric.</li> </ul> <p>The Then keyword is an integral part of the If statement which signifies the action to be taken if the condition is met.</p> <ul style="list-style-type: none"> <li>The statement following the Then keyword may be a Begin/End pair to allow for a block of statements when the condition is met</li> <li>In addition, the user may enter <i>Else</i> following the statement(s) entered with If (condition) Then (statement). The syntax is: IF (condition) THEN (statement) ELSE (statement)</li> </ul>
Then	An integral part of the If statement and specifies the starting point for all actions to be taken when the condition of the If statement is met.
Else	May be entered following the statement(s) that are entered with the <i>If (statement) Then (statement)</i> . These statements will be executed when the condition of the If statement is not met.

### Symbols

=	Equal
≠	Not Equal
>	Greater Than
<	Less Than
>=	Greater Than or Equal To
<=	Less Than or Equal To

### Rules

1. The semicolon (;) is not required to end the statement following the Else, or the Then when there is no Else.
2. If Begin/End are to be nested between the Then and Else, the semicolon should be used after each individual statement but not following the End.

For example: **Rule 1**  
**A simple If...Then**

```
Begin
  If sfttdj = ' ' Then
    sfttdj := $$edt
  End
```

For example: **Rule 2**  
**If...Then with a Begin...End**

```
Begin
  If zaclst = '999' Then
    Begin
      rr#nin := '1';
      $#nim := 1;
    End
  End
```

For example: **Rule 1**  
**If...Then...Else**

```
Begin
  If in98 = '0' Then
    vc0003 := abalph
  Else
    vc0003 := 'NOT DEFINED'
  End
```

For example: **Rule 2**

**If...Then...Else with Begin...End**

```
Begin
  If zaclst = '900' Then
    Begin
      rr#nin := '0';
      $#nin := 0;
    End;
  Else
    If zaclst < '900' Then
      Begin
        rr#nin := '<0';
        $#nin := 1-;
      End;
    Else
      Begin
        rr#nin := '>0';
        $#nin := 1;
      End
    End
  End
End
```

## Understanding Miscellaneous Keywords and Syntax

### Keywords and Syntax

Keyword	Explanation
Include	<p>Provides the ability to include other PDL modules in the User Calculation.</p> <p>The syntax is: INCLUDE (module name)</p>
Return	<p>Provides for user specification of the <i>Source of Data (Read From)</i> field alone or as the result of a series of procedures.</p> <p>The syntax is: RETURN variable;</p>

### Rules for Include

1. PDL modules can be kept in the form of a copy book by designating \*FORMULA in the Program ID field and \*LIBRARY in the File ID field.
2. The include module should have a unique name (i.e. @CONCAT).
3. It is J.D. Edwards naming convention to begin module names with the @ symbol.
4. Will cause the Program Generator to automatically generate the appropriate code for the include module. This will prevent the need to reenter user calculations that are needed in numerous programs.

The following page illustrates an include module and the include statement that calls the module.

F5=Variables                      F6=Repository Services                      F24=More

F5=Variables      F6=Repository Services      F24=More

### Rules for Return

1. Specifying the Return keyword is the same as entering \*PROC in the *Read From* field in the Detail Programming Facility.
2. All standard processing for this data field will be omitted. In other words, by specifying the Return keyword, the code generated by PDL will be used *instead* of any standard logic. For example:

```
begin
  ##b1 := 0;
  ##b1 := q2xqt * q2uncs;
  return ##b1
end
```



# Understand Additional PDL Operations

---

## Understanding Additional PDL Operations

Additional Program Design Language operations include:

- Editing
- Parsing
- Source Generation

### PDL Editing

The Data Item Formula is one long continuous field. If there is an error, the entire field is displayed in reverse image, and the cursor will be placed on the field following the error. Error messages can be displayed by using the F7 key.

The maximum number of statements that may be entered is 200.

### PDL Parsing

The Data Item Formula is stored in the File Specifications database in two forms.

- The unparsed form is stored in F93109, with one record for one formula.
- The parsed form is stored in F93110, where there are multiple records for each formula, and each record corresponds to an RPG operation code.

### PDL Source Code Generation

The PDL code will be merged into the program based on what field the PDL is entered for in the Detailed Programming Facility:

Read From	Write To	Affected
1) QXXIT	VDXIT	S004 (moves database fields to video fields)
2) VDXIT	QXXIT	S005 (moves video fields to data base fields)

The code generated by PDL is placed before the standard code generated by the Program Generator for the field in the Write To column. If you want the code

generated by PDL to REPLACE the standard code, then enter \*PROC in the Read From column.

- This action must be considered very carefully, as no editing or formatting of the field will be performed, except what you do in the Data Item Formula.

If you want the code generated by PDL to come after the standard code generated by the Program Generator for the field in the Write To column, then you must place the Data Item Formula on the immediately following field in the Detailed Programming Facility.

- Fields are listed in the Detailed Programming Facility in alphabetical order, and this cannot be changed.

### Examples – PDL

Two examples will illustrate the PDL statements and syntax. Both come from an SFL inquiry program.

The first example is a data item formula that is attached to a user defined PDL entry point in subroutine S004. The purpose of the formula is to determine if the database record meets the search criteria for order number. The program work fields \$doco1 and \$doco2 hold the lower and upper values entered on the SFL inquiry search fields.

```
93109                                Data Item Formula Revisions

Action Code. . . . . I
Program ID . . . . . P594213 FOCUS/CASE - Sales Order Inquiry
File ID. . . . . S004-2
Field Name . . . . . PDL01

Data Item Formula
\ Test order number for inclusion. \
begin
  If vddoco $= ' ' Then
    If qldoco < $doco1 Then
      $sel := '0';
  If vd#doc $= ' ' Then
    If qldoco > $doco2 Then
      $sel := '0';
end
```

F5=Variables

F6=Repository Services

F24=More

This example illustrates the following types of PDL statements:

Assignment	<code>\$sel := '0';</code>
Blocks	<code>begin...end</code>
Comment	<code>\ Test order number for inclusion. \</code>
Condition	<code>If q1doco &lt; \$doco1 Then \$sel := '0';</code>

This example illustrates the nesting of conditions.

```
    If vddoco $= ' ' Then
        If q1doco < $doco1 Then
            $sel := '0';
```

Note that the statement separator, the semicolon, is not needed until the outermost If...Then is complete.

The second example is attached to a subfile field that is a computed field. The program is inquiring on sales order header records. The computed field is the order total and is based on the sales order detail records (F59422).

```
93109                                Data Item Formula Revisions

Action Code. . . . . I
Program ID . . . . . P594213      FOCUS/CASE - Sales Order Inquiry
File ID. . . . . V594213      FOCUS/CASE - Sales Order Inquiry
Field Name . . . . . SF#AMI

Data Item Formula
\ Compute Order Total \
begin
    $#ami := 0; $#xtp := 0;                \ initiates work fields                \
    q2doco := q1doco;                      \ q2doco is key field for f59422          \
    poseq f59422;
    read f59422;
    while in98 = '0' do
        begin
            $#xtp := q2xqt * q2uncs;        \ compute extended cost for each item \
            $#xtp := $#xtp / 100;
            $#ami := $#ami + $#xtp;         \ accumulate the extended costs      \
            read f59422;
        end;
        $#ami := $#ami / 100;
    end
end
F5=Variables      F6=Repository Services      F24=More
```

This example illustrates the following types of PDL statements:

Assignment	<pre>\$#ami := 0; \$#xtp := 0; q2doco := q1doco;</pre>
Blocks	Note the begin...end nested within the while...do
Comments	Note the embedded comments as well as the heading comment
Database	<pre>poseq f59422; read f59422;</pre>

Loops

```
while in98 = '0' do  
begin...end;
```



The use of statement separators in the loop statements within the Begin...End block are separated, and then the end statement is followed by a separator.

### Function Key Exits from PDL



**F5 – Display Variable Definitions**



**F6 – Repository Services**





## Source Modifications

### Objectives

- To change source code
- To regenerate source code
- To copy and customize Model Control Language (CL) Programs

### About Source Modifications

After you generate source code, you can make modifications to it by changing the Program Generator steps or via SEU. When changes are made to your CASE specifications, you need to regenerate the source. CASE allows for continual improvement and modification to your original specifications.

Perform the following tasks:

- ☐ Change Generated Source Code
- ☐ Regenerate Source Code
- ☐ Work with Model Control Language Programs







# Change Generated Source Code

---

## About Changing Generated Source

After you generate code for a specific program, you can enter any necessary modifications to the code by using the Source Entry Utility (SEU). The program generator maintains a copy of all modifications in a separate file. When you make changes to the files specified for the program, or the detailed field definition parameters, you can regenerate the program source to reflect the changes and maintain your modifications.

The Program Generator indexes the modification lines based on the data the system maintains in column positions 81 through 113. After the program generates, it merges the modification lines in to the generated code according to their index values.

## Changing Generated Source

There are two different methods to change generated source code.

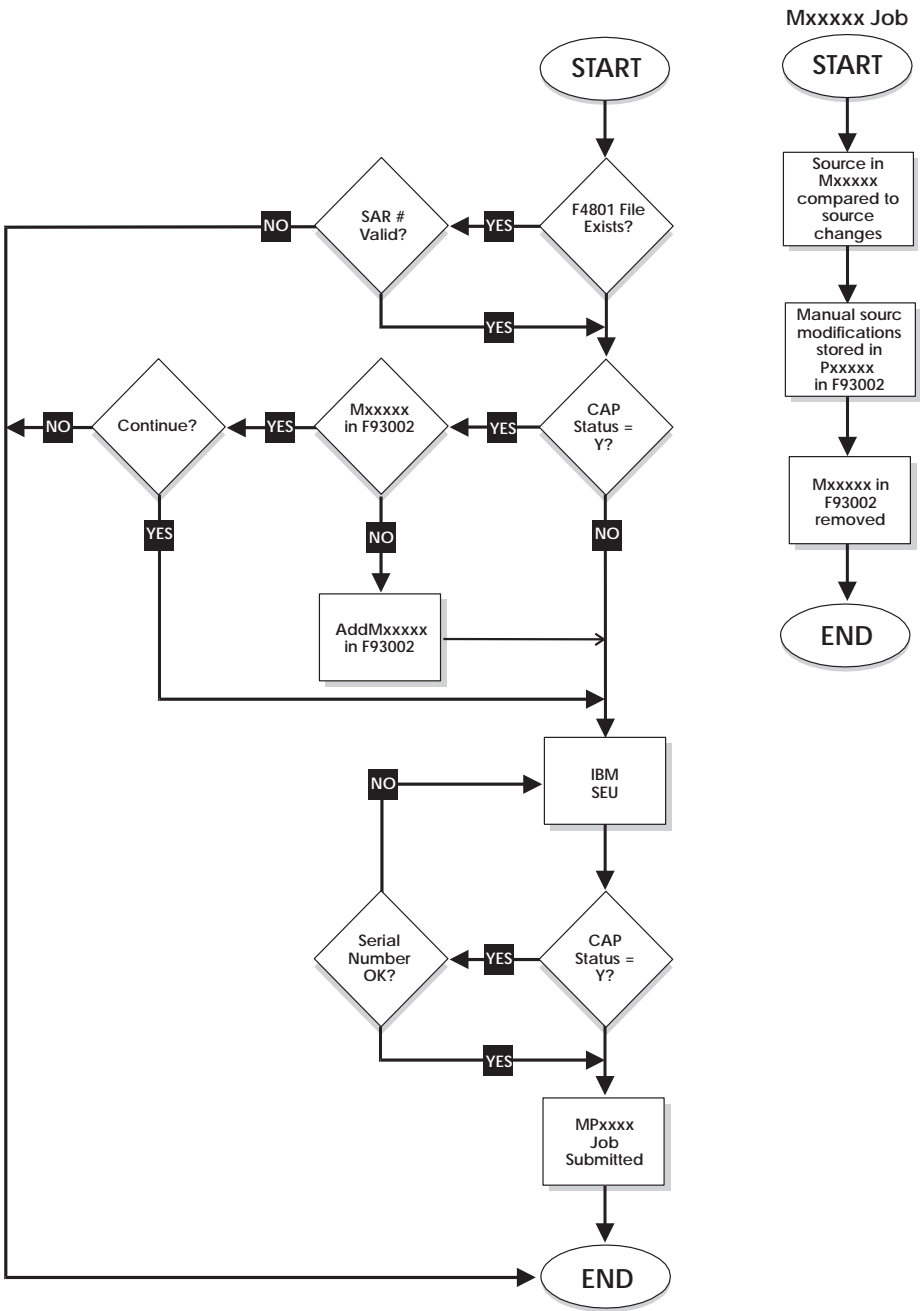
### ► To change generated source code

---

Select one of the following methods:

- From Software Versions Repository inquire on program member and select the edit option to access the J.D. Edwards SEU feature
  - If you change your program via SEU, you do not have to regenerate. You must only recompile at that point.
- Access the Program Design Language feature of the Program Generator
  - SEU changes are monitored by a front-end J.D. Edwards program – MPxxxxxx
  - The MPxxxxxx job does not have to finish before you recompile
  - SEU changes are automatically merged at time of program generation (*not* compilation)
  - All SEU changes are stored in the 'Pxxxxxx' member in F93002 file
  - Source code lines that are moved or copied must have their serial number blanked out from column 80 on
  - All SEU changes can be viewed via option 30 from the Software Versions Repository
    - Columns 1 and 2 will contain a 21 for added lines, 22 for changes, and 23 for deletions

Pre-SEU and Post-SEU Process





# Regenerate Source Code

---

## Regenerating Source Code

When regenerating source code you should know:

- When to regenerate source code
- How to solve a generation error

## When to Regenerate Source Code

You should regenerate a program whenever a program specification has been modified. You should regenerate your programs when you:

- Change the program type
- Add or remove a file from the program file specifications
- Change the content of a file
  - You must repeat the File Specifications step in this situation to put the new information into the Detailed Programming Facility.
  - You must access the File Specifications form from the Define Generator Specification screen and press enter to submit the Detailed Field Specifications interactive job. This procedure applies when you add, change, or remove a file referenced in the program or change the program type.
- Add or change a function exit or selection exit
- Change a data field definition parameter, for example:
  - Add, change, or delete a PDL
  - Add, change, or delete the Full Data Field Parameters
- Add Processing Options to a program that previously didn't have any, or delete all Processing Options
- Change the action lockout codes

## Changing CAP Status

If you change the CAP Status, the system deletes your SEU modifications that were saved in the P member in the F93002 file. JDE recommends that you do not change CAP Status unless the modifications made to your program become unmanageable. When the CAP Status = Y, you can regenerate your program from one J.D. Edwards release to the next.

Change the CAP Status to N when either of the following occur:

- You test the program and are ready to move it into production.
- You must make large manual modifications that the program generator cannot generate for you. For example, adding special subroutines or complicated calculation logic.

### ► To change CAP status

---

1. From the Program Design Aid form inquire on the program to change.
2. Enter 10 next to the location of the member to select the Define Generator Specification option.
3. Select Program Purpose and Type and press Enter.

```
93100                                Program Purpose and Type

Action Code. . . C

Program ID . . . P92801
Title. . . . . Item Maintenance
Purpose
To allow for the addition, revision, deletion, and inquiry of items based upon
their own business unit.
_____
_____
_____

Install System 92                    SAR Number . . . 672835
CAP Status . . . Y                    Program Type . D0040
                                         SFL/T/F - w/Act - w/Sel - Keys

Lockout Act. . _ _ _ _ _

                                         F11=Pgm Type Selection   F2=Program Type X-Ref
```

4. In the CAP Status field, enter N. The Delete Generator Modifications form displays.

93100DM

Delete Generator Modifications

File ID. . . . . F93002  
Src Library. . . . . JDFCLONE7  
Member ID. . . . . P92801

You have changed the program type or the CAP Status of the member listed above. This change requires the deletion all source modifications.  
Press F6 to continue with this change.

F6=Delete F3=Cancel

5. To remove the modifications member, press F6. The Define Generator Specification screen displays.

## Solving Generation Problems

The Program Generator will verify prior to each source generation that the previous job completed normally. When this generation does not complete normally or if you delete the CASE specifications for a program, an error message will be sent to your workstation that says:

- CAP Status Invalid for program Pxxxx ... generation terminated

If you receive a message in the job log indicating that the buffer length is longer than record, or field AGSRCS is not found:

- Verify that the JDESRC file in your source library has been created with a length of 142 and 8 fields.



---

### **To correct the CAP Status Invalid Error**

---

1. Make sure the CAP status is set to *Y* in the Program Purpose and Type
  - Any job that prevents the MPxxxxx job from completing normally will change the CAP Status to *N*
    - Allow the MPxxxxx job to complete
    - Do not cancel it in the job queue
  - If you change the CAP Status to *N*, all SEU modifications ( program member in F93002) will be deleted
  - If the file specifications step ended abnormally, the CAP Status will change to *D*
  - Change the status back to *Y* and reprocess the file specifications
2. Make sure the Pxxxxx member does exist in F93002
  - The Pxxxxx member must exist in order to generate a program
  - The Pxxxxx member is initially created during the Program Purpose and Type definition step
3. Make sure the Mxxxxx member does not exist in F93002.
  - The Mxxxxx member must not exist in order to generate a program
  - Use the RMVM command to remove this member
4. Make sure that you are not trying to complete one step of the generation process before the batch job of another has completed.



# Work with Model Control Language Programs

---

## Working with Control Language Programs

Control language (CL) is the primary interface between the system programmer and the AS/400 operating system. A command is a single control language statement. A series of commands can serve as source statements you can use to create a CL program. You compile the commands into a program the system calls whenever it needs the functions the program provides.

J.D. Edwards provides you with a series of model CL programs that you can copy and change to call the CASE programs you develop. You must use a CL program to add an RPG program as an option on a J.D. Edwards menu.

Perform the following tasks:

- ☐ Copy a model CL
- ☐ Customize a model CL

You use the IBM Source Entry Utility (SEU) to change the source for your CL programs. When adding a new CL member using SEU, you can copy the J.D. Edwards model CLs and tailor their specifications to fit your needs.



You can only view source if source resides on your machine.

## Copying a Model CL

### ► To copy a model CL

1. On the Software Versions Repository screen, inquire on the model.
2. In the OP (Option) field next to the location of the program, enter 3. The source code for the model CL program displays.
3. Fill in the new member name and the copy is performed.
4. Inquire on your new CL program and select option 2 to edit. The source code displays.

The following example illustrates the code for J98MODEL1 using the program Help. You can also view the code using the Software Versions Repository.

```
Columns . . . : 1 71          Browse          JDFSRC61/JDESRC
SEU==          J98MODEL1
FMT **
..... 1 ..... 2 ..... 3 ..... 4 ..... 5 ..... 6 ..... 7
***** Beginning of data *****
0001.00 /* ***** */
0002.00 /* */
0003.00 /* Program. . . . . Jxxxxxxxxx */
0004.00 /* */
0005.00 /* Description. . . . Model Interactive Execution Program */
0006.00 /* */
0007.00 /* Program Revision Log */
0008.00 /* */
0009.00 /* */
0010.00 /* Date Programmer Description */
0011.00 /* */
0012.00 /* xx/xx/xx xxxxxxxxxx SAR # xxxxxxxx */
0013.00 /* */
0014.00 /* ***** */
0015.00 Jxxxxxxxx: PGM
0016.00 /* */

F3=Exit F5=Refresh F9=Retrieve F10=Cursor F12=Cancel
F16=Repeat find F24=More keys
(C) COPYRIGHT IBM CORP. 1981, 1992.
```

The lines in the model that require modifications contain lower-case 'xx'. This design lets you easily scan the code for the 'xx' and insert your changes.

5. To exit, press F3 twice. The previous menu displays.

## Customizing a CL Model

### **To customize a CL model**

---

1. Inquire on the newly created member, and display the source code.
2. Scan for the 'xx' strings that reside where you need to make changes for your specific application. When you finish your customization of the program, exit and save the CL.
3. Compile the program.

## J.D. Edwards Model CL Programs

J.D. Edwards has written a series of model CL programs you can copy and customize to meet your programming needs. The following table describes each model CL program.

<b>J98MODEL1</b>	Serves as a template for all interactive programs that do not retrieve processing options in the CL code.
<b>J98MODEL2</b>	Serves as a template for batch programs that need the DREAM Writer but have no printer file.
<b>J98MODEL3</b>	Serves as a template for interactive programs that need a prompt for parameters.
<b>J98MODEL4</b>	Serves as a template for either batch or interactive programs that require the retrieval of processing options in the CL code, but do not require DREAM Writer selection or sequencing.
<b>J98MODEL5</b>	Serves as a template for batch CL programs that call report programs with fixed selection and sequencing while still passing all printer file overrides, processing options, and page-heading functions to the RPG report program.
<b>J98MODEL6</b>	Serves as a template for batch CL programs that require all DREAM Writer functions.
<b>J98MODEL7</b>	Serves as a template for batch CL programs that require all DREAM Writer functions and call multiple print programs over the same OPNQRYF access path.
<b>J98MODEL8</b>	Serves as a template for batch programs that have a control file.

NOTE: You can create selected model CL programs using the Quick Start CL Generator.



### Exercises

See the exercises for this chapter.



# CASE Programs

## Objectives

- To create CASE programs

## About Creating CASE Programs

Perform the following tasks:

- ☐ Create Subfile Maintenance Programs
- ☐ Create Subfile Inquiry Programs
- ☐ Create Report Programs





# Create Subfile Inquiry Programs

---

## About Creating Subfile Inquiry Programs

You can create subfile inquiry programs that allow a user to process data and run programs using an inquiry form you create. A subfile inquiry form presents a subfile of information, allowing a user to view several records at one time.

The intended use and required entries for a typical Interactive Subfile Inquiry Program (A0010) follow:

### Program Type Description

Use this program type for the creation of an interactive subfile program. This subfile program is inquiry only. This program type processes a single master file by key. Lockout Action Codes are not used. Create a display file prior to generating this program type.

### Display File Definition

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of SDA with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is optional. Define a default cursor location if there is no action code.

### CL Program Definition

Copy and revise model CL Program J98MODEL1 to create a CL program for use with program type A0010. You can use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Input column. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use a key list for record retrieval from the master file. If you are not using the complete key list, update the Key Sequence Field in the Detailed Programming Facility to include only those data items which are needed. This key list should match your key field definition from the control format of the display file.

### Special Considerations

Add special logic if you want to process the master file by using the key as a restrictive key. The default logic performs a SETLL, which positions the records from the file by using the key and then reading without a key until the subfile fills.

### Quick Start Generation

You can generate this program type using Quick Start.



#### Exercises

See the exercises for this chapter.



# Create Subfile Maintenance Programs

---

## About Creating Subfile Maintenance Programs

You can create subfile maintenance programs that allow a user to process data and run programs using an interactive form you create.

The intended use and required entries for a typical Interactive Subfile Maintenance Program (D0040) follow:

### Program Type Description

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes a single master file by key. User defined selection exits and function keys are optional.

### Display File Definition

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

This subfile maintenance program type lets special logic permit the deletion of individual subfile records. This logic is performed by entering C in the Action Code, comparing the previous value with the current value and deleting the record if the current value is blank. The previous value is stored in a hidden field at the subfile record level by using the Display All Defined Fields in Screen Design Aid.

### CL Program Definition

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0040. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Update column. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use selection 4 to exit to the field details for the subfile field controlling the database update. Update the Entry Optional Y/N field to be N. This tells the generator that this field is a required entry before the database can be updated.

### Special Considerations

This program type uses the key information in the display file for chaining to the master file. This type must also have a hidden field and an entry optional field.

### Quick Start Generation

Generate this program type using Quick Start.



### Exercises

See the exercises for this chapter.

# Create Report Programs

---

## Creating Report Programs

When using Report Design Aid (RDA) in conjunction with the JDE World CASE tools, you can have the program generator assist you with subheadings and totals.

You should be familiar with the definition and use of report totals and subheadings when using the CASE tools. You should also be aware of some DREAM Writer considerations.

Perform the following tasks:

- ☐ Create a total format
- ☐ Define a subheading

## RDA Special Use Fields

Certain fields are used in RDA when generating reports that will contain subheadings or dynamic (hierarchical) totaling. The following illustrates how these fields are used within a report.

VC0KEY		VC0DSC	
Item Description		Quantity	HEADING1
VC0ROW	Business Unit . . .	4	Denver
RR Fields			HEADING2
	1 Bolt	300	DETAIL1
	2 Nut	400	
	3 Nail	150	
\$\$ Fields		850	
VC1ROW	Business Unit . . .	4	Denver
			TOTAL1
	Business Unit . . .	9	Boulder
	VC1KEY		VC1DSC

The following fields are used in the TOTAL1 format:

Field	Explanation
VC1ROW	Will print the data dictionary row description of the level break field. Default length is 30.
VC1KEY	Will print the value of the level break field. Default length is 12.
VC1DSC	Will print the description of the value of the break field. Default length is 30. Only works with the following fields: User defined codes Company Number Address Book Number Business Unit

The following fields are only used in the HEADING2 format, so would only be used in a C0020 or C0025 program type – Report w/Subheadings.

When subheadings are used, they are automatically underlined for you.

Field	Explanation
VC0ROW	Will print the data dictionary row description of the level break field. Default length is 30.
VC0KEY	Will print the value of the level break field. Default length is 12.
VC0DSC	Will print the description of the value of the break field. Default length is 30. Only works with the following fields: User defined codes Company Number Address Book Number Business Unit

In Case generated programs, the level breaks are softcoded. They are determined by DREAM Writer setup.

## Creating a Total Format

When you define a total format, you can define the area of the report where the description of the level break occurs. You can display up to three pieces of information for each total level break: the field description of the level break, the value of the field at the time of the level break, and the description of that value.

For example, if you choose to total your report at the business unit level, the report can read:

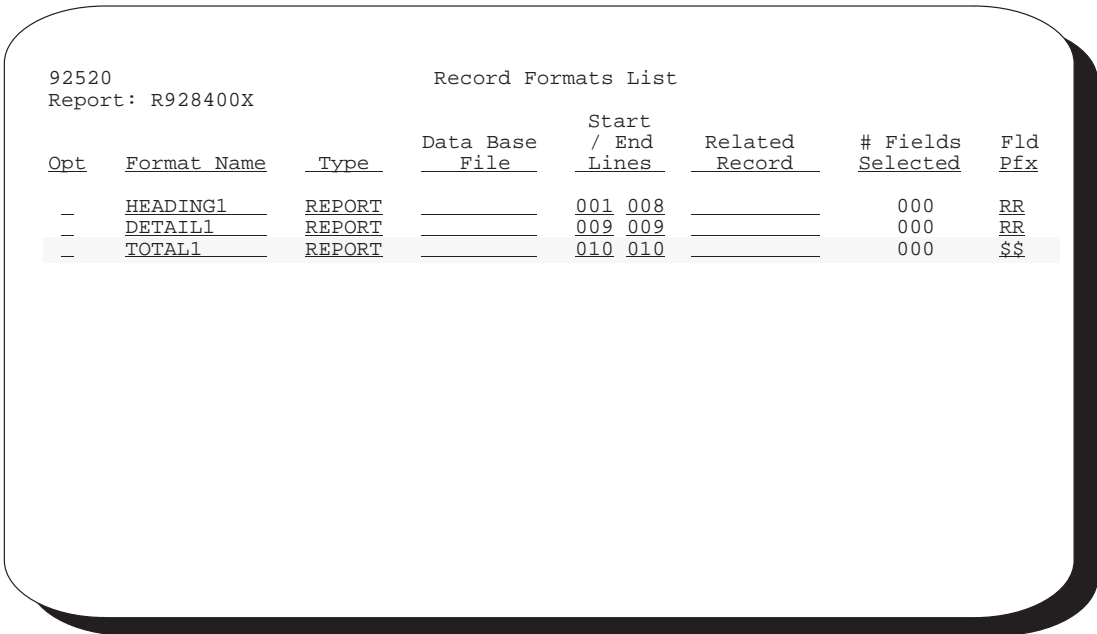
Business Unit            5            San Francisco

► **To create a total format**

---

From Software Versions Repository

1. Inquire on the report for which you want to add a total format and select option 10 for Report Design Aid.
2. From Report Design Aid press F10 to access the Record Formats List

A screenshot of a terminal window titled "Record Formats List". The window shows a table with columns: Opt, Format Name, Type, Data Base File, Start / End Lines, Related Record, # Fields Selected, and Fld Pfx. The table contains three rows: HEADING1, DETAIL1, and TOTAL1. The TOTAL1 row is highlighted. The window also shows the report number 92520 and the report name R928400X.

Opt	Format Name	Type	Data Base File	Start / End Lines	Related Record	# Fields Selected	Fld Pfx
—	HEADING1	REPORT		001 008		000	RR
—	DETAIL1	REPORT		009 009		000	RR
—	TOTAL1	REPORT		010 010		000	\$\$

3. On the Record Formats List form, add the TOTAL1 format.
4. Press enter twice to return to design area.
5. Enter an asterisk (\*) in the column and row position to begin the total description. The Field Definition window displays.
6. In the Field Name field, type VC1ROW. Press Enter twice. The window closes. The description for the total field replaces the asterisk (\*).

```

00000000000000000000000000000000
Inventory by Business Unit
00000000000000000000000000000000
00000000000000000000000000000000

```

```

Page - . . . 6666
Date - . . . 66666666

```

```

Description          Item      Description      Ship      Quantity
                    Number      Date          Date      On Hand    UM
00000000000000000000000000000000 00000000 00000000000000000000 00000000 000000000000 00
*

```

```

Report: R928400X          Field Definition          Format: TOTAL1
Dict Name      _____ Text
Data Type      A          Field Name  VC1ROW          Cond Ind
Row/Column     10  20    Field Use   Q          Highlight  -  -  -  -
Size           30          Text Form  -          Underline   -  -  -  -
Lines          Cond Ind          Field Cond   -  -  -  -
Space Before   2          Char per Inch -  -  -  -
Space After    -          Edit Code    -
Skip Before    -          Asterisk Fill -
Skip After     -          Float Symbol  -
F3=Exit F12=Prev Screen F17=Dictionary

```

In the example shown previously, this field contains the descriptive text “Business Unit” when the report prints.

7. Enter an asterisk (\*) in the column and row position to display the key value. The Field Definition window displays.
8. In the Field Name field, type VC1KEY.

```

00000000000000000000000000000000
Inventory by Business Unit
00000000000000000000000000000000
00000000000000000000000000000000

```

```

Page - . . . 6666
Date - . . . 66666666

```

```

Description          Item      Description      Ship      Quantity
                    Number      Date          Date      On Hand    UM
00000000000000000000000000000000 00000000 00000000000000000000 00000000 000000000000 00
00000000000000000000000000000000 *

```

```

Report: R928400X          Field Definition          Format: TOTAL1
Dict Name      _____ Text
Data Type      A          Field Name  VC1KEY          Cond Ind
Row/Column     11  52    Field Use   Q          Highlight  -  -  -  -
Size           12          Text Form  -          Underline   -  -  -  -
Lines          Cond Ind          Field Cond   -  -  -  -
Space Before   -          Char per Inch -  -  -  -
Space After    -          Edit Code    -
Skip Before    -          Asterisk Fill -
Skip After     -          Float Symbol  -
F3=Exit F12=Prev Screen F17=Dictionary

```

9. Press Enter. The window closes. The description for the key value replaces the asterisk (\*).

In the example shown previously, this field contains the key value “5” when the report prints.

10. Enter an asterisk (\*) in the column and row position to begin the key value description. The Field Definition window displays.
11. In the Field Name field, type VC1DSC.

00000000000000000000000000000000 Inventory by Business Unit 00000000000000000000000000000000 00000000000000000000000000000000	Page - . . . 6666 Date - . . . 66666666
----------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------

Description	Item Number	Description	Ship Date	Quantity On Hand	UM
00000000000000000000000000000000	00000000	000000000000000000000000	00000000	00000000000000	00
00000000000000000000000000000000	00000000	*			

Report: R928400X	Field Definition	Format: TOTAL1
Dict Name _____	Text _____	
Data Type <u>A</u>	Field Name <u>VC1DSC</u>	<u>Cond Ind</u>
Row/Column <u>11 66</u>	Field Use <u>Q</u>	Highlight - - - - -
Size <u>30</u>	Text Form -	Underline - - - - -
Lines <u>Cond Ind</u>		Field Cond - - - - -
Space Before - - - - -		Char per Inch - - - - -
Space After - - - - -		Edit Code -
Skip Before - - - - -		Asterisk Fill -
Skip After - - - - -		Float Symbol -
F3=Exit F12=Prev Screen F17=Dictionary		

12. Press Enter. The window closes. The description for the key value replaces the asterisk (\*).
  - In the example shown VC1DSC field displays the key value description “San Francisco” when the report prints.
13. After you add the total format, the screen displays as follows. The highlighted area contains the VC1ROW, VC1KEY, and VC1DSC fields.



```

00000000000000000000000000000000
Inventory by Business Unit
00000000000000000000000000000000
00000000000000000000000000000000

Page - . . . 6666
Date - . . . 66666666

Description      Item      Description      Ship      Quantity
Number          Date          On Hand      UM
00000000000000000000000000000000 00000000 00000000000000000000 00000000 000000000000 00
00000000000000000000000000000000 00000000 00000000000000000000000000000000

```

14. Add the field to be accumulated to the report.

- The field which contains data for Quantity on Hand column is RRXQTY. The total amount of Quantity On Hand is placed in field \$\$XQTY as part of the TOTAL1 format. The program generator creates the total amount field by adding a \$\$ prefix to the data item name.

The illustration below shows the finished report, without the cover page.

928400	J.D. Edwards & Company					Page No. . . . 2		
	Inventory by Business Unit Report					Date - . . . 12/02/93		
Bus Unit	Description	It Ty	Description	Item Number	Description	Ship Date	Quantity On Hand	UM
5	San Francisco Branch	N	Non-Refrigerated	2524	1 Inch Nail	06/01/91	100.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2532	2 Inch Nails	06/15/91	250.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2541	2 1/2 Inch Nails	05/31/91	75.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2559	3 Inch Nails	07/20/91	51.00	BX
Business Unit				5	San Francisco Branch		476.00	

## Defining a Subheading

You can define a subheading prior to the associated detail.

### ► To define a subheading

1. On the Record Formats List add the HEADING2 format on the first blank line.
  - The system handles the placement of the fields on the printed report.

92520 Report: R928400X		Record Formats List					
<u>Opt</u>	<u>Format Name</u>	<u>Type</u>	<u>Data Base</u> <u>File</u>	<u>Start</u> <u>/ End</u> <u>Lines</u>	<u>Related</u> <u>Record</u>	<u># Fields</u> <u>Selected</u>	<u>Fld</u> <u>Pfx</u>
—	HEADING1	REPORT		001 008		000	RR
—	DETAIL1	REPORT		009 009		000	RR
—	TOTAL1	REPORT		010 011		000	\$\$
—	HEADING2	REPORT		012 012		000	RR

## 2. Complete the form

Subheading field descriptions are similar to those for totals. You can display up to three pieces of information at each subhead: the field description, the value, and the description of the value of the level break fields. For example, if you choose to add a subheading to your report using business unit as the level break field, the report can read:

Business Unit            5            San Francisco

When adding the field description for the subhead, use field VC0ROW. When adding the value of the subhead, use field VC0KEY. When adding the description of the value of the subhead, use field VC0DSC.

Add these fields on the Field Definition screen in the same manner as the VC1 fields for the TOTAL1 format. On the design area, enter an asterisk (\*) where the subheading field should begin. The Field Definition window opens. Enter the field name and any other appropriate information.

The following report shows an example of how a report can look using a HEADING2 format. The highlighted area is the area defined as HEADING2. This is created using a C0020 program type.

928400		J.D. Edwards & Company Inventory by Business Unit Report				Page No. . . . 2 Date - . . . 12/02/93		
Bus Unit	Description	It Ty	Description	Item Number	Description	Ship Date	Quantity On Hand	UM
5	San Francisco Branch	N	Non-Refrigerated	2524	1 Inch Nail	06/01/91	100.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2532	2 Inch Nails	06/15/91	250.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2541	2 1/2 Inch Nails	05/31/91	75.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2559	3 Inch Nails	07/20/91	51.00	BX
Business Unit				5	San Francisco Branch		476.00	
							476.00	

Program type C0025 will print the subheadings above the column titles as follows. The Report Design Aid steps would be the same.

928400	J.D. Edwards & Company				Page No. . . . 2			
	Inventory by Business Unit Report				Date - . . . 12/02/93			
Business Unit 5 San Francisco Branch								
Bus		It		Item		Ship	Quantity	
Unit	Description	Ty	Description	Number	Description	Date	On Hand	UM
5	San Francisco Branch	N	Non-Refrigerated	2524	1 Inch Nail	06/01/91	100.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2532	2 Inch Nails	06/15/91	250.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2541	2 1/2 Inch Nails	05/31/91	75.00	BX
5	San Francisco Branch	N	Non-Refrigerated	2559	3 Inch Nails	07/20/91	51.00	BX
Business Unit				5 San Francisco Branch			476.00	
							476.00	

Grand totals are automatically added by the report program produced by the program generator to utilize the total format. When you use DREAM Writer and create a version, you define the fields to use as total levels. For further information regarding DREAM Writer, refer to the *Technical Foundation Guide*.

## DREAM Writer Considerations

When compiling your report, use PRTF to receive a cover page. PRTS does not print a cover page when the compile completes.

When the program generator creates the report program, it includes a cover page. Using the DREAM Writer, you can decide if you want to print the cover page.

The title fields the system includes on the cover page include the following:

Program ID . . P928400 Version. . . . 002		The Organization's Name Inventory by Business Unit Report San Francisco Branch Additional Line of Text if Required		Report Date. . 12/02/93 Report Time. . 9:35:50	
----------------------------------------------	--	-----------------------------------------------------------------------------------------------------------------------------	--	---------------------------------------------------	--

The VC0CO field is the name of the company. The TTL@ field is line 1 of the DREAM Writer version. TXT2 and TXT3 are lines 2 and 3 of the DREAM Writer version.

The field names for report headings are similar to those of the cover page. For report headings, the VC0CO field contains the name of the company. The DREAM Writer fields have an RR prefix. For example, RRTTL@ field contains line 1 of the DREAM Writer version ID description. The RRTXT2 and RRTXT3 are lines 2 and 3 of the DREAM Writer version description, respectively.

928400

J.D. Edwards & Company  
Inventory by Business Unit ReportPage No. . . . 2  
Date - . . . 12/02/93

On the Additional Parameters DREAM Writer setup, you must specify “2” for Type Report Totalling. This will enable you to specify your total level fields on the Data Sequence form.



### Exercises

See the exercises for this chapter.





## Additional Tools

### Objectives

- To use the Quick Start CL Generator
- To use the Quick Start Application Tool
- To use Action Diagramming

### About Additional Tools

Quick Start lets you quickly create programs, screens, and reports using:

- Quick Start CL Generator
- Quick Start Application Tool

Produce a diagram to illustrate the different groupings of logic and the interrelationships of code using:

- Action Diagramming

Perform the following tasks:

- ☐ Work with Quick Start CL Generator
- ☐ Work with the Quick Start Application Tool
- ☐ Work with Action Diagramming







# Work with Quick Start CL Generator

---

## About the Quick Start CL Generator

The Quick Start CL Generator provides a quick and easy way for you to create a Control Language (CL) program for any of the following four types of programs.

- Standard interactive program
- Standard report program with DREAM Writer
- File processor
- File processor with DREAM Writer

Quick Start CL Generator will not add your newly created CL program to a menu. You must perform that task manually.

Perform the following tasks:

- ☐ Access Quick Start CL Generator
- ☐ Compile a CL program

## Accessing the Quick Start CL Generator

### ► To access the Quick Start CL Generator

---

1. From the Computer Assisted Programming (CAP) menu G93, select Quick Start CL Generator.
2. Complete the Quick Start CL Generator Form.
3. Press F3 to return to the menu.

```
93513J                               Quick Start CL Generator

Define Application:
Description. . . . . Sample Item Master
Program Name . . . . . P55TEST
Screen or Report Name. . . . V55TEST

Select a Program Type(1-4) . . . 1
1) Interactive Program . . .      2) Output Report . . . . .
3) Batch File Processor. . .      4) Batch File Processor w/DW

Select Data From:
Master File. . . . .

Select Source and Object file:
Source File Name . . . . . JDECLSRC
Source Library Name. . . . . PGFSRC71      Object Library Name. PGFOBJ71

F24=More Keys
```

Field	Explanation
Description	Use this field to enter a short one-line description of the program you are creating.
Program Name	Type the name of the RPG program that the CL program will call. This is a required field. Do not leave it blank.  The name of the CL program generated will be the same as the RPG program name, but prefaced with a J instead of a P.
Screen or Report Name	Type the screen or report file name associated with the program. This field is only required for program type 2.

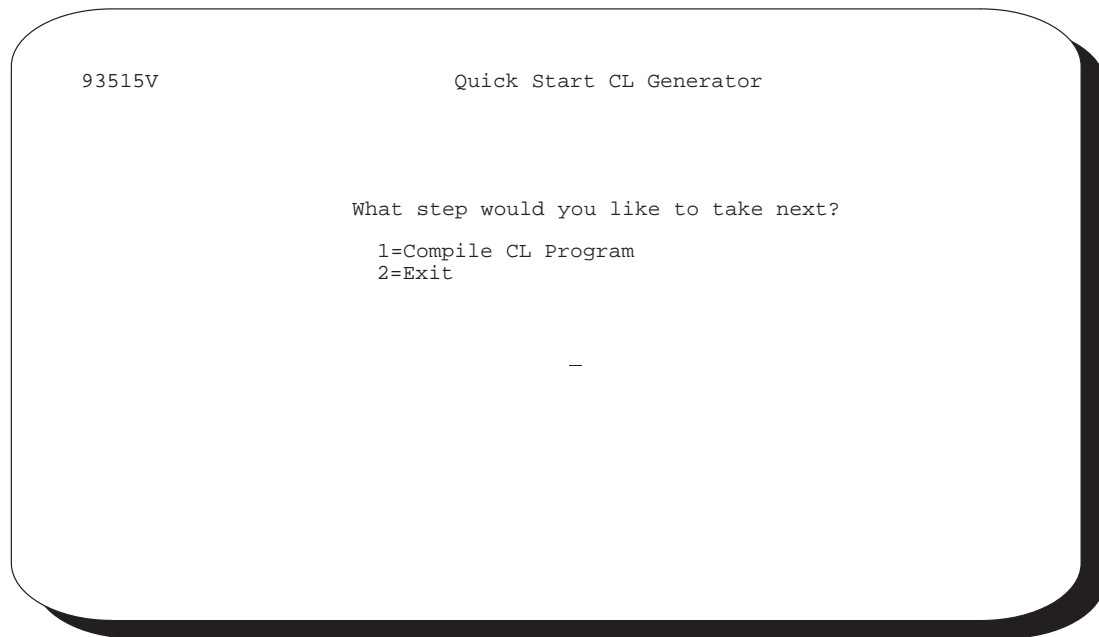
Field	Explanation
Select a Program Type(1-4)	Type one of the following in this field to indicate the type of program you are creating: 1 Standard Interactive Program 2 Standard Report Program with DREAM Writer 3 File Processor 4 File Processor with DREAM Writer
Master File	Type the name of the data file to use for the program you are creating. This field is required for program types 2, 3, and 4.
Library Name	Type in the name of the library your master file is in.
Source File Name	Enter the name of the source file for the program you are creating. This is defaulted from the CASE Profiles.
Source Library Name	Enter the name of the source library where the program you are creating resides. This is defaulted from the CASE Profiles.
Object Library Name	Enter the name of the object library where the program you are creating will reside. This is defaulted from the CASE Profiles.

## Compiling a CL Program

### ► To compile a CL program

---

From the Quick Start CL Generator form



93515V Quick Start CL Generator

What step would you like to take next?

1=Compile CL Program  
2=Exit

—

Choose Option 1 – Compile CL Program



### Exercises

See the exercises for this chapter.

# Work with the Quick Start Application Tool

---

## About the Quick Start Application Tool

The Quick Start Application Tool lets you quickly create initial versions of programs, screens, and reports. Once you have done this, you can access the Screen or Report Design Aid or the Program Generator for the member you've created and make the necessary adjustments.

The tool provides an easy way for you to create a prototype of a screen or a report and a program, if you choose. This program offers the following features:

- Lets you create a screen or report quickly. You can also create the program associated with the screen or report, if you choose.
- Lets you select fields dynamically from the master and detail files, as well as other database files.
- Lets you compile your screen or report, if you choose to.
- Creates specifications for the Program Generator and optionally creates and compiles your source code.
- Creates a Control Language (CL) program to call your new screen or report program.

## Steps of Quick Start

Quick Start has several distinct steps:

1. Quick Start Application Definition
  - Describe the application
  - Describe the type of program you want to create
  - Describe the screen options
  - Describe the report options
  - Describe the database to select from
  - Describe the source file to be used to create the application
2. Data Field Selection
  - Select the individual data fields used to create the screen or report using J.D. Edwards Screen/Report Design Aid
  - Once you select the fields you want to use, sequence them any way you choose.
3. Browse or update screens
  - View the screen or report you are creating in either Browse or Update mode.
4. Screen/Report Compilation (optional)
  - Compile the screen or report.
5. Modify Specifications
  - Based on the program type you selected, generate the File Specifications, the Detailed Programming Facility, and the Help Instructions.
  - If you compiled the screen or report in a previous step, you will also be prompted to compile the program in this step.
6. Submit to Compile
7. Update Data Dictionary and Glossary

## Selecting the Quick Start Application Definition

### ► To select the quick start application definition

From the Computer Assisted Programming (CAP) menu G93, select Quick Start Application Tool.

```

93513                               Quick Start Application Tool

Define Application:
Description. . . . . Item Maintenance
Program Name . . . . . P55TEST      Create Program(Y/N)? Y
Screen or Report Name. . . . V55TEST

Select a Program Type(1-4) . . 3
1) SFL Transaction Processor      2) SFZ Inquiry
3) Single Record Maintenance     4) Output Report

Select Screen/Report Options:
Action Code. . . . . Y           Selection Option. . . . N
Report Detail Subheadings. . N   Report Total Subheadings . . N
Report Totals. . . . . N

Select Data From:
Master File. . . . . F92801      Library Name . . . . *LIBL
Detail File(optional). . . .        Library Name . . . . *LIBL

Select Source and Object:
Source File Name . . . . . JDESRC
Source Library Name. . . . . PGFSRC   Object Library Name . PGFOBJ
F24=More Keys

```

To return to the menu, from a prompt screen with an exit, select Exit. If there is no exit option on a screen, you must continue until the process is complete, or advance to a screen that has an exit option.

Field	Explanation
Description	Use this field to enter a short one-line description of the program you are creating.
Program Name	Type the RPG program name in this field. You can leave this field blank if you choose not to create a program.
Create Program(Y/N)?	Indicate in this field whether you want to create the RPG program.
Screen or Report Name	Type the screen or report name in this field. It will default to J.D. Edwards naming standard if left blank.

Field	Explanation
Select a Program Type(1-4)	Type one of the following in this field to indicate the type of program you are creating: 1 – SFL Transaction Processor 2 – SFL Inquiry 3 – Single Record Maintenance 4 – Output Report
Action Code	Enter Y if you are creating a screen with an <i>Action Code</i> field.
Selection Option	Enter Y if you are creating a screen with a selection option.
Report Detail Subheadings	Enter Y if you are creating a report with detail subheadings.
Report Total Subheadings	Enter Y if you are creating a report with total subheadings.
Report Totals	Enter Y if you are creating a report with totals.
Master File	Type in the name of the primary file from which you select data.
Library Name	Type in the name of the library your master file is in.
Detail File (Optional)	Type in the name of an optional secondary file from which you want to select data.
Library Name	Type in the name of the library your secondary file is in.
Source File Name	Enter the file name for the screen or report and program source library (usually JDESRC). This is defaulted from the CASE Profiles.
Source Library Name	Enter the library name where the source file resides. This is defaulted from the CASE Profiles.
Object Library Name	Enter the name of the Object library. This is defaulted from the CASE Profiles.

All of the information on this screen will default in from the previous definition if you have not signed off.



## Selecting Data Fields

Key fields from each data file are pre-selected and pre-sequenced. You can deselect or resequence these if you choose using the Field Selection Window.

### ► To select specific data fields

From Quick Start Application Tool select the Field Selection Window

93515V
Quick Start Application Tool
JDED

```

93514-----Quick Start Field Selection-----
File and Library:  F92801      JDFDATA      PF
Op
      I92801      - SDM Item Master File
01  QXXIT  K01  Item ID. . . . . S      8 0      1
2  QXXDS      Description. . . . . A     30      9
3  QXXTY      Item Type . . . . . A      2     39
4  QXXDT      Date Last Ship . . . . S      6 0     41
5  QXXCC      Business Unit. . . . . A     12     47
6  QXXQT      Quantity on Hand . . . P     15 0     59
7  QXXUM      Unit of Measure. . . . A      2     67
   QXX001      Item Code 001. . . . . A      3     69
   QXX002      Item Code 002. . . . . A      3     72
-Select field sequence then F3 to continue-----
          
```

The Field Selection window, an example of which is shown above, is used to select the specific fields that you want to use in your program.

- The fields from your primary data file appear first in the list, followed by the fields from the secondary data file, if you specified one
- To see the current sequence of selected fields, press Enter
- To select a field, type *1* in the field to the left of the field name and press Enter
- For transaction processors, specify heading or subfile fields by entering 1 or 2, respectively, in the column to the right of the selection and sequencing column. This field only appears if the program you are creating is a transaction processor.
- To resequence a field, enter the sequence number in the field to the left of it and press Enter

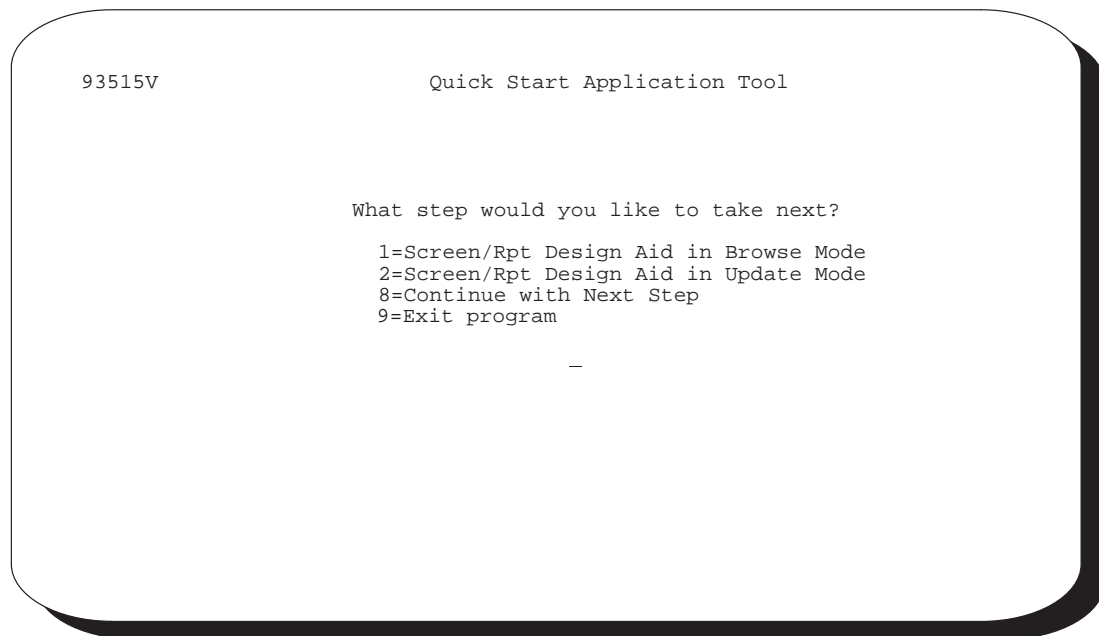
- You can enter the names of additional fields in the window to select data fields from them as well
- When you have finished with this window, press F3 to continue with the next step

## Accessing the Screen or Report You are Creating

### ► To access the screen or report you are creating

---

From Quick Start Application Tool



Choose one of the following options:

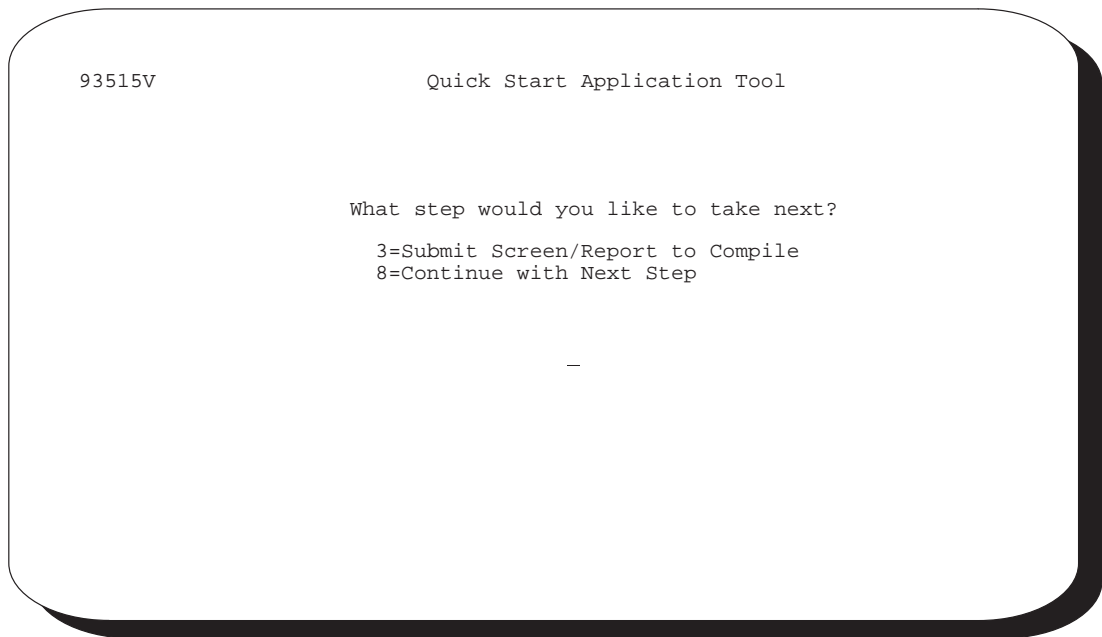
- 1 Access Screen/Report Design Aid in Browse mode.
- 2 Access Screen/Report Design Aid in Update mode.
- 8 Continue with the next step in the process.
- 9 Exit the program. This returns you to the Computer Assisted Programming (CAP) menu.

## Compiling the Screen or Report

### ► To compile the screen or report

---

From Quick Start Application Tool



The screenshot shows a terminal window titled "Quick Start Application Tool". In the top left corner, the text "93515V" is displayed. The main content of the screen is a prompt "What step would you like to take next?" followed by two numbered options: "3=Submit Screen/Report to Compile" and "8=Continue with Next Step". A horizontal line is visible below the options.

Choose one of the following options:

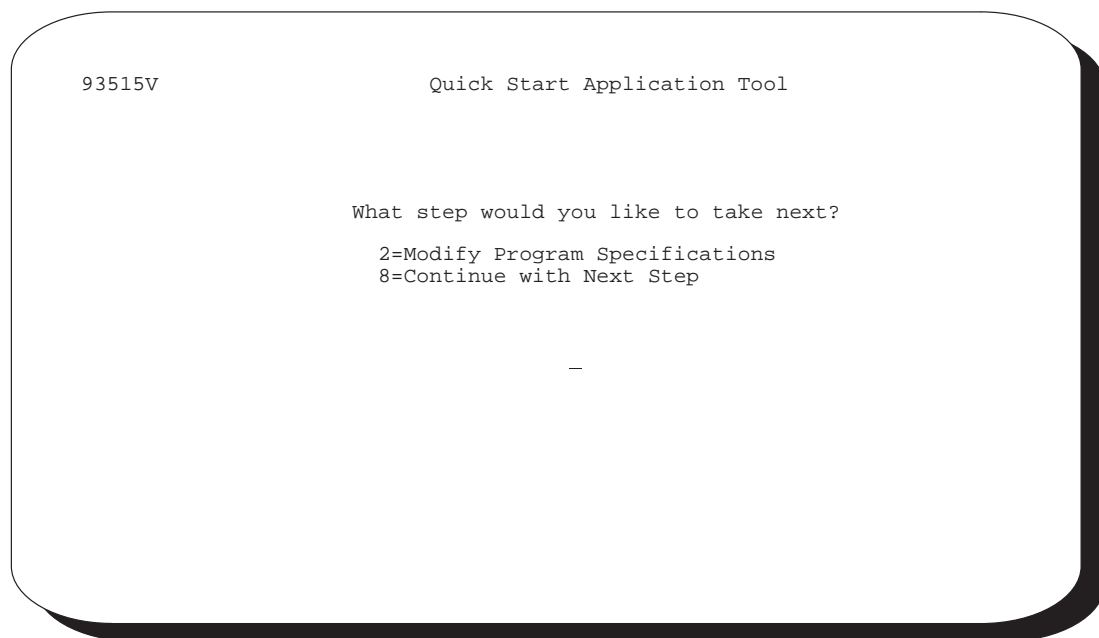
- 3     Submit the screen or report to compile.
  - The object library for the compile is retrieved from the CASE Profiles.
- 8     Continue with the next step in the process.

## Changing the Program Specifications

### ► To change the program specifications

---

From Quick Start Application Tool



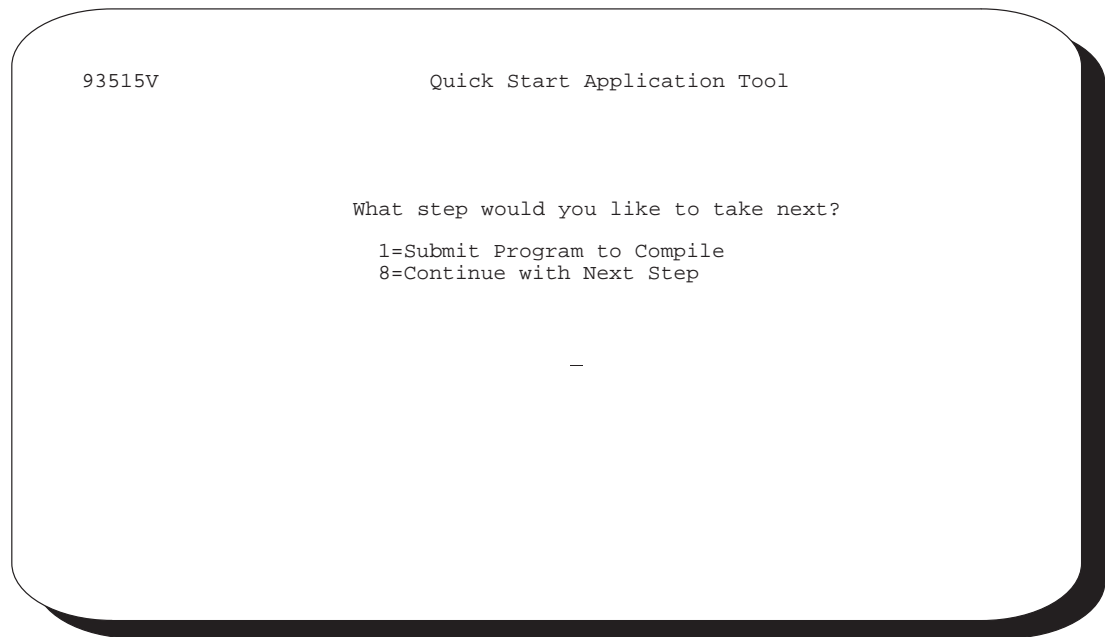
Choose one of the following options:

- 2     Modify the program specifications.
  - This option takes you to the Program Generator Specifications screen.
- 8     Continue with the next step.

## Submitting the Program to Compile

### ▶ To submit the program to compile

From Quick Start Application Tool



93515V Quick Start Application Tool

What step would you like to take next?

1=Submit Program to Compile  
8=Continue with Next Step

—

Choose one of the following options:

- 1 Submit the program to compile.
- 8 Continue with the next step.



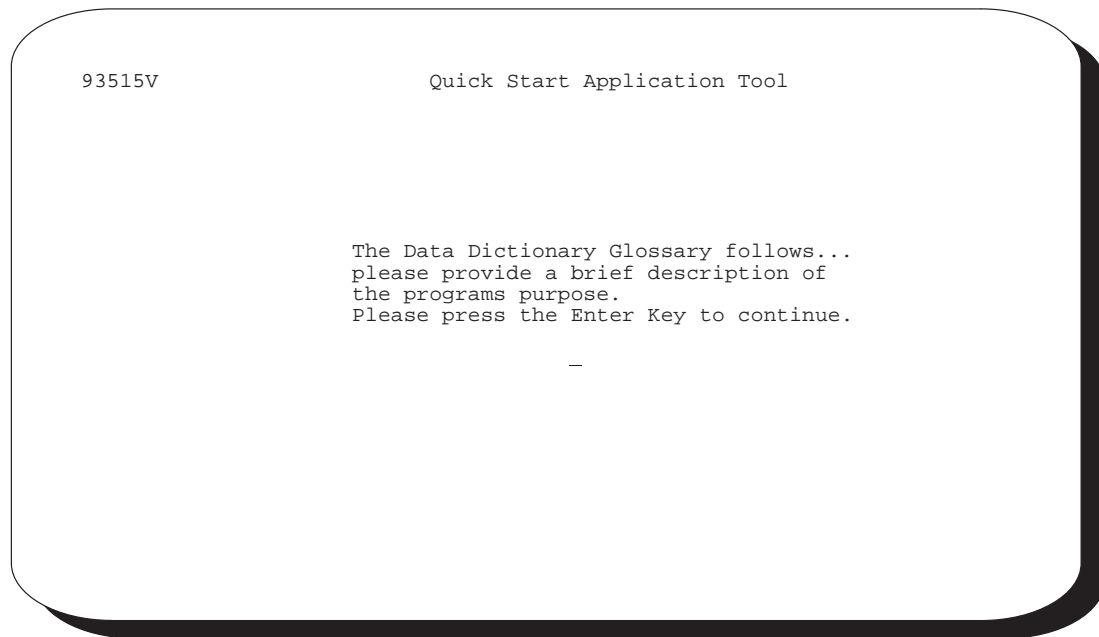
Do not submit the program to compile unless you have received a successful compile of the video or report.

## Accessing the Data Dictionary Glossary

### ► To access the data dictionary glossary

---

From Quick Start Application Tool



Press Enter to continue.

# Updating the Glossary

► **To update the glossary**

On Data Item Glossary Revisions

92001

Data Item Glossary Revisions

Language  
Applic Override  
Scrn/Rpt

Action Code. . . . .

C

Data Item. . . . .

P55TEST

Desc Sample Item Master

System Code

55

Reporting System Code

55

Glossary Group

P

This is a sample program that illustrates the Quick Start facility.

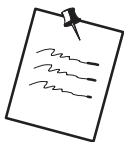
F12=Specifications

F15=Where Used

F4=Search

F5=Usr Def. Cds

Enter the description of the program’s purpose that appears in the online help instructions.



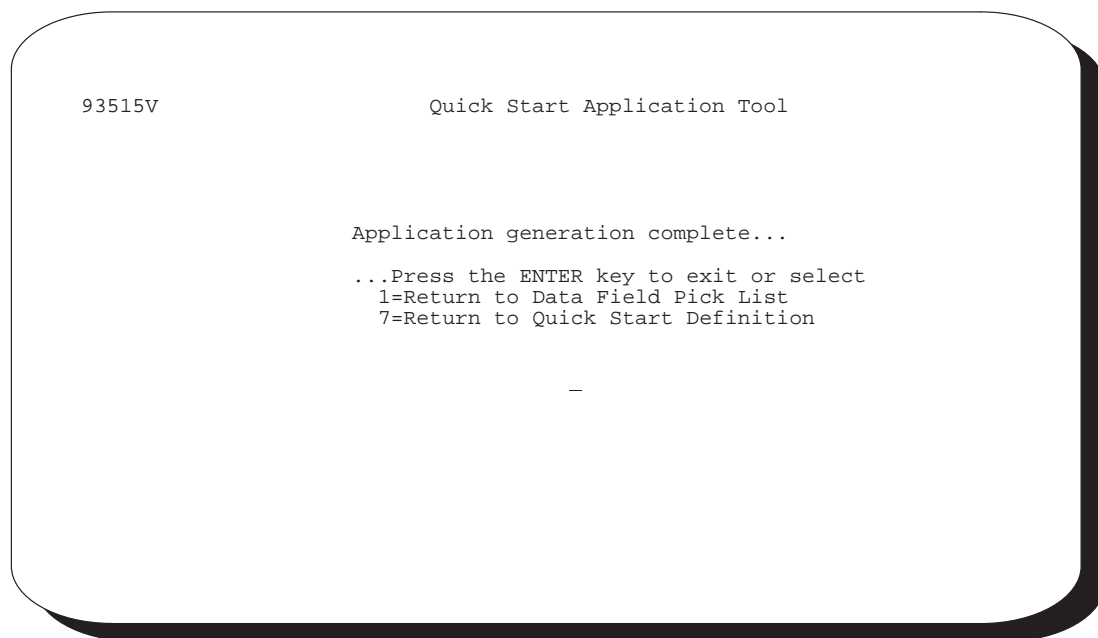
- Use a C in the *Action Code* field to add the program purpose statement.
- Press F3 to continue with the next step.

## Completing Application Generation

### ► To complete application generation

---

On Quick Start Application Tool



Choose one of the following options:

Enter Exit the program and return to the menu.

- 1 Return to the Data Field Pick List.
- 7 Return to the Quick Start Definition screen.



Notes about Quick Start

Quick Start will:

- Know if the program is a subfile.
- Add a hidden field to the video for a subfile maintenance.
- Set the *Entry Optional* field to N for a subfile maintenance.



Quick Start will not:

- Define the loading of VC0 fields.
- Add the CL program, if created, to a menu.
- Add a Fold Area.



### Exercises

See the exercises for this chapter.



# Work with Action Diagramming

---

## About Action Diagramming

The Action Diagramming facility allows you to produce a diagram which will illustrate the different groupings of logic and the interrelationships of code within a program. The diagrams are generated from the program source code. They provide easy access to more detailed information about the files, fields and programs referenced in the code.

You will perform the following tasks:

- ☐ Build an Action Diagram
- ☐ View an Action Diagram
- ☐ Understand Functions Within the Action Diagram
- ☐ View the Logic Translation Used to Create an Action Diagram

G9363  
Sr Programmers

J.D. Edwards & Company  
Action Diagramming

JDED

1. Display Action Diagram
2. Build Action Diagrams
3. Translation Table

Selection or command

==> \_\_\_\_\_



## Viewing an Action Diagram

When you view an action diagram for a program, you are seeing a graphic representation of the code's hierarchy within the program and how different subsets of code are related to other subsets of code. You can view the code for a subroutine called from the program or exit to facilities that show more detail for fields, files, and programs.

### ► To view an action diagram

From menu G9363 select the Display Action Diagram option

To view the action diagram for the program P92801 from the Action Diagramming menu G9363, select Display Action Diagram and enter P92801 in the *Program ID* field.

```

92705                                Display Action Diagram      Lvl/Sbr: 0/MAINLINE
Program ID . . . P92801            Item Maintenance           Scan: _____
====> MAINLINE PROCESSING
      Execute subroutine S999
-    <--When *INLR equals '1' Branch EOJ
-    ---> When $AUTO equals '1' Execute subr S003
      ---> End logic group
====> Do While *INLR equals '0'
      ---> If #SFRNO equals 0
            Set value of #SFRNO to 1
      ---> End logic group
      ---> If I1 less than or equal 0
            ..Else
      ---> End logic group
            Compare #####MD to '0' (High Low Equal 04)
            Write record to V928011
            Write record to V92801C
            Move '1' to @@AID
-    Execute subroutine S001
-    ---> When $998 equals ' ' Execute subr S998
      ---> End logic group
Opt:  5=View  F12=Prev  F16/F17=Scan F/B  F21=Print  F23=Flow Cht  F24=More

```

The logic groups for the program are displayed.

Group	Description
Lvl/Sbr	Specifies the logic level and subroutine that is currently displayed.
Program ID	The program name for the action diagram being displayed.
Scan	Allows the user to search for specific information.



- The use of colors, arrows, indentation, and connecting vertical lines indicates the hierarchy and relationships of the code within the program.
- The key to the symbols used is explained below:

==> Signals the beginning or ending of a loop.

---> Signals an IF or WHEN statement or their associated end statement.

Labels are presented in reverse image.

## What Are the Function Key Exits?



### F10 – Display File Usage

To view the files used in the file specifications of the program.

```

92705                               Display Action Diagram      Lvl/Sbr: 0/MAINLINE
Program ID . . . P92801          Item Maintenance             Scan: _____
==> MAINLINE PROCESSING
    Execute subroutine S999
-   <--When *INLR equals '1' Branch EOJ
-   ---> When $AUTO equals '1' Execute subr S003
-   ---> End logic group
==> Do While *INLR equals '0'
    ---> If #SFRNO equals 0
        Set value of #SFRN
    ---> End logic group
    ---> If I1 less than or
        ...Else
    ---> End logic group
        Compare ####MD to '0
        Write record to V928
        Write record to V928
        Move '1' to @@AID
        Execute subroutine S
-   ---> When $998 equals
-   ---> End logic group
Opt: 5=View F12=Prev F16/F17=Scan F/B F21=Print F23=Flow Cht F24=More

```

Q	File	U	Description
-	F0001	N	Business Unit Security
-	F92801	N	SDM Item Master File
-	F92801LA	Y	LF - Cost Center, Item ID
-	V92801	N	Item Maintenance

Opt: 8=FPD F9=Glossary F24=More Keys



### F12 – Return to Previous Logic Level

Allows the user to return to the logic level that is immediately prior to the one currently displayed.



### F16 – Scan Forward

Allows user to enter a value they want to search for in the *Scan* field and then scan forward through the code to find it.



### **F17 – Scan Backward**

Allows user to enter a value they want to search for in the *Scan* field and then scan backward through the code to find it.



### **F19 – Skip to Start Group**

Allows user to skip to the beginning (start) of a section of code.

User places cursor within the section of code and then presses F19 to go to the beginning of that section of code.



### **F20 – Skip to End Group**

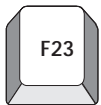
Allows user to skip to the end of a section of code.

User places cursor within the section of code and then presses F20 to go to the end of that section of code.



### **F21 – Print**

Allows the user to obtain a printout of the action diagram.



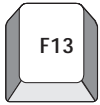
### **F23 – Flowchart**

Allows user to view and/or print a flowchart which illustrates the interaction of files and processes related to a single program.

Can continue to view lower levels of detail as well.

## What Are the Cursor Sensitive Function Key Exits?

To determine related information for fields, files, and programs appearing in the program code, you can use cursor sensitive function keys to access related information by placing the cursor at the beginning of the field, file, or program desired.



### F13 – Software Versions Repository

Exits to the Software Versions Repository.



### F14 – File Field Description

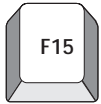
Displays the File Field Description window.

```

92705          Display Action Diagram          Lvl/Sbr: 1/S005
Program ID . . . P92801      Item Maintenance      Scan: S005
      Move N@XIT to PSIDX
      Execute program 'X0010'
      Move #NXTNO to QXXIT
      Move #NXTNO to SFXIT
      Set file pointer F92801 low limit QXXIT
      ==> End Repeating Group
98FFD          File Field Descriptions          S/FMT
File and Libr:  F92801      PGFDTA61      PF
- I92801      - SDM Item Master File
- QXXIT      K01 Item ID. . . . . S          8 0          1
- QXXDS      Description . . . . . A         30          9
- QXXTY      Item Type. . . . . A           2         39
- QXXDT      Date Last Ship . . . S          6 0         41
- QXXCC      Business Unit. . . . A          12         47
- QXXQT      Quantity On Hand . . P         15 0         59
- QXXUM      Unit of Measure. . . A           2         67
- QXX001      Item Code 001. . . . A           3         69
- QXX002      Item Code 002. . . . A           3         72
Opt:  2=Dictionary  4=Sel  F15=Resequenece  F3=Return
Opt:  5=View  F12=Prev  F16/F17=Scan F/B  F21=Print  F23=Flow Cht  F24=More

```





## F15 – Data Cross Reference

Exits to the cross reference program.



## F18 – Data Dictionary

Exits to the data dictionary program.

The chart below indicates which function keys provide relevant information for the different elements.

Object Type	Function Key	Description
Fields	F15	Displays all the programs that use the data item.
	F18	Displays the Data Dictionary definition for the selected data item.
Files	F10	Displays the files used within the program.
	F13	Displays the Software Versions Repository record for the selected file.
	F14	Displays the File Field Descriptions for the selected file.
	F15	Displays all the programs that use the file.
Programs	F13	Displays the Software Versions Repository record for the selected program.
	F15	Displays all the programs that call the selected program.

## What Are the Selection Exits?

### Selection 5 — View

- Allows user to view subroutine code whenever it is indicated that the program is to execute a subroutine.

## Accessing Logic Translation Feature

The Logic Translation feature allows you to view how the Action Diagrammer translates the RPG code of a program into its Action Diagram.

### ► To access the logic translation feature

---

From menu A9363, select the Action Diagram Translation option.

```
92710                                Translation Table

Action Code. . I

      Internal
Operation  Translate to Operation
-----
ADD       Add &1 to &2 giving &3
ADDA      Add &2 to &3
ANDEO     And &1 equals &2
ANDGE     And &1 greater or equal &2
ANDGT     And &1 greater than &2
ANDLE     And &1 less than or equal &2
ANDLT     And &1 less than &2
ANDNE     And &1 not equal &2
BEGSR     Begin Subroutine &1
CABEQ     When &1 equals &2 Branch &3
CABGE     When &1 greater or equal &2 Branch &3
CABGT     When &1 greater than &2 Branch &3
CABLE     When &1 less than or equal &2 Branch &3
CBLT     When &1 less than &2 Branch &3
CABNE     When &1 not equal &2 Branch &3
CALL      Execute program &2
```

The system displays the RPG operation in the first column and then displays how that operation is translated within an action diagram.



# Source Inventory and Database

## Objectives

- To understand the Source Sequence Line Number
- To create or modify program types
- To create or modify logic modules
- To understand directives
- To understand the Question and Answer system
- To create user defined PDL

## About Using the Source Code Inventory and Database

Perform the following tasks:

- ☐ Understand the Source Sequence Line Number
- ☐ Create or modify program types
- ☐ Create or modify logic modules
- ☐ Understand directives
- ☐ Work with the Question and Answer system
- ☐ Create user defined PDL



Access the Model Program Design Menu (G9361).

```
G9361                      J.D. Edwards & Company          JDED
Sr Programmers             Model Program Design

... PROGRAM TYPES:
 2. Create/Modify
 3. Index
 4. Cross Reference
 5. Maintain Q/A
 6. Program Search (w/logic type)
... LOGIC MODULES:
 8. Create/Modify
 9. Index
10. Cross Reference
11. Op Codes
12. Formula Library Entry

... OTHER TOOLS:
14. Parameter Copy/Move
15. Print Program Specification
16. Review Source Modifications
17. Generator Updates
18. CASE Specifications Inquiry
... GENERATION OPTIONS:
20. Help Instructions Edit/Build
21. All Help Instructions
22. Global Program Regeneration
```

Selection or command

```
====> _____
        _____
        _____
```

# **Understand the Source Sequence Line Numbers**

---

## **Understanding Source Sequence Line Numbers**

You must understand:

- Source Serial Numbers
- Source Sequence Line Structure
- Structure of the Serial Number

### **Source Serial Numbers**







When the program generator creates a new program, it assigns each line of source code within the program a twelve-digit serial number. If you regenerate a program after making changes, the program generator uses the serial numbers to integrate your changes, then renumbers the entire source.

## Source Sequence Line Structure

The source sequence line structure includes six elements:

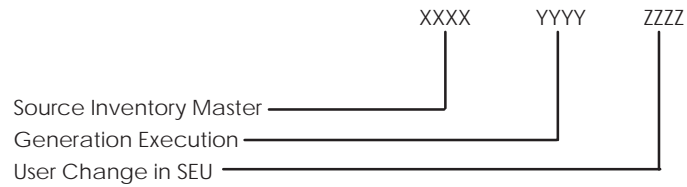
Primary Key	The primary key represents source code lines that come from a Primary Logic Module. The primary key begins in column 80.
Secondary Key	The secondary key represents the source code lines that come from a Detail Logic module. The secondary key begins in column 90.
Serial Number	The serial number is a 12-digit number the program generator assigns to each line of source code in a program. The serial number begins in column 100.
User ID	When the program generator creates a program, it places the User ID of the program's creator within the source sequence line.
SAR Number	When the program generator creates a program, it places the SAR Number, if available, within the source sequence line.
Date Last Change	When the program generator creates a program, it places the date the code was added or changed within the source sequence line.

The following illustration displays the parts of the source sequence line.

Primary Key	Secondary Key	Serial Number	User ID	SAR Number	Date Last Change
					
S999-4	RR#BEN	007000700000	QUARLES	721561	000000
S999-4	RR#BEN	007000800000	QUARLES	721561	000000
S999-4	RR#BEN	007000900000	QUARLES	721561	000000
S999-4	RR#BEN	007001000000	QUARLES	721561	000000
S999-4	RR#BEN	007001100000	QUARLES	721561	000000
S999-4	RR#BEN	007001200000	QUARLES	721561	000000
S999-4	RR#BEN	007001300000	QUARLES	721561	000000

The Primary and Secondary keys and serial number make up a unique key for each line of code.

## Structure of the Serial Number



### XXXX

- Assigned by the Source Inventory Master file (F93001).
- Incremented by 10 to allow lines to be inserted as the Program Generator Source Inventory Master file has changes made to it.
- Maximum of 9999 lines.

### YYYY

- Assigned at generation time.
- Represent lines that are part of a detail logic module.
- Incremented by 10 to allow for line insertion.
- Maximum of 9999 lines.

### ZZZZ

- Represent lines of code that the user has inserted via SEU.
- Maximum of 9999 lines.





# Create or Modify Program Types

---

## About Creating or Modifying Program Types

Within the program generator, the program type specifies the basic function or type of program that you create. For example, there are separate program types for basic interactive maintenance programs:, programs which use subfiles, conversion programs, report programs, and batch update programs.

The system ties each program type to the question and answer process with the program generator.

- After answering a series of questions about the program to generate, the system determines the program type and assigns it to your program specifications.
- The program generator constructs the program using primary and detail logic modules defined within the program type.

JDE provides you with program types for the most common programs. You can create your own program types for your organization's needs.

## Creating or Modifying Program Types

### **To create or modify program types**

---

- Keep the alpha order requirement in mind when creating new program types.
1. Select the create/modify option under program types on the Model Program Design menu
  2. Inquire on an existing program type

```
93001                                Create/Modify Program Types
Action Code . . . . I
Program Type . . . D0040          SFL/T/F - w/Act - w/Sel - Keys
Seq Prim Modul Glossary K
1.00 FILEDEFN01 _____ File Specification
2.00 FILEEXTN1  _____ Tables & Arrays      - SFL Video
3.00 INPUT1     _____ Data Structures    - STD Video
4.00 MAINLINE   _____ Mainline           - Video
5.00 S00EX-5    _____ Exits Subroutine   - SFL Trans
6.00 S00QP      _____ Options Subroutine
6.40 S00VL-1    _____ Return Values Subr - Standard
7.00 S001-3     _____ Clear Subroutine   - SFL Trans
8.00 S003-4     _____ Edit Key            - SFL T/Fld
9.00 S004-5     _____ Load Subfile Subr  - SFL Trans
10.00 S005-2    _____ Edit SFL Upd Subr  - SFL Trans
11.00 S010-2    _____ Update Subroutine  - SFL Trans
12.00 S999-5    _____ Housekeeping Subr  - SFL Trans
_____
_____
_____                               F24=More
```

All of the pieces required to create program type D0040.

## Program Type

- The Bill of Materials List.
- It is the list of what segments of code are required to build this type of program.

## Primary Module

- The main sections of code that will be used to create the first level of program source.

## Glossary K

- Used to document logic modules within a program type.
- When a program is generated, the field is validated against the Data Dictionary, and the glossary for the key is added as documentation for the logic module.

## Abbreviations for the Program Types Index

The following abbreviations are heavily used:

ACT	- Action Code	w/	- with
B	- Batch (pure-w/o reports or videos)	w/o	- without
CLP	- Control Language Program	T	- SFL Trans Processor
F	- SFL Fold Area Processing	TOT	- Dynamic Totaling
I	- Inquiry Only		
Keys	- Unique SFL Record Key Processing		
M	- Simple Maintenance Program		
MST	- Master Files		
R	- Report Writer		
RRN	- Relative Record Number		
READC	- Read modified SFL records only		
SEL	- Selection Exit Processing		
SFL	- Sub-File Processing		
STD	- Standard		

( F6 - Execute )

The above screen explains the abbreviations used on the next page.

## Program Types Index

93900

## Program Types Index

<u>Q</u>	<u>Prog Type</u>	<u>Description</u>
-	A0010	SFL/I
-	A0020	STD/I - Single record - No action
-	B0010	STD/M - Action Code
-	C0010	STD/R
-	C0020	STD/R - Subhead
-	C0025	STD/R - Subhead above Column Headings
-	D0010	SFL/T/F - w/Act - wo/Sel - RRN
-	D0020	SFL/T - wo/Act - wo/Sel - RRN
-	D0030	SFL/T/F - wo/Act - wo/Sel - RRN - Readc
-	D0040	SFL/T/F - w/Act - w/Sel - Keys
-	D0045	SFL/T/F - wo/Act - - Keys
-	D0050	SFL/T/F - w/Act - w/Sel - RRN - 2 Mst
-	D0060	SFL/T/F - w/Act - wo/Sel - Keys
-	D0070	SFL/T/F - w/Act - w/Sel - RRN
-	D0080	SFL/T/F - wo/Act - w/Sel - RRN
-	D0090	SFL/T/F - w/Act - wo/Sel - RRN - Bal
-	D0100	SFL/T/F - w/Act - w/Sel - Keys - 2 Mst
-	E0010	STD/W OBSOLETE
-	E0020	STD/W Using SL01-SL10 OBSOLETE
Opt: 1=Defn 2=Prt Src 3=Dsp Src 4=Select 5=X-Ref 6=Chg Pgm Type		

### Selections

1 — Glossary from the Data Dictionary

2 — Print Source

- Prints the generic source of what the shell program will look like without any of the specifics (detail logic modules) inserted.
- Helpful if creating your own program types and you want to see how it looks.

3 — Display Source

- Displays the generic source of what the shell program will look like without any of the specifics (detail logic modules) inserted.
- Helpful if creating your own program types and you want to see how it looks.

4 — Select

- Returns the program type when called from another program.

5 — Cross Reference to Programs

- Shows which programs were created using this logic type.

6 — Chg LC

- Displays the Bill of Materials list for the program type.

## Program Types Cross Reference

```

93953                                Program Types X-Reference

Program Type . . . . . A0010_____ SFL/I
Program      Program Title
-----
J98COMPILE  Compile a single object
PHELPCAT
PTOM
PYU
P00HELP     P00HELP    - Help Inquiry
P00HLP2     Help View
P00NS       Business Unit Name Search Window
P00005      System Level Protection Codes
P0001Z      Batch File Review - AB,AR,AP,GL
P0006S1     Business Unit Search
P0006S2     Business Unit Search by Level of Detail
P0011W      Address Inquiry
P00121      Automatic Accounting Instructions Displ
P00152EC    Currency Exchange Rate calculation list
P00192T1    - General Message Information
P00192T2    - Message Information
P00201      Journal Review (All Systems)
                                F14=KBG Status All/Only Active Toggle

```

Same program as Selection 5 from the Program Types Index screen (Cross Reference to Programs).



### F14 – Clone Status All/Only Active Toggle

- Allows the user to toggle back and forth between seeing all programs using the program type or only the programs with a CAP status of 'Y'.



# Create or Modify Logic Modules

---

## About Logic Modules

There are two types of Logic Modules:

- ☐ Primary Logic Modules
- ☐ Detail Logic Modules

### What Are Primary Logic Modules?

- Main segments of code used in the definition of a program type.
- Normally they are full sections of a program or subroutines within the program.
- Contain Functional directives to the generation program.

Primary logic modules are full sections of a program or subroutines within the program and contain functional directives to the generation program. Each primary logic module is coded with a five character directive code (see Columns 1 through 5 in the Master Source Code File - F93001).

The World CASE software provides approximately 100 different primary logic modules. This includes many variations on mainline logic, field initialization, update logic, housekeeping, and so on. Use the Logic Module Index to become familiar with the various types of primary logic modules.

The primary logic modules are the main segments of code used in the definition of a program type. For example, primary logic modules contain:

- Program identification specifications
- Extension specifications
- Data structures
- Mainline calculations
- Default logic from Data Dictionary
- Subroutine calculations
- Update subroutine
- Housekeeping subroutine, etc.

### What Are Detail Logic Modules?

Detail logic modules are used to direct the final integration of the database, video, and/or report specifications into the primary logic modules that make up the finished program type.

Detail logic modules are usually functional or data field-related segments of code. Detail logic modules are referenced by functional directives and contain substitution directives to the generation program. A prefix of X indicates the detail logic module is not used in conjunction with a conditional directive. A prefix of Z indicates the detail logic module is used in conjunction with a conditional directive. For further information about directives, refer to the chapter *Directives* in this guide.

- Normally functional or data field related segments of code.
- Referenced by Functional directives.
- Contain Substitution directives to the generation program.
- Begin with either an 'X' or a 'Z'
  - 'X' means it is NOT used in conjunction with a conditional directive.
  - 'Z' means it IS USED in conjunction with a conditional directive.



## Creating or Modifying Logic Modules

### ► To create or modify logic modules

---

1. From the Model Program Design menu select the Create/Modify option under LOGIC MODULES and enter a logic module name.
  - You can use F1 to search for logic modules.

```
93001SEU          Create/Modify Logic Modules
Primary Logic Module Key . . . S002-1_____
Logic Module Description . . . _____
Duplicate from Logic Module. . _____
```

2. Create or change the appropriate lines of code

```
Columns . . . : 1 71          Browse          QTEMP/F93001WRK
SEU==>          F93001
***** Beginning of data *****
0001.00      C*
0002.00      C*      SUBROUTINE S002 - Monitor Report Level Breaks
0003.00      C*      -----
0004.00      C*
0005.00      C*      Processing:  1.  Check for change in field values.
0006.00      C*                      2.  Set total printing level.
0007.00      C*
0008.00      CSR          S002          BEGSR
0009.00      C*          ----          ----
0010.00      C*
0011.00      C*      If no level breaks requested, bypass subroutine.
0012.00      C*
0013.00      CSR          MOVEA@@L          #@LCA  36
0014.00      CSR          #@LCA          IFEQ *BLANK
0015.00      CSR          MOVE '1'          $LVLB  1
0016.00      CSR          GOTO END002
0017.00      C*          ----          ----
0018.00      CSR          END
0019.00      C*-----
```

- Three steps are immediately performed when you take this option.
  - Work file is created in QTEMP/F93001WRK.
  - Member is added to F93001WRK.
  - Member is cleared in F93001WRK.
- Allows the user to exit without saving changes.
- Allows for seeing only the logic module the user wants, otherwise all 12,000 lines of code would be brought in because F93001 is a single member file.

## Accessing the Logic Module Index

### To access the Logic Module Index

---

From the Model Program Design menu select the Index option under LOGIC MODULES

81QM	User	Defined Codes	Window	PKEY
93	LM	Primary Logic Modules		
Skip To Code . . . . .				
-	FILEDEFN01	File Specification		
-	FILEEXTN0	Tables & Arrays	-	STD Video
-	FILEEXTN1	Tables & Arrays	-	SFL Video
-	FILEEXTN2	Tables & Arrays	-	STD Rpt
-	FILEEXTN3	Tables & Arrays	-	2F - Conv
-	FILEEXTN4	Tables & Arrays	-	Batch
-	FILEEXTN5	Tables & Arrays	-	Windows
-	INPUT1	Data Structures	-	STD Video
-	INPUT2	Data Structures	-	STD Rpt
-	INPUT3	Data Structures	-	2F - Conv
Opt: 4=Select F9=Glossary F14=Memo				

ny JDED  
er

ER KBG TOOLS:  
rameter Copy/Move  
int KBG Specification  
view Source Modifications  
G Updates

ERATION OPTIONS:  
lp Instructions Edit/Build  
l Help Instructions  
obal Program Regeneration

Selection or command  
==> 9

- There may be multiple logic modules for each subroutine.
  - The same subroutine looks different based on the type of program it is used in.

## Using Logic Module Cross Reference

The Logic Module Cross Reference allows the you to determine which program types use a particular logic module.



### To use the Logic Module Cross Reference

1. From the Model Program Design menu select the Cross Reference option under LOGIC MODULES

```
93952                                Logic Module  X-Reference
Primary Logic Module . S002-1      Level Breaks      - STD Rpt
Program Type      Description
C0010            STD/R
C0020            STD/R      - Subhead
C0025            STD/R      - Subhead above Column Headings
X0010            STD/B      - Updt
```

F24=More Keys

2. Enter a primary logic module name

## Using Logic Module Op Codes

Logic Module Op Codes allow you to

### ► To use the logic module Op codes

From the Model Program Design menu select the Op Codes option under LOGIC MODULES

```

93108                                Logic Module  Op Codes

Action Code. . . . . I
Op Code  X module      Description_____
|<_____ XTCAT        Concatenate with Truncation
||_____ XCONCAT       Concat calcs
|>_____ XBCAT        Concatination w/Blank
ADD_____ XADDITION1   Addition calcs
CALL_____ XCALL       Call Statement
CDESC1_____ XCDESC1   Beginning Formula comment line
CDESC2_____ XCDESC2   Ending Formula comment line
CHAIN_____ XCHAIN     Chain Calculation
CINIT_____ XCINIT     Concat initialization calcs
COMNT_____ XCOMNT     Comment calcs
CONCAT_____ XCONCAT    Concat calcs
DELETE_____ XDELET     Delet Operation
DIV_____ XDIV         Division calcs
DOUEQ_____ XDOUEQ     DOUEQ Calculation Logic
DOUGE_____ XDOUGE     DOUGE Calculation Logic
DOUGT_____ XDOUGT     DOUGT Calculation Logic
                        F24=More Keys

```

- Left column lists the PDL op codes.
- Right column shows the x-module that will be called to generate the source code.
- If PDL does not generate source code, this file (F93108) may have been accidentally cleared.

## Maintaining the Logic Module File

The following programs do not appear on a menu and must be called manually. They should be used with extreme caution.

### Resequenece Logic Module

- P93998
  - Submits a program to resequence an existing logic module.
  - **THIS IS VERY DANGEROUS!!!!**
  - Used when several lines need to be added to a logic module and the line numbers need to be resequenced.
  - Normally, a new logic module will be created and incorporated into a new program type and people are told to use the new program type and eventually the old program type will be deleted when there are no more programs with that program type that have a CAP status of 'Y'.
  - CALL P93998 PARM(logic module name).
  - If the user adds or changes lines in a logic module, they **MUST** manually change/add the serial numbers for the logic module or run this.

### Remove Logic Module

- P93999
  - Takes lines out of F93001.
  - Submits a program to remove an existing logic module.
  - **THIS IS VERY DANGEROUS!!!!**
  - Used when a logic module is no longer used in order to reduce the amount of source in the F93001 file.
  - Must make sure that there are not any programs with a CAP status of 'Y' that are using a program type that looks for this logic module.
  - CALL P93999 PARM(logic module name).

## Creating or Modifying Formula Library Entry

### ► To create or modify the formula library entry

From the Model Program Design under LOGIC MODULES select Formula Library Entry

93109 Formula Library Entry

Action Code. . . . . -  
Program ID . . . . . \*FORMULA  
File ID. . . . . \*LIBRARY  
Field Name . . . . .

Data Item Formula

Data Item	Formula

F5=Variables F24=More

- This is the same form that is accessed through the Detailed Programming Facility to enter PDL.
- From this menu, the screen is pre-loaded with the keys for entering a formula.

## Creating or Modifying Parameter Copy/Move

Parameter Copy/Move allows you to copy from one library to another and/or one program ID to another:

- Program Generator specifications
- Data Dictionary glossary (program purpose)
- DREAM Writer processing options

### ► To create or modify parameter copy/move

---

1. From the Model Program Design under OTHER TOOLS select Parameter Copy/Move

```
93890                                Parameter Copy/Move

      Description                      From Lib   To Library (Blank = From Lib)
-----
Program Generator Specs. . . . .      _____
Data Dictionary (Purpose). . . . .    _____
Processing Opt (If Required) . . . . .  _____
From Program ID. . . . .              _____
To Program ID. . . . .                _____ (Blank = From ID)
```

2. Complete the Parameter Copy/Move form
  - You can use Software Versions Repository, selection exit 3, to copy Program Generator specifications within a library





## Reviewing Source Modifications

The Review Source Modifications option shows the source code that the user added manually through SEU.

- Same as using Selection Exit 30 from the Software Versions Repository.

### ► To review source modifications

---

1. From the Model Program Design menu under OTHER TOOLS select Review Source Modifications
2. On the Software Versions Repository form inquire on desired program
3. Select option 30 to view source code modifications

```
Columns . . . :   1  71           Edit           JDFCLONE/F93002
SEU==>                                     P928401
FMT **   ...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
***** Beginning of data *****
0001.00 21  C          QXXDT      IFLT $#BDAT
0002.00 21  C          QXXDT      ORGT $#EDAT
0003.00 21  C          GOTO END
0004.00 21  C          END
0005.00 21  CSR        MOVE@OP,1    $#BDAT  60
0006.00 21  CSR        MOVE@OP,2    $#EDAT  60
0007.00 21  C*-----
***** End of data *****

F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
F16=Repeat find   F17=Repeat change   F24=More keys
```

- These lines are the result of the MPxxxxx job that runs and compares the 'before image' of the source with the source after the user makes changes and stores the changed lines in the Pxxxxx member in the F93002 file.
- The user is viewing the Pxxxxx member in the F93002 file.

## Using Program Generator Updates

Program Generator Updates merge in J.D. Edwards updates for the Program Generator.

- These jobs are used during a PTF install.



### **To use program generator updates**

---

From the Model Program Design menu under OTHER TOOLS select Generator Updates

G9366  
Sr Programmers

J.D. Edwards & Company  
Generator Updates

JDED

1. Program Type Compare/Update
2. Logic Module Compare/Update

Selection or command  
===> \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

## Using CASE Specifications Inquiry

The CASE Specifications Inquiry allows you to view programs designed by using the J.D. Edwards CASE Tools. You may modify and delete CASE Specifications through this utility.

### ► To use CASE specifications inquiry

1. From the Model Program Design under OTHER TOOLS select CASE Specifications Inquiry.

```

93130                                CASE Specifications Inquiry

Program ID . . . P92801             to _____
System Code. . . _____         CAP Status . . . _   Program Type . . _____

O  Program                               Syst   Program   S
P  ID                               Code   Type   t   SVR Status
- P92801      Item Maintenance          92   D0040   Y
- P928011     Item Master Information    92   B0010   Y
- P92802      Item Maintenance          92   A0010   Y
- P928200     Item Search                92   A0010   Y
- P928400     Inventory by Business Unit 92   C0010   Y
- P928401     Inventory by Cost Center w/Sub 92   C0020   Y
- P92910      Copy ADW Files into Production 93   X0010   N
- P93KBG      Check if member is a KBG Progr 98   X0010   N
- P93KL       File Server Key Lists      93   E0010   N
- P93001      Create/Modify Program Types 93   D0040   N
- P930011     Logic Module Compare/Update 93   C0020   N
- P930012     Program Type Compare/Update 93   C0020   N
- P930013     Update/Merge Application Gener 93   X0030   N
- P930014     Print Logic Module          93   C0020   N

Opt: 1=SVR  2= CASE Specs  F4=More Data  F24=More Keys

```

2. Specify search criteria, type the Program ID, System Code, CAP Status, or Program Type and press Enter. Selected records display interactively.
  - Option 1 allows you to work with the source code in the Software Versions Repository.
  - Option 2 allows you to modify and delete the CASE Specifications defined for a specific program.

## Generation Options

### Help Instructions Edit/Build

- Exits to the Software Versions Repository so the user can rebuild the Helps for a single program.

### All Help Instructions

- Submits a job to regenerate the helps for all programs.

### Global Program Regeneration

- Regenerates all programs that have a CAP Status of 'Y'.
- **THIS IS VERY DANGEROUS!!!**



# Understand Directives

---

## Understanding Directives

Directives are contained in the logic modules. They instruct the program generator on the type of action to take when constructing source code. They use the first five columns of the RPG statement.

There are several types of directives, including:

- ☐ Functional Directives
- ☐ Substitution Directives
- ☐ Exception Directives
- ☐ Conditional Directives

Users cannot create their own directives. J.D. Edwards supplies all directives.

## Functional Directives

- Control major functions within a program.
- Provide the initiation point for creation of data base specific logic and video/report file control logic.
- Initiate the inclusion of copy modules into the source code.
- Only found within the realm of PRIMARY logic modules.
- **CANNOT** be in a detail logic module.
- Grab detail logic modules for inclusion.

## Functional Directives

Directive Code	Detail Logic Module	Source Created	Functional Directive
*ACTN	None	S999	Load action code lock out array
*ATOT	XADDTOT1	S010	Accumulate report total logic
AUTHR	None	F spec	Program author
*AUTOI	X*ENTRYI	S999	Automatic inquiry at execution test logic
*CLRN	None	S001	Clear user requested fields
CLRY	None	S001	Clear all data fields for next transaction
CLSFL	None	S003	Clear all subfile fields
COPY	XCOPY-SUB	Various	RPGIII copy function for common subroutines
CTOT	XCLRTOT1	S010	Clear report totals
*DATES	XDSDATE	I Spec	Data structures for gregorian dates (not using record buffer)
*DATER	None	I Spec	Data structures for gregorian dates in the record buffer #BUFIN
DESC	None	F Spec	File or program description
*DPARM	XFIELDVAL	S998	Retrieve all Data Dictionary values for videos
*DPRMS	XFIELDVL2	S998	Retrieve Data Dictionary values for detail subheading reports
*DPRMR	XFIELDVAL	S998	Retrieve Data Dictionary values for total subheading reports
DSPF	None	Various	Variable name substitution for display file(s) fields
DSP1	XDSPFLD1	S004	Display logic for primary video fields
	XDSPFLD2	S004	Format Alpha field for output
	XDSPFLD3	S004	Format Gregorian Date for output
	XDSPFLD4	S004	Format Julian Date for output
	XDSPFLD5	S004	Format VC0 field from VTX
	XDSPFLD6	S004	Format VC0 field from designated description file (field details)
	XDSPFLD7	S004	Format VC0 field from F0005
	XDSPFLD8	S004	Format Alpha 3 or 28
			Repeat of XDSPFLD1



Directive Code	Detail Logic Module	Source Created	Functional Directive
DSP2	XDSPFLD1	S004	Display logic for primary video fields
	XDSPFLD2	S004	Format Alpha field for output
	XDSPFLD3	S004	Format Gregorian Date for output
	XDSPFLD4	S004	Format Julian Date for output
	XDSPFLD5	S004	Format VC0 field from VTX
	XDSPFLD6	S004	Format VC0 field from designated description file (field details)
	XDSPFLD7	S004	Format VC0 field from F0005
	XDSPFLD8	S004	Format Alpha 3 or 28
*EMK	XLOADEMK	S999	Repeat of XDSPFLD1
ENTRY	X*ENTRY P	Various	Load user defined error messages
	X*ENTRY M		
*EXITC	XEXIT-CMD0	S00EX	Load program execution passed parameters
	XEXIT-CMD1		
*EXIT S	XEXIT-SEL0	S00OP	Function key exit execution logic
*EXIT W	XEXIT-SEL0	S000P	Selection exit execution logic
*FIELD	XFIELDDEDT1	S005	Selection exit execution logic
	XFIELDDEDT2	S005	Active Data Dictionary field validation logic
	XFIELDDEDT3	S005	Data Dictionary alpha edit
	XFIELDDEDT4	S005	Validation n=Master – Alpha
	XFIELDDEDT5	S005	Gregorian edit
	XFIELDDEDT6	S005	Julian edit
	XFIELDDEDT7	S005	Data Dictionary numeric edit
	XFIELDDEDT8	S005	Alpha field size 10
	XFIELDDEDT9	S005	User defined code edit
	XFIELDDEDT10	S005	No dictionary
	XFIELDDEDT11	S005	Validation n = Master – Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTC	S005	Cost center edit
	XFIELDDEDTE	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTS	S005	Validation = Master – Alpha
	XFIELDDEDTT	S005	Validation = Master – Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master – Numeric
FILES	None	F spec	Program file descriptions

Directive Code	Detail Logic Module	Source Created	Functional Directive
*FLDxx			Active Data Dictionary field validation for primary data
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n=Master – Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master – Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTTC	S005	Cost center edit
	XFIELDDEDTTE	S005	Numeric field size 7
	XFIELDDEDTTR	S005	Right adjust
	XFIELDDEDTTS	S005	Validation = Master – Alpha
	XFIELDDEDTT	S005	Validation = Master – Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master – Numeric
INFDS			File information data structures, if specified
	XINFDS1	I spec	Standard database file information data structure. The field prefix is incremented from \$1 to \$x where x = number of files
	XINFDS2	I spec	OBSOLETE. Use SRVFDS.
KEYI			Load master file key fields for inquiry programs.
	XFIELDLD1	S003	Load video input – Alpha
	XFIELDLD2	S003	Load video input – Numeric
	XFIELDLD3	S003	Load video input – Cost Center
	XFIELDLD4	S003	Load video input – Julian Date
	XFIELDLD5	S003	Load video input – Gregorian Date

Directive Code	Detail Logic Module	Source Created	Functional Directive
KEYS	XFIELDLD1	S003	Load master file key fields in subfile format.
	XFIELDLD2	S003	Load video input – Alpha
	XFIELDLD3	S003	Load video input – Numeric
	XFIELDLD4	S003	Load video input – Cost Center
	XFIELDLD5	S003	Load video input – Julian Date
	XFIELDLD5	S003	Load video input – Gregorian Date
	XNEXT-NBR	S003	Load video input – Next Numbering
KEYS2			Load master file key fields in primary video format
	XFIELDLD1	S005	Load video input – Alpha
	XFIELDLD2	S005	Load video input – Numeric
	XFIELDLD3	S005	Load video input – Cost Center
	XFIELDLD4	S005	Load video input – Julian Date
	XFIELDLD5	S005	Load video input – Gregorian Date
	XNEXT-NBR	S005	Load video input – Next Numbering
KLIST	XKEYLIST	S999	Create data file key list
*LVLS	XSAVVAL1		Save report level break data
MF	None	Various	Variable name substitution for master database files
*MCUxx	None	S003	Cost center security logic where xx = master filed designation 1 thru 9
		S004	
		S00EX	
*OPEN	XFILEOPN1	S999	Open report program data files
OPTE	None	S005	Subfile processing condition test based on mandatory entry fields in subfile format
*OTOT	XPRTTOT1	S010	Print all report level totals
PDL	None	Various	User defined entry point
*RKYxx	None	S999	Load softcoding record key for reports where xx = master file designation 1 thru 9

Directive Code	Detail Logic Module	Source Created	Functional Directive
RPTD	XDSPFLD1	S004	Format data for report detail format
	XDSPFLD2	S004	Format Alpha field for output
	XDSPFLD3	S004	Format Gregorian Date for output
	XDSPFLD4	S004	Format Julian Date for output
	XDSPFLD5	S004	Format VC0 field from VTX
	XDSPFLD6	S004	Format VC0 field from designated description file (field details)
	XDSPFLD7	S004	Format VC0 field from F0005
	XDSPFLD8	S004	Format Alpha 3 or 28
RPTH	XDSPFLD1	S004	Repeat of XDSPFLD1
	XDSPFLD2	S004	Format data for report heading format
	XDSPFLD3	S004	Format Alpha field for output
	XDSPFLD4	S004	Format Gregorian Date for output
	XDSPFLD5	S004	Format Julian Date for output
	XDSPFLD6	S004	Format VC0 field from VTX
	XDSPFLD7	S004	Format VC0 field from designated description file (field details)
	XDSPFLD8	S004	Format VC0 field from F0005
*RPTT	XDSPFLD1	S004	Format Alpha 3 or 28
	XDSPFLD2	S004	Format data for report total format
	XDSPFLD3	S004	Format Alpha field for output
	XDSPFLD4	S004	Format Gregorian Date for output
	XDSPFLD5	S004	Format Julian Date for output
	XDSPFLD6	S004	Format VC0 field from VTX
	XDSPFLD7	S004	Format VC0 field from designated description file (field details)
	XDSPFLD8	S004	Format VC0 field from F0005
*RTA	XTOTARRY	E spec	Format Alpha 3 or 28
*RTS	None	I spec	Load totaling arrays
*RTX	None	I spec	Report softcoding array
*RTXI	XVTIDX	S999	Report softcoding text fields
			Set maximum VTX index used

Directive Code	Detail Logic Module	Source Created	Functional Directive
*SFFLD			Active Data Dictionary data field validation for subfile fields.
	XFIELDED1	S005	Data Dictionary alpha edit
	XFIELDED2	S005	Validation n = Master – Alpha
	XFIELDED3	S005	Gregorian edit
	XFIELDED4	S005	Julian edit
	XFIELDED5	S005	Data Dictionary numeric edit
	XFIELDED6	S005	Alpha field size 10
	XFIELDED7	S005	User defined code edit
	XFIELDED8	S005	No dictionary
	XFIELDED9	S005	Validation n = Master – Numeric
	XFIELDEDTA	S005	Account ID
	XFIELDEDTB	S005	Cost center edit
	XFIELDEDTD	S005	Numeric field size 7
	XFIELDEDTT	S005	Right adjust
	XFIELDEDTS	S005	Validation = Master – Alpha
	XFIELDEDTT	S005	Validation = Master – Alpha Rt Adj
	XFIELDEDTU	S005	Validation = Master– Numeric

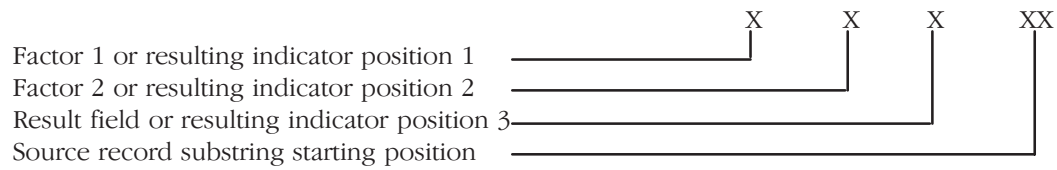
Directive Code	Detail Logic Module	Source Created	Functional Directive
SLDxx			Active Data Dictionary data field validation for subfile data fields. Where xx = specified master file 1 thru 9.
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n = Master – Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master – Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTTC	S005	Cost center edit
	XFIELDDEDTTC	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTTS	S005	Validation = Master – Alpha
	XFIELDDEDTT	S005	Validation = Master – Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master– Numeric
*S00VL	None	I spec	Cursor Control, F1
TITLE	None	H spec	Program title
*VKYxx	None	S999	Load softcoding record key for display files where xx=display file designation 1 – 9.
*VTS	None	I spec	Display file softcoding array
*VTX	None	I spec	Update softcoding text field ending positions based upon size definition in display file
*VTXI	XVTXIDX	S999	Set maximum VTX index used

\*J.D. Edwards standards included automatically which are above and beyond normal requirements

## Substitution Directives

- Control the translation of symbolic names to the actual data field names required for an individual line of source code.
- Actually substitute information within a line of code.
- If a field is going to be replaced, the field being replaced begins with an '&'.
- If the substitution is going to be positional, this directive tells the generator where to place something on a line of code.

Columns 1 to 5



Directive	Column Allowed				Function
	1	2	3	45	
@	x	x	x	x	Four character Data Dictionary name
#	x				Primary passed parameter for *ENTRY
A	x	x	x		Highest VTX field defined.
B					Unused at this time.
C	x				Function key exit indicator test
D	x			x	Descriptions for fields, files, and copy modules
E		x			Error message key
F		x			Validation file name
G		x			User defined calculation logic result field name
H		x			Descriptive display file name
I	x	x	x		Display field error condition attribute indicator
J	x	x	x	x	Data file names
K	x	x	x		Descriptive display file key field name
L	x				Data file key list name and optional file/format name
M	x				File information data structure name
N	x	x	x		Full data field name (Write to)
O	x			x	Common subroutine name
P		x			Function key/selection exit program to execute

Directive	Column Allowed				Function
	1	2	3	45	
Q			x		Field name to receive returned description value
R			x		Field name to receive returned key value
S		x			Selection exit value test
T	x				Function key/selection exit
U	x				File information data structure subfield prefix
V	x	x	x		Source of data (Read From) field name
W			x		Data file key list key field name
X			x	x	Error message array index
Y			x		Function key/selection exit parameter field name
Z			x		Numeric field size definition (right adj alpha)
0	x				Gregorian date Data Structure numeric 6 byte date
1	x				Gregorian date Data Structure numeric 2 byte month
2	x				Gregorian date Data Structure numeric 2 byte day
3	x				Gregorian date Data Structure numeric 2 byte year
4	x	x	x		Parameter 1 from *PROC calculations
5	x	x	x		Parameter 2 from *PROC calculations
6	x	x	x		Parameter 3 from *PROC calculations
7	x	x	x		Parameter 4 from *PROC calculations
8	x	x	x		Parameter 5 from *PROC calculations



## Exception Directives

- Mostly fall in the category of substitution directives but are out of the normal syntax used by the substitution directives.
  - Provide unusual option definition to the program generation process.
- Combines two other types of directives.

Example:

DSPF      &01FILE

- Combines a Functional directive (DSPF) with a Substitution directive (&01FILE), so it is an Exception directive.

Example:

Create a line of code for the READ Master file and then substitute the Master file name.

Factor 1	Factor 2	Result	Keyword	Function
	x		&xxFILE	Master/video/report file name
	x		&xx(FILE)	File name in single quote marks
	x		&xxFORMAT	Master/video format name
	x		&xxFORMAT1	Subfile line 24 format name
	x		&xxFORMATC	Subfile control record format name
	x		&xxFORMATS	Subfile record format name
x	x		&xxKEYFLD	Master file primary key field name
x	x		&xxPGCTL	Number of subfile records in 1 page
x			%	Factor 1 intentionally left blank
x	x	x	=	User defined calculation logic result

## Conditional Directives

- Most flexible and most powerful.
- Checks for specific condition(s) before determining what, if anything should be done.
- Subroutine S010–11 contains good examples of these.

Example:

If SFSELC exists, include code for selection exits.

- Uses positions 1 to 5 to provide directive initiation and uses Factor 1, Factor 2 and the Result field to complete the directive definition. You can combine conditional directives.

<b>Pos 1</b>	+	Include detail logic module if true
	–	Include detail logic module if false
<b>Pos 2–5</b>	FLDN	Test existence of data field
	DTAI	Test existence of data item
	FILE	Test existence of file
	FMT	Test existence of file
<b>Factor 1</b>	Name of field, item, file or format to test. May also contain *ANYx for file test which can be used to test for types of files used in a program where x may optionally designate number of files	
<b>Oper (file test only)</b>	DSPF	Display file
	PF	Physical file only
	LF	Logical file only
	PRTF	Printer file only
	DB	Database file

<b>Factor 2</b>		Name of detail logic module to include into source code. May also use *AND to produce compound conditions
<b>Result Field Pos 1</b>	@	Any input file
	M	Master input file with M designation in file specifications
	1-9	Master input file with 1 – 9 designation in field specifications
<b>Result Field Pos 2</b>	@	Any output file
<b>Result Field Pos 3</b>	@	Any update file
	M	Master update file with M designation in file specifications
	1-9	Master update file with 1 – 9 designation in field specifications
<b>Result Field Pos 4</b>	@	Any add file



### Exercises

See the exercises for this chapter.



# Work with the Question and Answer System

---

## Working with the Question and Answer System

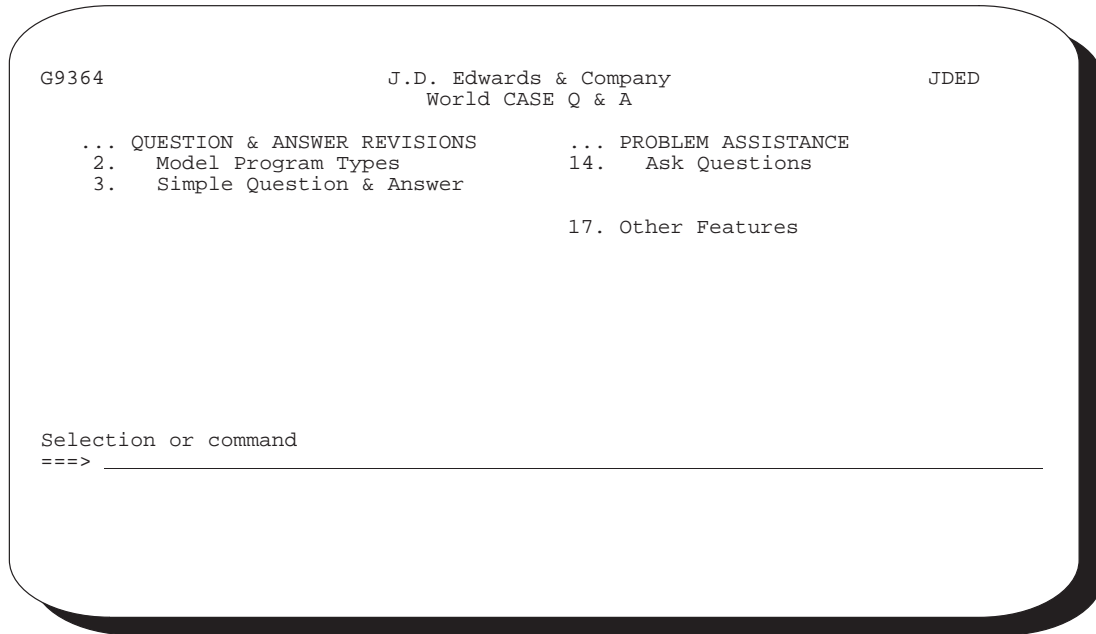
The program generator uses J.D. Edwards Question and Answer system as a method of determining the appropriate program type. Based on the answers you give to certain questions, the system selects a program type for you. You can modify the dialogue the program generator uses through this feature. You can also create your own questions and answers to arrive at your own custom program type.

You should be able to:

- Review questions
- Add new Q&A Dialogue
- Review Dialogue
- Change Dialogue
- Copy Dialogue
- Rename a Dialogue
- Run a Dialogue
- Delete a Dialogue
- Run a Quiz

From the Model Program Design menu, select Maintain Q/A. The World CASE Q & A menu appears.

The Question and Answer System allows you to work with question and answer dialogue.



The screenshot shows a terminal window with a menu for 'World CASE Q & A'. The menu is titled 'G9364' on the left, 'J.D. Edwards & Company' and 'World CASE Q & A' in the center, and 'JDED' on the right. The menu options are listed in two columns: '... QUESTION & ANSWER REVISIONS' (with sub-options 2. Model Program Types and 3. Simple Question & Answer) and '... PROBLEM ASSISTANCE' (with sub-options 14. Ask Questions and 17. Other Features). At the bottom, there is a prompt 'Selection or command' followed by '==>' and a horizontal line for input.

```
G9364                                J.D. Edwards & Company          JDED
                                World CASE Q & A

... QUESTION & ANSWER REVISIONS      ... PROBLEM ASSISTANCE
2.   Model Program Types              14.   Ask Questions
3.   Simple Question & Answer

                                17. Other Features

Selection or command
==> _____
```

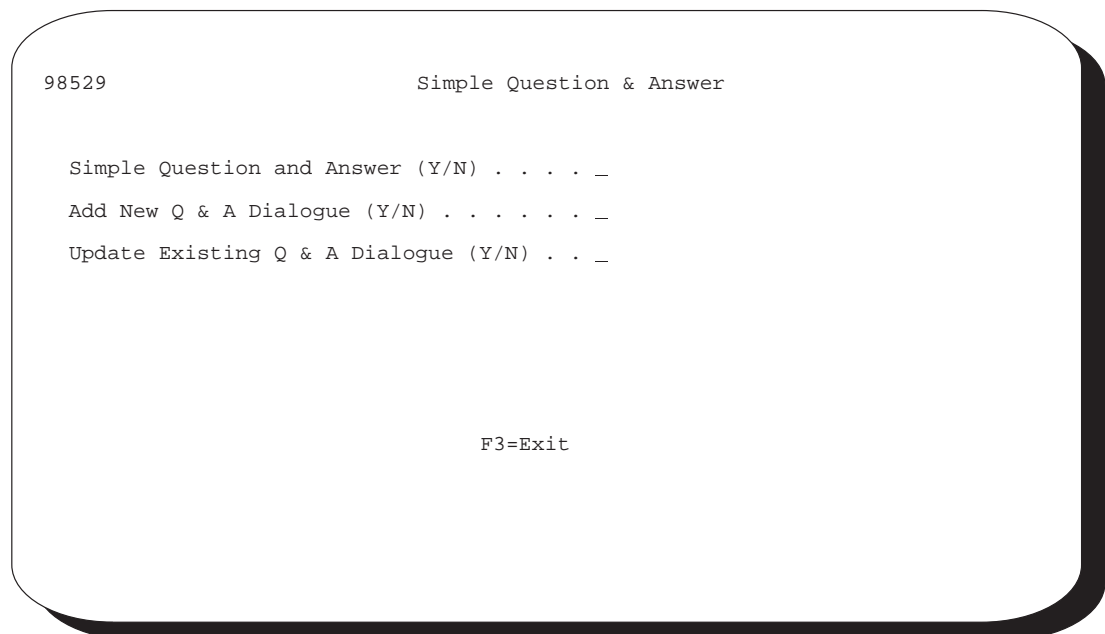
## Menu G9364, Option 3 — Simple Question & Answer

The Simple Question & Answer form provides several options.

**Simple Question and Answer** Takes the user to *Question Entry*

**Add New Q & A Dialogue** Takes the user to *Dialogue Descriptions*

**Update Existing Q & A Dialogue** Takes the user to a *Dialogue Lists* screen

A screenshot of a terminal window showing a menu titled "Simple Question & Answer". The window has a title bar with "98529" on the left and "Simple Question & Answer" on the right. The menu lists three options: "Simple Question and Answer (Y/N) . . . . \_", "Add New Q & A Dialogue (Y/N) . . . . . \_", and "Update Existing Q & A Dialogue (Y/N) . . \_". At the bottom, it says "F3=Exit".

```
98529                               Simple Question & Answer

Simple Question and Answer (Y/N) . . . . _
Add New Q & A Dialogue (Y/N) . . . . . _
Update Existing Q & A Dialogue (Y/N) . . _

F3=Exit
```

## Reviewing Questions

### ▶ To review questions contained in a master dialogue

---

1. On the Simple Question and Answer screen, in the Simple Question and Answer field, enter Y. The Question Entry screen displays.
2. On the Question Entry screen, type I in the Action Code field. Enter the question number. The question detail displays.

98551	Question Entry		
Action Code. . . . I			
Question Number. . . 00000123			
System Code. . . . 93	Category . TUTORIAL	Release. . A71	
Subject(noun). . . Generator	Show(Y/N). _	Tickler. .	
Addl Keywords. . .			
Question Description:			
<u>Of what general type is the program?</u>			
OR			
<u>If you know the correct logic type enter the</u>			
<u>desired value where indicated.</u>			
F19/F20=Next/Previous Question		Roll Up = Additional Text Lines	

- You can make changes to the master question displayed.
- The F19 and F20 keys allow you to roll through all master questions.
  - You will see ALL questions, not just the ones utilized by the selected dialogue.
- You may review the answers for the master question displayed by entering a “C” in the action code and pressing Enter.

## Adding New Q & A Dialogue

### ► To add new Q&A dialogue

---

1. From the Simple Question & Answer screen, enter Y in the field Add New Q & A Dialogue. The Dialogue Descriptions screen displays.
2. Complete the screen. The following illustration shows a sample screen.



```

98541                               Dialogue Descriptions

Dialogue Keys: Primary . . . . *DEFAULT
               : Secondary . . . LC

Dialogue Type. . . . . TUTORIAL

Summary description . . . . . Choose a clone program type

Beginning Question Number. . . 00000123

Dialogue description.

The following tutorial is designed to help you choose a program type
for the program generation process.
_____
_____
_____

Enter=Continue           F12=Previous Screen
  
```

3. Press Enter. The first Question Entry screen appears. Type A in the Action Code and complete the screen. The following illustration shows a sample screen. Add additional keywords to assist in future searches for this question.

```

98551                               Question Entry

Action Code. . . . A

Question Number. . 00000123

System Code. . . . 55      Category . TUTORIAL  Release. . A51
Subject(noun). . . Generator Show(Y/N). _      Tickler. . 
Addl Keywords. . . PROGRAM _____

Question Description:
Of what general type is the program?
OR
If you know the correct logic type enter the
desired value where indicated.
_____
_____
_____

F19/F20=Next/Previous Question      Roll Up = Additional Text Lines
  
```

4. Press Enter. The Answer Entry screen displays. Type A in the Action Code field and complete this screen. The following illustration displays a sample answer.

When the user returns the answer, the next question is 131, as shown in the Next Question field in the illustration above. There is no return value. For any question, there is either a return value or a next question.

If inquiring on an existing question and answer use F19 or F20 to roll through all other possible answers for this question.

5. To create a second answer to the question, press Enter. The answer clears. Type the number of the next answer in the Answer Number field. Type the new next question and a return value if necessary. Enter the text for the next answer.

NOTE: The Return Value field is optional. The screen below shows the use of the Return Value field. In this illustration, the return value is the program type for an interactive window program. In this case, there is no next question. The dialogue ends after returning the value E0010 to the calling program.

The Return Value field can contain a member name, or \*PROMPT. \*PROMPT lets the user manually complete the Return Value field.

```
98552                                Answer Entry
Action Code. . I                     Question No. . 00000123   Answer Number. 00002
Question . . . Of what general type is the program?
                        OR
                        If you know the correct logic type enter the
                        desired value where indicated.
Next Question. 00000000   Return Value . E0010
Answer . . . .           An interactive window program
                        _____
                        _____
```

6. To define the next question, press F3 to return to the Question Entry screen. Complete the screen for the question and press Enter to display the Answer Entry screen.
7. When the questions and answers are complete, press F3 until the Simple Question & Answer screen displays.

## Inquiring on a Dialogue

### ► To review a dialogue

1. From the Simple Question and Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.

98530	Dialogue Lists			Type . . .
Opt	Member	Data Item	Type	Description
—	ASM	1	QUIZ	MI language quiz #1
—	CLONE2.5	TEST1	QUIZ	Training Class Day 1 Quiz
—	CLONE2.5	TEST2	QUIZ	Training Class Day 2 Quiz
—	RPG	1	QUIZ	RPG language quiz #1
—	*DEFAULT	##PE	TUTORIAL	Define editing program
—	*DEFAULT	CMD	TUTORIAL	What Report Writer to Use
—	*DEFAULT	KOPT	TUTORIAL	Mandatory processing options
—	*DEFAULT	LC	TUTORIAL	Choose a clone program type
—	*DEFAULT	OC	TUTORIAL	Determine menu option code
—	RAPID	SCREEN	TUTORIAL	Quick Screen Creation

Opt: 2=Chg 3=Cpy 5=Run 6=Flow 7=Rename 9=Dlt 11=Quiz F24=More Keys

## Selection Exits

### 2 – Change

- Change the Q&A for the Dialogue

### 3 – Copy

- Copies one Dialogue to another Dialogue

### 5 – Run

- Run the Q&A
- Can specify the number of responses to allow

### 6 – Flow

- Shows the flow of the Q&A
- How one question leads to another

- Can exit to Q&A revisions from here

7 – Rename

9 – Delete

11 – Quiz

- If the dialogue is a 'Quiz', the user can take the quiz from this screen

2. Enter 6 in the Opt (Option) field. The Dialogue Flow Revisions screen displays.

```
98531                               Dialogue Flow Revisions
Dialogue Key: Primary. . . *DEFAULT   Secondary. . . LC
Q  Question      Text
- 00000123 Of what general type is the program?
      OR
      If you know the correct logic type enter the
      desired value where indicated.
      Answer(s) to Question
Ans  1 .---Next Question = 00000131 Return Value =
      An interactive program
Ans  2 .---Next Question = 00000000 Return Value = E0010
      An interactive window program
Ans  3 .---Next Question = 00000254 Return Value =
      Print a report
Ans  4 .---Next Question = 00000262 Return Value =
      Conversion program
Ans  5 .---Next Question = 00000271 Return Value =
      Batch update program
Ans  6 .---Next Question = 00000000 Return Value = *PROMPT
      Desired logic type is:
Opt:   2=Revision                      F11=Alternate Format
```

Press F11 for the Alternate Format.

```

98531                               Dialogue Flow Revisions

Dialogue Key: Primary. . . *DEFAULT   Secondary. . . LC

Q  Question      Text
- 00000123 Of what general type is the program?
      OR
      If you know the correct logic type enter the
      desired value where indicated.
      Answer(s) to Question
Ans  1 An interactive program
Ans  2 An interactive window program
Ans  3 Print a report
Ans  4 Conversion program
Ans  5 Batch update program
Ans  6 Desired logic type is:

- 00000131 Does the display file for this interactive program
      contain a subfile? If the program simply uses one
      display format with no subfile you should answer
      "NO".
      Answer(s) to Question

Opt:      2=Revision                      F11=Alternate Format
  
```

## Changing a Dialogue

### ► To change a dialogue

1. Inquire on the dialogue to change.
2. Enter 2 in the O (Option) field next to the question to change. The Question Entry screen displays.
3. Type C in the Action Code. Make the changes to the questions and answers.

## Copying a Dialogue

### ► To copy a dialogue

---

1. From the Simple Question & Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.
2. In the Opt field, enter 3. The Dialogue Copy screen displays.

98536 Dialogue Copy

Dialogue list keys

Member . . . . .	<u>*DEFAULT</u>
Data item . . . . .	<u>LC</u>

New dialogue list keys

Member . . . . .	<u>          </u>
Data item . . . . .	<u>          </u>

From Library . . . . . JDFCLONE

To Library . . . . .           

F12=Previous Screen

3. Type the name of the new primary key in the field New dialogue list keys, Member. Type the name of the new secondary key in the field New dialogue list keys, Data item. Type the name of the library in which the new dialogue resides. Press Enter. The system copies the dialogue.

## Rename a Dialogue

### ► To rename a dialogue

1. From the Simple Question & Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.
2. In the Opt field, enter 3. The Dialogue Copy screen displays.

98536 Dialogue Copy

Dialogue list keys

Member . . . . .	<u>*DEFAULT</u>
Data item . . . . .	<u>LC</u>

New dialogue list keys

Member . . . . .	<u>          </u>
Data item . . . . .	<u>          </u>

F12=Previous Screen

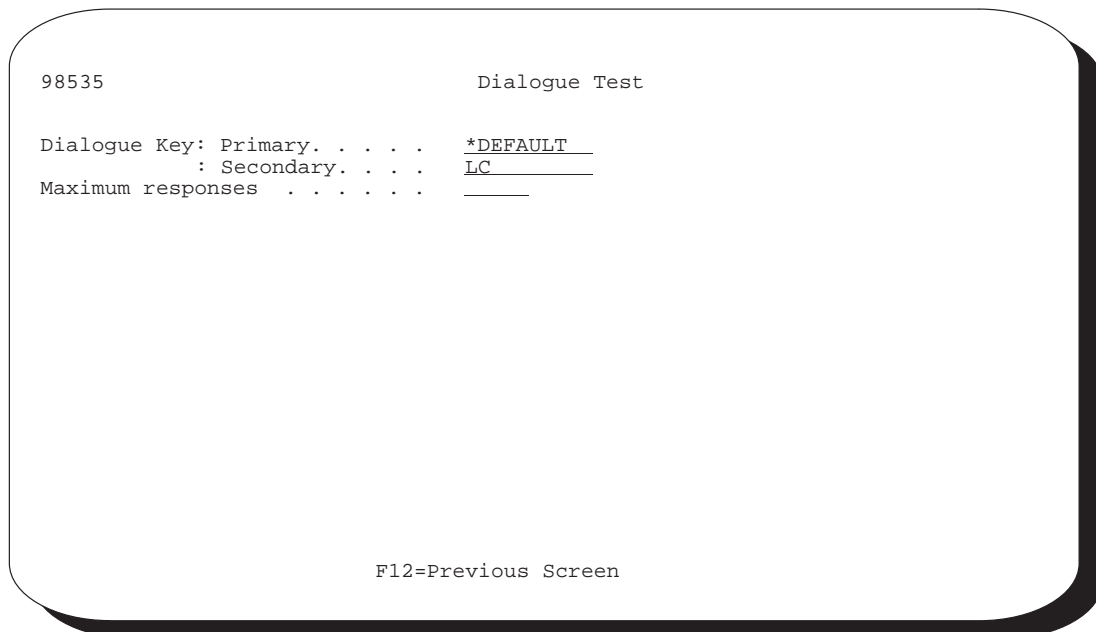
3. Type the new name of the primary key in the field New dialogue list keys, Member. Type the new name of the secondary key in the field New dialogue list keys, Data item. Press Enter. The system renames the dialogue.

## Running a Dialogue

### ► To run a dialogue

---

1. From the Simple Question & Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.
2. In the Opt field, enter 5 next to the dialogue to run. The Dialogue Test screen displays.



98535 Dialogue Test

Dialogue Key: Primary. . . . .	<u>*DEFAULT</u>
: Secondary. . . . .	<u>LC</u>
Maximum responses . . . . .	<u>      </u>

F12=Previous Screen

3. Enter the maximum number of times to run this dialogue. The questions of the dialogue display in sequence. When you reach the last question, a message appears at the bottom of the screen. The following screen illustrates this message.



98533	Dialogue Selection	*DEFAULT LC
<p>The following tutorial is designed to help you choose a program type for the program generation process.</p>		
<p>Question:</p> <p>Does the Transaction file have a unique key?</p>		Opt
<p>Responses:</p> <p>Yes..... X</p> <p>No..... _</p>		
<p>Question &amp; Answer complete, To review press F5 else press Enter.</p>		Bottom

- If you press F5 on the last question screen, the Quiz Answer Review screen displays.

98534	Quiz Answer Review	*DEFAULT LC
<p>The following tutorial is designed to help you choose a program type for the program generation process.</p>		
<p><u>Answers to Questions at This Time</u></p> <p>_ Of what general type is the program?</p> <p style="padding-left: 40px;">OR</p> <p style="padding-left: 40px;">If you know the correct logic type enter the desired value where indicated.</p> <p style="padding-left: 40px;">An interactive program</p> <p>_ Does the display file for this interactive program contain a subfile? If the program simply uses one display format with no subfile you should answer "NO".</p> <p>Yes</p>		
<p>Opt: 4=Return to Question</p>		More...

The screen displays the questions and the answers you entered. To review the remaining questions and answers, use the roll keys. Enter 4 in the O field to return to a specific question.

- If you press Enter on the last question screen, the Dialogue Test screen displays.

98535	Dialogue Test		
Dialogue Key: Primary. . . . .		<u>*DEFAULT</u>	
: Secondary. . . . .		<u>LC</u>	
Maximum responses . . . . .		<u>10</u>	
Dialogue type . . . . .		TUTORIAL	
Description . . . . .		Choose a clone program type	
Responses returned . . . . .		3	
Array	Question	Response	Answer
Offset	Number	Returned	Number
1	00000174	D0100	00001
2	00000174	D0100	00001
3	00000174	D0100	00001

F12=Previous Screen

Bottom

This screen displays the number of times the dialogue was run, the response returned at the end of the dialogue, and the number of the answer to the last question which returned the response.

Press F3 from the Dialogue Test screen to return to the Dialogue Lists screen.

## Deleting a Dialogue

### ► To delete a dialogue

---

1. From the Simple Question & Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.
2. In the Opt field, enter 9. The system deletes the dialogue.

## Running a Quiz

### ► To run a quiz

---

To run a quiz, the dialogue type must be QUIZ.

1. From the Simple Question & Answer screen, enter Y in the Update Existing Q & A Dialogue. The Dialogue Lists screen displays.
2. In the Opt field, enter 11 next to the quiz to run. The first question of the quiz displays.

3. Answer the questions. When you finish answering the questions, a message displays at the bottom of the last screen, "Question and Answer complete, To review press F5 else press Enter."
  - If you press F5, the questions and answers display on the screen. If you press Enter, the system calculates the number of errors and displays your score. Press F5 from this Dialogue Test screen to review your errors.

```

98537                               Dialogue Test

Dialogue Key: Primary. . . . . CLONE2.5
               : Secondary. . . . . TEST1

Quiz description . . . . . Training Class Day 1 Quiz

Total questions in quiz . . . . . 16
Incorrect answers . . . . . 5
Score . . . . . 69 % Time to hit the books.
    
```

F5=Review Incorrect Answers F12=Previous Screen

## Guidelines

The dialogue the CASE tool uses to determine the program type is Primary Key  
 \*Default, Data Item LC.



### Exercises

See the exercises for this chapter.



# Create User Defined PDL

---

## About User Defined PDL

Currently \*PROCs have to be attached to either a master file field or to a device file field (video/report). If it is attached to a master file field, then the generated code will be placed in S005. If it is attached to a device file field, then the generated code will be placed in S004.

The purpose of User Defined PDL Entry Points is to allow the user to create \*PROCs in any subroutine and to allow them to exist without being attached to a master file field or device file field.

- A new feature of the Program Generator as of the PTF A52PC000T1.
- A functional directive that the user can enter into a primary logic module.
- Causes RPG code to be created in the same way as through the PDL that users enter through the Detailed Programming Facility, but is connected to logic modules instead of fields.
- Defines entry points within subroutines where the user can enter PDL code via the Detailed Programming Facility.

## Creating a User Defined PDL

### ► To create a user defined PDL

1. Determine which program type is affected, and the names of the logic modules within the program type where you want to create a PDL entry point.
2. For all single record maintenance videos, you create a user defined PDL entry point in the mainline subroutine.
3. Enter PDL to bring in a default value for a constant field.

```

93001                                Create/Modify Program Types
Action Code. . . . . I
Program Type . . . . . B0010      STD/M - Action Code
Seq  Prim Modul Glossary K
1.00 FILEDEFN01 File Specification
2.00 FILEEXTN0  Tables & Arrays - STD Video
3.00 INPUT1     Data Structures - STD Video
4.00 MAINLINE   Mainline - Video
5.00 S00EX-1    Exits Subroutine - STD Video
6.00 S00QP      Options Subroutine
6.50 S00VL-1    Return Values Subr - Standard
7.00 S001-1     Clear Subroutine - STD Video
8.00 S003-1     Edit Key - STD Video
9.00 S004-1     Load Display Subr - STD Video
10.00 S005-1    Edit Subroutine - STD Video
11.00 S010-1    Update Subroutine - STD Video
12.00 S999-1    Housekeeping Subr - STD Video
_____
_____
_____
F24=More

```

The logic module that you will change is MAINLINE because this creates the mainline code for all single record maintenance videos.

4. Create the user defined PDL entry point(s) within the affected primary logic modules.
  - The naming convention for user defined PDL entry points is PDLxx, where xx is a two digit number between 01 and 99.
  - You may either add the PDL directive to an existing line of code that does not contain a directive, or insert a new line and put the directive on this line. The directive goes in positions 1 to 5 of the source line. If you insert a new line, remember to add the source sequence and serial number in the appropriate columns. (Window over to column 80.)

```

Columns . . . :   1  71           Edit           AHTEST/F93001WRK
SEU==>                                     F93001
FMT *   ..... *. 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7
***** Beginning of data *****
0001.00      C*****
0002.00      C*   MAINLINE PROGRAM
0003.00      C*   -----
0004.00      C*
0005.00      C*   Process housekeeping.
0006.00      C*
0007.00      C               EXSR S999
0008.00      C*               ----
0009.00 PDL01/*
0010.00      C*
0011.00      C*   If LR on, end program.
0012.00      C*
0013.00      C               *INLR      CABEQ'1'      EOJ
0014.00      C*               -----      ---
0015.00      C*
0016.00      C*   If automatic inquiry set, process inquiry.

F3=Exit   F4=Prompt   F5=Refresh   F9=Retrieve   F10=Cursor
F16=Repeat find   F17=Repeat change   F24=More keys

```

In this example, PDL01 has been entered. Any PDL code entered for this entry point will come immediately after the statement EXSR S999 and before the test for \*INLR.



Limitations

In any primary logic module you may insert up to 99 PDL directives. Ordinarily you would number the first one PDL01, the second one PDL02, and so on. However, it is not required that the PDL directives be in sequential order. It is required that each PDL directive have a unique number within that logic module.

Enter the PDL code through the Detailed Programming Facility.

All user defined PDL entry points will appear after the display/report file fields in the Detailed Programming Facility.

93105

Detailed Programming Facility

Program Name: P55TPDL  
Test PDL Entry Points

Locate  
File Name : V55TPDL      Test PDL Entry Points  
Field Name : VDPH1      Phone Number

O

P Purpose

— Phone Number

—

—

—

—

—

—

—

—

ZPDL User Defined PDL Entry Points

6 User Defined PDL Entry Points

— User Defined PDL Entry Points

ZPDL User Defined PDL Entry Points

... Data Flow ... KY R D

Read From Write To PS A D

ABPH1 VDPH1 — — N

VTX001 — — N

VTX002 — — N

VTX003 — — N

VTX004 — — N

VTX005 — — N

VTX006 — — N

VTX007 — — N

PDL01 MAINLINE — — —

PDL02 MAINLINE — — —

Opt: 2=Data Dic 4=Field Dtl 6=\*PROC 9=Dlt Fld F24=More



5. Enter the PDL code in the usual manner.

```
93109                                Data Item Formula Revisions

Action Code. . . . . I
Program ID . . . . . P55TPDL Test PDL Entry Points
File ID. . . . . MAINLINE
Field Name . . . . . PDL01

Data Item Formula
\This code will be brought into the Mainline portion\
\of the program, immediately following the EXSR S999 statement.\
Begin
  If $auto = '1' Then
    Begin
      If vdxit = ' ' Then
        vdxit := ' 1001'
      End;
    End
  End
```

F5=Variables

F3=Exit

F24=More



# Appendices



# Appendix A – Program Generator Checklist

---

This brief section is a tool to help users when they are using the Program Generator. It consists of items that should be considered and/or remembered as the user generates programs.

## Data File Design Aid

- Try to create files with keys to avoid having to process by relative record numbers.
  - More complicated than processing by keys.

## Screen Design Aid

- Note the video fields that VC0 fields are related to.
  - You will need this information in the Detailed Programming Facility in order to load the VC0 fields.
- Subfile Programs
  - If a maintenance subfile, define a hidden field for the parts of the file key used for the subfile video.
  - Define the hidden field, SH#RRN if processing by relative record numbers.
  - Define a hidden field for the data structure if processing by relative record numbers.
- Make a note of the error indicators assigned to screen fields.
- Final checks
  - Is the Default Cursor keyword assigned to a video field?
  - Have you allowed for upper and lower case on description fields?
  - Is a 'K' specified in the 'Edited' field for the key fields only?

## Report Design Aid

- Change the Start/End lines for format HEADING1 from 1-4 to the needed length.
  - Usually 1-8 will suffice.
- Add DETAIL1 format.
- Add TOTAL1 format if using Hierarchical (dynamic) totaling.
  - MUST include one or all of the following fields:

VC1ROW, VC1KEY, VC1DSC, VC0TO2

- These are the fields that enable dynamic totaling to work.
- Add HEADING2 format if using subheadings.
  - Must include one or all of the following fields:

VC0ROW, VC0KEY, VC0DSC

- These are the fields that enable subheadings to work properly.

## The Program Generator

- General for All Programs:
  - Make sure the CAP Status is set to 'Y'.
  - If CAP Status is not 'Y', then something could have ended abnormally.
  - Verify the program type.
- File Specifications
  - Must have one file specified with an 'M'.
  - Do not have one file specified with an 'M' and another specified as a '1' as the Program Generator views both files on an equal basis.

Specify the main file as an 'M' and subsequent files starting with '2'.
  - Must specify a video or report file.
  - Do not include description files if a field is in the HEADING2 format for a report.
- Define Option and Function Key Exits
  - Make sure the called program is setup to accept parameters being passed by the function key or selection exit.

- Modify any CL programs that also call the called program to pass blank parameters.

i.e. CALL Pxxxxx PARM(' ')

- Program to be called must exist to be used in this screen.
- Always try to pass PSxxxx fields instead of VDxxxx or SFxxxx fields.

May inadvertently get changed in the called program.

Will have to define and load the PSxxxx fields manually.

### Detailed Programming Facility

- Specify 'N' in the Entry Optional field for key fields in a subfile.
  - Specify for the subfile fields, not the hidden fields.
  - This enables the delete function.
- Link VC0 fields to description files.
- PDL
  - If on the data base field, will affect subroutine S005.
  - If on the video field, will affect subroutine S004.
  - Use the Return keyword if you want to replace the standard code generated by the Program Generator.
- Specify a PLIST sequence if the program is going to receive parameters from another program. Use the video field(s) for this instead of the file fields.
- Use \*OUTPUT to get the row description from the Data Dictionary for fields that are only being used in the HEADING2 format and not the DETAIL1 format.





## Appendix B – Programming Standards

### Error Handling

J.D. Edwards has devised an efficient means of handling errors by way of arrays. The examples below show how the error handling arrays are defined within the Single Record Maintenance Program you generated earlier in this manual.

- The EMK array holds the four byte data dictionary name of every error that could occur in this program.
- The @MK array maintains a flag setting for each error identified in EMK. If one of the errors occurs, the flag is set.
- The @ER array loads the related error messages when the user presses F7 to view the errors that actually occurred.

The call to the error message handling program is shown in the following illustration.

```
Columns . . . : 1 71      Browse      JDFSRC71/JDESRC
SEU==>                                     P92801
0335.00      C*
0336.00      C*      If Display errors pressed, exit to error messages.
0337.00      C*      -----
0338.00      C*
0339.00      CSR      @@AID      IFEQ #FERRD
0340.00      CSR      Z-ADD1      #G
0341.00      CSR      Z-ADD1      #H
0342.00      CSR      #G      DOWLE64
0343.00      CSR      @MK, #G      IFEQ '1'
0344.00      CSR      MOVE EMK, #G      @ER, #H
0345.00      CSR      ADD 1      #H
0346.00      CSR      END
0347.00      CSR      ADD 1      #G
0348.00      CSR      END
0349.00      CSR      CALL 'P0000E'      98
0350.00      C*      -----
0351.00      CSR      PARM      @ER
0352.00      CSR      GOTO ENDEXE
0353.00      C*      -----
0354.00      CSR      END
0355.00      C*
```

If any error flag is set to one, the program moves the corresponding data item from the array of all possible errors (EMK) into the array of the errors that have actually occurred (@ER).

The next piece of code shows how a flag is set in the @MK array.

```

Columns . . . : 1 71          Browse          JDFSRC71/JDESRC
SEU==>          P92801
0770.00         C*
0771.00         CSR          *IN41      IFEQ '1'
0772.00         CSR          MOVE '1'    @MK, 2
0773.00         CSR          SETON
0774.00         CSR          END
                                     93

```

The standard indicator for an error (93) is set on and indicator 41 is set on to highlight the field in error.

The next piece of code shows the loading of the array that contains every possible error for this program. This loading takes place only once (in S999).

```

Columns . . . : 1 71          Browse          JDFSRC71/JDESRC
SEU==>          P92801
2605.00         C*-----
2606.00         C*
2607.00         C*      Load error messages array.
2608.00         C*
2609.00         CSR          MOVE '0001'  EMK,01      Inv Action
2610.00         CSR          MOVE '0002'  EMK,02      Inv Key
2611.00         CSR          MOVE '0003'  EMK,03      Inv Blanks
2612.00         CSR          MOVE '0004'  EMK,04      Inv Date
2613.00         CSR          MOVE '0005'  EMK,05      Inv Next Nbr
2614.00         CSR          MOVE '0007'  EMK,06      In Use
2615.00         CSR          MOVE '0025'  EMK,07      Inv Values
2616.00         CSR          MOVE '0026'  EMK,08      Inv MCU
2617.00         CSR          MOVE '0027'  EMK,09      Inv Desc Ttl
2618.00         CSR          MOVE '3438'  EMK,12      No SFL Rcds
2619.00         CSR          MOVE '3523'  EMK,13      Partial SFL
2620.00         CSR          MOVE '0052'  EMK,10
2621.00         C*-----

```

## Indicator Usage

There are 99 indicators available for use. They are grouped by purpose. JDE has defined standards for the use of the indicators specified in the following chart. JDE has not specified standards for indicators not mentioned.

### Indicator Usage

INDICATOR	DESCRIPTION
01	Causes the <i>Invalid Function Key Pressed</i> message to display
02	Dictates the color palette to be used
04	Controls subfile keywords SFLDROP and SFLFOLD for fold areas
20	Handles the clear screen action code
21	Handles the add action code
22	Handles the change action code
23	Handles the delete action code
24	Handles the inquire action code
31	Used in conjunction with subfile processing to initiate the INVITE or SFLCLR keyword
32	Used in conjunction with subfile processing initiating the keyword SFLNXTCHG
37	Used in conjunction with subfile processing to highlight the last record in the display (used only with inquiry subfiles)
38	Used in conjunction with subfile processing to control the display keyword SFLDSP
42-79	Used for error processing to indicate which fields are in error
40	Reserved for errors in the Action Code field
41	Reserved for errors in the key fields
80-89	General reusable one-time indicators
93	Global error indicator that highlights line 24
98	Indicates a chain or read failure
99	Indicates a record is in use
OF	Indicates overflow for report processing
LR	Indicates that the last record has been read and the program should end normally
RT	Indicates that a temporary or final halt in the program should take place and returns to the calling program leaving files open

## Naming Conventions

Use the following first character to distinguish different item names:

- @ — Array names

- \$ — Program created field names (flags and work fields)
- # — Fields defined in common subroutines

### Key List (KLIST)

Define key lists in the housekeeping subroutine.

Begin the key list name with the data file prefix. For example, the Address Book Master file prefix is AB, so the key list would be ABKY01.

The program generator creates key lists using the following naming conventions:

- XXKY01 for physical files where XX = the file prefix. For example: ABKY01
- When a physical needs to have more than one key list in a program, the successive files are noted in the last character space. For example, for three key lists for the physical F0101, the key lists would be: ABKY01, ABKY02, and ABKY03.
- XXKY0x for logical files where XX is equal to the file prefix and x is equal to the last letter of the logical file name. For example: ABKY0A for F0101LA, ABKY0B for F0101LB
- When a logical needs to have more than one key list in a program, the successive files are noted in the second to last character space. For example, the three key lists for the logical F0101LA would be: ABKY0A, ABKY1A, and ABKY2A.

### Work Fields

Define work fields only once within a program. The use of the LIKE DEFN command is highly recommended for defining work fields when their attributes are directly tied to those of database fields.

For example, if the work field needs to have the same attributes as a field that exists in a file:

```
MOVE ABANS $ANS,
```

then define \$ANS as follows:

```
*LIKE DEFN ABANS $ANS
```

The advantage of this method is that the work field and database field retain the same attributes even if the database field changes.

When using work fields as a flag, you should assign them the prefix \$ and have the remainder of the name be descriptive. For example, a work field name such as \$GLOBL is more descriptive than a field name such as \$G.

For numeric indices, use the fields defined in the data structure I00DSINX.

### Current Date and Time

When retrieving the current date and time, use the TIME operation code instead of UDATE. UDATE obtains the date format of the system from which the program was compiled on. The date format cannot be changed without recompiling the program. TIME uses the system's date format at the time the operations code is executed.

NOTE: The TIME operation requires significant system resources. If possible, use it only once in a program. Typically, this would occur in the Housekeeping Subroutine (S999).

Always use program X0028 to edit dates and format them for output.



## Appendix C – CASE Program Types

---

We have created this guide to assist you in using the CASE Program Types provided by J.D. Edwards. Each program type is listed along with its intended use and required entries. This material gives users of this product a quick reference to all program types.

### A0010 — Interactive Subfile Inquiry

#### Program Type Description

Use this program type for the creation of an interactive subfile program. This subfile program is inquiry only. This program type processes a single master file by key. Lockout Action Codes are not used. Create a display file prior to generating this program type.

#### Display File Definition

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key field are noted by updating the Edited Field in the Field Definition screen of SDA with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key field update automatically.

The definition of Action Code is optional. Define a default cursor location if there is no action code.

#### CL Program Definition

Copy and revise model CL Program J98MODEL1 to create a CL program for use with program type A0010. You can use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Input column. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a key list for record retrieval from the master file. If you are not using the complete key list, update the Key Sequence Field in the Detailed Programming Facility to include only those data items which are needed. This key list should match your key field definition from the control format of the display file.

### **Special Considerations**

Add special logic if you want to process the master file using the key as a restrictive key. The default logic performs a SETLL which positions the records from the file using the key and then reads without a key until the subfile fills.

### **Quick Start Generation**

You can generate this program type using Quick Start.

## **A0020 — Interactive Single Record Inquiry**

### **Program Type Description**

Use this program type for the creation of an interactive single record program. This program is inquiry only. Create a display file prior to generating this program type. This program type processes a single master file by key.

### **Display File Definition**

This program type scrubs the key field in the display file prior to processing the master file. The key field is noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key field updates automatically.

The definition of Action Code is optional. Define a default cursor location if there is no action code. Lockout Action Codes are not used with this program type.



### **CL Program Definition**

Copy and revise model CL Program J98MODEL1 to create a CL program for use with program type A0020. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Input column. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Special Considerations**

This program type uses a key list for record retrieval from the master file. This key list should match your key field definition from the control format of the display file. One record displays per inquiry.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **B0010 — Interactive Single Record Maintenance**

### **Program Type Description**

Use this program type for the creation of an interactive single record maintenance program. Create a display file prior to generating this program type. This program type processes a single master file by key. User defined selection exits and function keys are optional.

### **Display File Definition**

This program type scrubs the key field in the display file prior to processing the master file. The key field is noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key field updates automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

### **CL Program Definition**

Copy and revise model CL Program J98MODEL1 to create a CL program for use with program type B0010. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Update column. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a key list for record retrieval from the master file. If you are not using the complete key list, update the Key Sequence Field in the Detailed Programming Facility to include only those data items which are needed. This key list should match your key field definition from the control format of the display file.

### **Quick Start Generation**

Generate this program type using Quick Start.

## **C0010 — Batch Report with Totals**

### **Program Type Description**

Use this program type for the creation of a batch report program that is DREAM Writer controlled. Create a printer file prior to generating this program type. This program type processes a single master file. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. Lockout Action Codes and user defined selection exits and function keys are not used.

### **Printer File Definition**

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

### **CL Program Definition**

Copy and revise model CL Program J98MODEL6 to create a CL program for use with program type C0010. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a printer file. The master file has M or 1 in the Input column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Special Considerations**

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits the entry of totaling and page breaks along with the data sequencing.

### **Quick Start Generation**

Generate this program type using Quick Start.

## **C0020 — Batch Report with Totals and Subheadings**

### **Program Type Description**

Use this program type for the creation of a batch report program that is DREAM Writer controlled. Create a printer file prior to generating this program type. This program type processes a single master file. The data passed to the print program is based on the DREAM Writer Selection and Sequencing parameters. Lockout Action Codes and user defined selection exits and function keys are not used.

### **Printer File Definition**

This program type requires that formats HEADING1, HEADING2 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

### **CL Program Definition**

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type C0020. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a printer file. The master file has M or 1 in the Input column. The printer file begins with a R and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Special Considerations**

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

If printing subheadings using format HEADING2, use the special keywords for Data Dictionary description (VC0ROW), data key field (VC0KEY), and data key description (VC0DSC).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits the entry of totaling and page breaks along with the data sequencing.

### Quick Start Generation

Generate this program type using Quick Start.

## C0025 — Batch Report with Totals and Subheadings

### Program Type Description

Use this program type for the creation of a batch report program that is DREAM Writer controlled. Create a printer file prior to generating this program type. This program type processes a single master file. The data passed to the print program is based on the DREAM Writer Selection and Sequencing parameters. Lockout Action Codes and user defined selection exits and function keys are not used.

### Printer File Definition

This program type requires that formats HEADING1, HEADING2 and DETAIL1 exist in the printer file. Format HEADING2 is the format that prints subheadings. Format TOTAL1 is optional for totals.

### CL Program Definition

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type C0025. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a single master file and a printer file. The master file has M or 1 in the Input column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

### Special Considerations

This program type is not a Q&A response in the Program Purpose and Type step. To use this program type, use the input capable field of the first Q&A question to provide this program type name.

This program type is identical to C0020 except that the subheadings headings print above the column headings. If using this program type, control the page breaks to match the subheadings.

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

If printing subheadings using format HEADING2, use the special keywords for Data Dictionary description (VC0ROW), data key field (VC0KEY), and data key description (VC0DSC).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits the entry of totaling and page breaks along with the data sequencing.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **D0010 — Interactive Subfile Maintenance with Action Code, without Selection Exits, by Relative Record Number**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records display. The secondary master file processes by relative record number and controls the database updates.

### **Display File Definition**

This program type scrubs the key field in the control format of the display file prior to processing the master file. The key field is noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key field update automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type S and a size of 9.0 using the Display All Defined Fields in Screen Design Aid.

### CL Program Definition

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0010. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a keyed master file, a secondary master file which is not keyed, and a display file. The master file has 1 in the Input column. Define a file information data structure in the fold area of the primary master file. The secondary master file has 2 in the Update column and the Keyed Y/N value in the fold area updated with N. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use a selection exit 4 to exit to the Detailed Programming Facility for the subfile field controlling the update to the database. Update the Entry Optional Y/N field to be N. This tells the generator that this field is a required entry before the database can be updated. Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

### Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## **D0020 — Interactive Subfile Maintenance without Action Code, without Selection Exits, by Relative Record Number**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records display. The secondary master file processes by relative record number and controls the database updates.

### **Display File Definition**

This program type scrubs the key field in the control format of the display file prior to processing the master file. The key field is noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key field update automatically.

Action Code is not used. Define a default cursor location.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0020. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a keyed master file, a secondary master file which is not keyed and a display file. The master file has a 1 entry under the Input column. A file information data structure is defined in the fold area of the primary master file. The secondary master file has 2 in the Update column and the Keyed Y/N value in the fold area updated with N. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Change the Entry Optional Y/N field to N. This tells the generator that this field is a required entry before the database can be updated.

Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

### Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## **D0030 — Interactive Subfile Maintenance without Action Code, without Selection Exits, by Relative Record Number with Read Next Modified Record**

### Program Type Description

Use this program type to create an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records are display. The secondary master file processes by relative record number and controls the database updates. Subfile updates are based on read next change (READC) logic.

### Display File Definition

This program type scrubs the key field in the control format of the display file prior to processing the master file. The key field is noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you



use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

Action Code is not used. Define a default cursor location.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0030. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a keyed master file, a secondary master file which is not keyed and a display file. The master file has a 1 entry under the Input column. Define a file information data structure in the fold area of the master file. The secondary master file has 2 in the Update column and the Keyed Y/N value in the fold area updated with a N. The display begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the update to the database. Update the Entry Optional Y/N field to be N. This tells the generator that this field is a required entry before the database can be updated.

Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from a Y to a N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

### Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## D0040 — Interactive Subfile Maintenance with Action Code, with Selection Exits, by Key

### Program Type Description

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes a single master file by key. User defined selection exits and function keys are optional.

### Display File Definition

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

This subfile maintenance program type lets special logic permit the deletion of individual subfile records. This logic is performed by entering a C action code, comparing the previous value with the current value and deleting the record if the current value is blank. The previous value is stored in a hidden field at the subfile record level by using the Display All Defined Fields in Screen Design Aid.

### CL Program Definition

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0040. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Update column. The display file begins

with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a selection 4 to exit to the field details for the subfile field controlling the database update. Update the Entry Optional Y/N field to be N. This tells the generator that this field is a required entry before the database can be updated.

### **Special Considerations**

This program type uses the key information in the display file for chaining to the master file. This type must also have a hidden field and an entry optional field.

### **Quick Start Generation**

Generate this program type using Quick Start.

## **D0050 — Interactive Subfile Maintenance with Two Master Files, with Action Code, with Selection Exits, by Relative Record Number**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and is updated from the fields in the control format of the display file. The secondary master file processes by relative record number and is updated from the fields in the subfile format of the display file.

### **Display File Definition**

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is required.

Lockout Action Codes are optional. This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### CL Program Definition

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0050. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a two keyed master files, a secondary master file which is not keyed and a display file. The first master file has 1 for an entry under the Update column. This file is updated from the control format of the display file. The second master file is a non-keyed file that is maintained from the subfile format of the display file. The second master file has 2 under the Update column and X under the Add column. The Keyed Y/N value in the fold area updates with N. The third master file is the logical file that the system uses for sequencing records in the subfile. This file has 3 under the Input column. Define a file information data structure in the fold area. The keyed master files have a similar key list sequence. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update function. Update the Entry Optional Y/N field to N. This tells the generator that this field is required entry before the database can be updated.

Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

### Special Considerations

This program type uses the key information in the subfile control format of the display file for retrieving one record from the first master file and multiple

records from the second master file. This type must also have a hidden field and an entry optional field.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **D0060 – Interactive Subfile Maintenance with Action Code, without Selection Exits, by Key**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes a single master file by key.

### **Display File Definition**

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

Definition of Action Code–required. Lockout Action Codes–optional.

This subfile maintenance program type permits the deletion of individual subfile records. This logic is performed by entering a C action code, comparing the previous value with the current value and deleting the record if the current value is blank. The previous value is stored in a hidden field at the subfile record level. Define this field in the display file prior to generating this program type by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0060. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single master file and a display file. The master file has an entry of M or 1 under the Update column. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use a selection 4 to exit to the field details for the SF field controlling the update to the database. Update the Entry Optional Y/N field to be N. This informs the generator that this field is required entry before the database can be updated.

### Special Considerations

This program type uses the key information in the display file for chaining to the master file. This type must also have a hidden field and an entry optional field.

### Quick Start Generation

Generate this program type using Quick Start.

## D0070 — Interactive Subfile Maintenance with Action Code, with Selection Exits, by Relative Record Number

### Program Type Description

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records display. The secondary master file processes by relative record number and controls the database updates.

### Display File Definition

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you are using the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### CL Program Definition

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0070. Use the Quick Start CL Generator for automatic creation of your CL program.

## **File Specifications**

This program type requires the definition of a keyed master file, a secondary master file which is not keyed and a display file. The master file has a 1 under the Input column. Define a file information data structure in the fold area. The secondary master file has 2 under the Update column and the Keyed Y/N value in the fold area updated with N. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

## **Detailed Programming Facility**

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Update the Entry Optional Y/N field to N. This tells the generator that this field is required entry before the database can be updated.

Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

## **Special Considerations**

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

## **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **D0080 — Interactive Subfile Maintenance without Action Code, with Selection Exits, by Relative Record Number**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records display. The secondary master file processes by relative record number and controls the database updates.

### **Display File Definition**

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

Action Code is not used. Define a default cursor location.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0080. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a keyed master file, a secondary master file which is not keyed and a display file. The master file has 1 in the Input column. Define a file information data structure in the fold area. The secondary master file has 2 in the Update column and the Keyed Y/N value in the fold area updated with N. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Update the Entry Optional Y/N field to N. This tells the generator that this field is required entry before the database can be updated.



Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the second master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, update the Key Sequence field within the Detailed Programming Facility. Listed to the right of the master file field names is the KY column which displays the sequence number for the key fields. Clear all sequence numbers that are not included in the key search as defined by the control format of the display file. Your key sequence definition in the Detailed Programming Facility should match the key fields defined in the control format.

### **Special Considerations**

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **D0090 — Interactive Subfile Maintenance with Action Code, without Selection Exits, by Relative Record Number, Balance**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and controls the sequence in which the records display. The secondary master file processes by relative record number and controls the database updates. All records are edited before the system performs any database updates.

### **Display File Definition**

This program type scrubs the key fields in the control format of the display file for positioning within the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

Action Code is not used. Define a default cursor location.

This program type processes the secondary master file by relative record number. The record number of each subfile record is stored in a hidden relative record number field. Add the field SH#RRN to the subfile format with a type of S and a size of 9.0 by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0090. Use the Quick Start CL Generator for creation of your CL program.

### **File Specifications**

This program type requires the definition of a keyed master file, a secondary master file which is not keyed and a display file. The master file has 1 in the Input column. Define a file information data structure in the fold area. The secondary master file has 2 under the Update column and the Keyed Y/N value in the fold area updated with N. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### **Detailed Programming Facility**

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Update the Entry Optional Y/N field to N. This tells the generator that this field is required entry before the database can be updated.

Because there are two master files defined to this program type, add special logic to control the roll key processing. Subroutine S001 contains logic to clear all non-key fields for each of the master files. Since the second master file has no keys, all fields clear. This causes the roll process for the keyed master file to work incorrectly after the first subfile page fills. To correct the roll key process, find the field within the secondary master file that is the key to the primary keyed master file. Use selection exit 4 to display the Detailed Programming Facility and change the Clear After field from Y to N. This prevents the key field for roll key processing from clearing.

### **Special Considerations**

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden field and an entry optional field. The update logic in this program type processes all subfile transactions prior to performing the database updates. This allows for transaction balancing or all record verification before any updates are made.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **D0100 — Interactive Subfile Maintenance with Two Master Files, with Action Code, with Selection Exits, by Key**

### **Program Type Description**

Use this program type for the creation of an interactive subfile maintenance program. Create a display file prior to generating this program type. This program type processes two master files. The primary master file is keyed and is updated from the fields in the control format of the display file. The secondary master file processes by key and is updated from the fields in the subfile format of the display file.

### **Display File Definition**

This program type scrubs the key fields in the control format of the display file prior to processing the master file. The key fields are noted by updating the Edited Field in the Field Definition screen of Screen Design Aid with the value K. If you use the Data Base Field Selection feature in Screen Design Aid, the known key fields update automatically.

The definition of Action Code is required. Lockout Action Codes are optional.

This program type requires the definition of one or more hidden fields in the subfile record. The fields in the subfile that are keys to the second master file must also have hidden fields. Add the hidden fields by using Display All Defined Fields in Screen Design Aid.

### **CL Program Definition**

Copy and revise model CL program J98MODEL1 to create a CL program for use with program type D0100. Use the Quick Start CL Generator for creation of your CL program.

### **File Specifications**

This program type requires the definition of two keyed master files and a display file. The first master file has 1 in the Update column. This file updates from the control format of the display file. The second master file has 2 in the Update column and X in the Add column. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

### Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Update the Entry Optional Y/N field to N. This tells the generator that this field is required entry before the database can be updated.

### Special Considerations

This program type uses the key information in the display file for chaining to the master file. This type must also have a hidden field and an entry optional field.

### Quick Start Generation

Generate this program type using Quick Start.

## E0010 — Interactive Window

### Program Type Description

Use this program type for the creation of an interactive window program. Create a display file prior to generating this program type. This program type processes a single master file by key.

### Display File Definition

Screen Design Aid builds the DDS for a window program when you select Fast Path Create for Window, Y. Update the predefined VTX field from Row Desc to a meaningful Skip To description. Screen Design Aid defines a key field. Delete this field and add a VD field which is the same as the key to the master file. If the key field is greater than 10 in length, you must also shorten the literal field that follows that key and proceeds the window border.

Action Code is not used.

### CL Program Definition

A CL program is not required for this model.

If you wish to create a CL program, copy and revise model CL program J98MODEL1 to create a CL program for use with program type B0010. Use the Quick Start CL Generator for automatic creation of your CL program. The program type for windows assumes three parameters. Add these to the call statement for your program.

## **File Specifications**

This program type requires the definition of a single master file and a display file. The master file has M or 1 in the Input column. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

## **Define Option and Function Key Exits**

User defined selection exits and function keys are optional. If you use this window to return values to the calling program, add #SSELC to the Function Key definitions.

## **Detailed Programming Facility**

If used, make updates to all VC0 description fields in the Detailed Programming Facility.

A key list is used for record retrieval from the master file. If you are not using the complete key list, update the Key Sequence Field in the Detailed Programming Facility to include only those data items which are needed. This key list should match your key fields definition from the control format of the display file.

Update the fields MNMNI and MNMTTL with the key and the key description fields. Subroutine S004 assumes that only two fields display per master file record. If you plan to display more than two fields, modify this subroutine.

## **Special Considerations**

This program type uses a key list for record retrieval from the master file. This key list should match your key fields definition from the control format of the display file. One record displays per inquiry.

Subroutine S004 assumes that only two fields display per master file record. If you plan to display more than two fields, modify Subroutine S004 through J.D. Edwards SEU or \*PROC. J.D. Edwards has added two entry points to this subroutine for your use.

The window key literal in the upper left hand corner of the display file is updated at run time. Modify subroutine S999 through \*PROC prior to compiling the RPG program. Assign the video screen name to the work field VC01 by using the entry point in subroutine S999.

## **Quick Start Generation**

You cannot generate this program type using Quick Start.

## X0010 — Batch Update with Report

### Program Type Description

Use this program type for the creation of a batch update program that is DREAM Writer controlled. Create a printer file prior to generating this program type. This program type processes a single master file. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters.

### Printer File Definition

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional exist for totals.

### CL Program Definition

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type X0010. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a single master file and a printer file. The master file has M or 1 in the Update column. The default in the Add column is X. Remove this default or add special logic to your program for writing to the master file. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

### Special Considerations

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits data totaling and page breaks along with the data sequencing.

This program updates the master file in subroutine S010. You may wish to add special logic to control when updates occur.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## X0020 — Batch Update

### Program Type Description

Use this program type for the creation of a batch update program that DREAM Writer controls. This program type processes two master files. The primary master file is read and used to retrieve data from the secondary master file. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. User defined selection exits and function keys are not used.

### Printer File Definition

No printer file is used with this program type.

### CL Program Definition

Copy and revise model CL program J98MODEL2 to create a CL program for use with program type X0020. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a keyed master file and a keyed secondary file. The master file has 1 in the Input column. The secondary master file has 2 in the Update column. Add files to retrieve descriptions if necessary.

### Special Considerations

This program type chains to the secondary master file in subroutine S003. Update the key field or key list prior to this chain. Key fields do not automatically update in this program type.

This program type contains subroutine S005 for all calculations. Add all special logic needed between the read of the primary master file and the update or write of the secondary master file.

This program updates or writes the master file records in subroutine S010.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## X0030 — Batch Update with Subroutine S001

### Program Type Description

Use this program type for the creation of a batch update program that DREAM Writer controls. This program type processes two master files. The primary master file is read and used to retrieve data from the secondary master file. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. User defined selection exits and function keys are not used.

### Printer File Definition

No printer file is used with this program type.

### CL Program Definition

Copy and revise model CL program J98MODEL2 to create a CL program for use with program type X0030. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a keyed master file and a keyed secondary file. The master file has 1 in the Input column. The secondary master file has 2 in the Update column. Add files to retrieve descriptions if necessary.

### Special Considerations

This program type clears the non-key fields from the primary master file between each record processed.

This program type chains to the secondary master file in subroutine S003. Update the key field or key list prior to the chain. Key fields do not automatically update in this program type.

This program type has a subroutine S005 for all calculations. Add all special logic needed between the read of the primary master file and the update or write of the secondary master file.

This program updates or writes the master file records in subroutine S010.

### Quick Start Generation

You cannot generate this program type using Quick Start.



## X0040 – Batch Update with Report

### Program Type Description

Use this program type for the creation of a batch update program that is DREAM Writer controlled. Create a printer file prior to generating this program type. The printer file should be designed to print an audit trail of each record that is updated. This program type processes two master files. The primary master file is read and the second master file is updated. The data passed to the program is based on the DREAM Writer Data Selection and Data Sequencing parameters. User defined selection and function key exits are not used.

### Printer File Definition

This program type is going to print an audit trail for each record that is written to or updates the second master file. Formats HEADING1 and DETAIL1 must exist in the printer file. Format TOTAL1 is optional, and may be used to have totals computed for the level breaks that could be defined in the DREAM Writer Data Sequencing screen.

### CL Program Definition

Copy and revise model CL program J98MODEL2 to create a CL program for use with program type X0040. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a keyed master file and a keyed secondary file. The master file has 1 in the Input column. The secondary master file has 2 in the Update column. Add files to retrieve descriptions, if necessary.

### Special Considerations

This program type chains to the secondary master file in subroutine S003. Update the key field or key list prior to this chain. Key fields do not automatically update in this program.

Subroutine S004 is used to format fields for output to the report. Add any special logic needed between the read of the primary master file and the update or write of the second master file.

Subroutine S005 is used to scrub and edit the fields for output to the second master file. Use the Detailed Programming Facility to associate fields in the primary master file with fields in the second file. Add any special logic that is needed to compute the proper value that is loaded to the output fields.

Subroutine S010 controls the printing of the report.

Subroutine S011 updates or writes the records to the second master file.

### Quick Start Generation

You can not generate this program type using Quick Start.

## Y0010 — Conversion, Two Files with Error Report

### Program Type Description

Use this program type for the creation of a batch conversion program that DREAM Writer controls. This program type processes two master files. The primary master file is read and used to retrieve data from the secondary master file. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. User defined selection exits and function keys are not used.

### Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

### CL Program Definition

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type Y0010. Use the Quick Start CL Generator for automatic creation of your CL program.

### File Specifications

This program type requires the definition of a keyed master file and a keyed secondary file. The master file has 1 in the Input column. The secondary master file had 2 in the Update column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

### Special Considerations

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

When printing the error report, format ERROR1 must contain the special fields for error message description (RRDSCA) and error message number (RREKEY).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change

permit the entry of data totaling and page breaks along with the data sequencing.

This program type chains to the secondary master file in subroutine S003. Update the key field or key list prior to the chain. Key fields do not automatically update in this program type.

This program type contains subroutine S005 for all calculations. Add all special logic needed between the read of the primary master file and the update or write of the secondary master file.

This program updates or writes the master file records in subroutine S010.

### **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **Y0020 — Conversion, One File Update with Error Report**

### **Program Type Description**

Use this program type for the creation of a batch conversion program that DREAM Writer controls. This program type processes a single master file by key. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. Lockout Action Codes are not used. User defined selection exits and function keys are not used.

### **Printer File Definition**

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

### **CL Program Definition**

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type Y0020. Use the Quick Start CL Generator for automatic creation of your CL program.

### **File Specifications**

This program type requires the definition of a single keyed master file. The master file has 1 in the Update column. This program type does not write to the master file. If it exists, remove the X from the Add column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

### Special Considerations

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

When printing the error report, format ERROR1 must contain the special fields for error message description (RRDSCA) and error message number (RREKEY).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits the entry of data totaling and page breaks along with the data sequencing.

This program type uses subroutine S005 for all calculations. Add special logic needed between the read of the primary master file and the update of the master file.

This program updates the master file records in subroutine S010.

### Quick Start Generation

You cannot generate this program type using Quick Start.

## Y0030 — Conversion, One File Write with Error Report

### Program Type Description

Use this program type for the creation of a batch conversion program that DREAM Writer controls. This program type processes a single master file by key. The data passed to the program is based on the DREAM Writer Selection and Sequencing parameters. Lockout Action Codes and user defined selection exits and function keys are not used.

### Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

### CL Program Definition

Copy and revise model CL program J98MODEL6 to create a CL program for use with program type Y0030. Use the Quick Start CL Generator for automatic creation of your CL program.

## **File Specifications**

This program type requires the definition of a single keyed master file. The master file has 1 in the Update column. This program type writes to the master file. Type X in the Add column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

## **Special Considerations**

If printing totals using format TOTAL1, use the special keywords for Data Dictionary description (VC1ROW), data key field (VC1KEY), and data key description (VC1DSC).

When printing the error report, format ERROR1 must contain the special fields for error message description (RRDSCA) and error message number (RREKEY).

When creating your DREAM Writer Version, change the Type Report Totaling field to 2. This field is found on the Additional Parameters screen. This change permits the entry of data totaling and page breaks along with the data sequencing.

This program type uses subroutine S005 for all calculations. Add special logic needed between the read of the primary master file and the update of the master file.

This program writes the master file records in subroutine S010.

## **Quick Start Generation**

You cannot generate this program type using Quick Start.

## **Guidelines**

- In all J.D. Edwards programs, general help instructions are optional but highly recommended. Fold areas and AAI's are also optional within program types.
- Processing Options are optional. Define processing options for batch processing. The program generation step which automatically includes the logic for retrieval of this information is subroutine S999. Define the special calculations for use of these options.
- User defined selection exits and function keys are optional for all program types.
- In the Detailed Programming Facility, you can make optional calculations using Program Design Language. You can also make updates to all VC0 description fields.



## Appendix D – Source Listings

---

The following sources are listed in this appendix:

- I00DSPROG — Program Status Data Structure
- I00SC — Copy Module – Retrieve Soft Coding
- P928011 — Item Master Information

## Data Structure — I00DSPROG

98330  
I00DSPROG .JDFSRC61  
Seq No.

J.D. Edwards & Company  
Print Source Code

Date - 27.01.94

Seq No.		Mod Date
1.00	I*****	00003 08.02.85
2.00	I*	08.02.85
3.00	I* PROGRAM STATUS DATA STRUCTURE	08.02.85
4.00	I* -----	08.02.85
5.00	I*	08.02.85
6.00	I* Portions of this data structure are loaded at the time the	08.02.85
7.00	I* program is loaded. Other portions of this data structure	08.02.85
8.00	I* are loaded as you perform I/O.	08.02.85
9.00	I*	08.02.85
10.00	I* PURPOSE	08.02.85
11.00	I* -----	08.02.85
12.00	I* This common subroutine is set up to be used with C0000	08.02.85
13.00	I* (Business Unit Security) common subroutine and C0001(Edit	08.02.85
14.00	I* Action Code) common subroutine. Those two subroutines	08.02.85
15.00	I* will retrieve ##USER for the user name.	08.02.85
16.00	I*	08.02.85
17.00	I* No program calcs are done in this subroutine.	08.02.85
18.00	I*	08.02.85
19.00	I##PSDS SDS	18.12.89
20.00	I*	08.02.85
21.00	I* Program Name	08.02.85
22.00	I 1 10 ##PROG	08.02.85
23.00	I* Status Code(09999=I/O Error)	08.02.85
24.00	I 11 150##STAT	08.02.85
25.00	I* Previous Status code	08.02.85
26.00	I 16 200##PSTA	08.02.85
27.00	I* RPG Source Statement Sequence Number	08.02.85
28.00	I 21 28 ##SEQN	08.02.85
29.00	I* RPG Routine in Which Exception/Error Occured	08.02.85
30.00	I 29 36 ##ROUT	08.02.85
31.00	I* Number of Parameters Passed to This Program	08.02.85
32.00	I 37 390##PARM	08.02.85
33.00	I* Exception Type(MCH=Machine, CPF=CPF)	08.02.85
34.00	I 40 42 ##ETYP	08.02.85
35.00	I* Exception Message Number	08.02.85
36.00	I 43 46 ##ENBR	08.02.85
37.00	I* Machine Instruction/Object Definition Template Number	08.02.85
38.00	I 47 50 ##MINO	08.02.85
39.00	I* Work Area for Messages	08.02.85
40.00	I 51 80 ##MWRK	08.02.85
41.00	I* Name of Library in Which Program is Located	08.02.85
42.00	I 81 90 ##PLIB	08.02.85
43.00	I* Retrieved Exdeption Data. CPF Messages.	08.02.85
44.00	I 91 170 ##MSG	08.02.85
45.00	I* Identification of Exception That Caused RPG9001	08.02.85
46.00	I 171 174 ##9001	08.02.85
47.00	I* Unused	08.02.85
48.00	I 175 200 ##FLR1	08.02.85
49.00	I* Name of File for Last I/O(Only Updated if Error)	08.02.85
50.00	I 201 208 ##LFIL	08.02.85
51.00	I* Status Info on Last File Used(Only on Error)	08.02.85
52.00	I 209 243 ##LFST	08.02.85
53.00	I* Status Code on Last File Used(Only on Error)	15.12.89
54.00	I 209 213 ##LFS5	15.12.89
55.00	I* Job Name	08.02.85
56.00	I 244 253 ##JOBN	08.02.85



98330		J.D. Edwards & Company	
I00DSPROG .JDFSRC61		Print Source Code	Date - 27.01.94
Seq No.			Mod Date
57.00	I*	User Name From User Profile	08.02.85
58.00	I	254 263 ##USER	08.02.85
59.00	I*	Job Number	08.02.85
60.00	I	264 2690##JOB#	08.02.85
61.00	I*	Date Job Entered the System(MMDDYY)	08.02.85
62.00	I	270 2750##JDT	08.02.85
63.00	I*	Date of Program Execution(MMDDYY)	08.02.85
64.00	I	276 2810##EDT	08.02.85
65.00	I*	Time of Program Exection(HHMMSS)	08.02.85
66.00	I	282 2870##ETM	08.02.85
67.00	I*	Date Program Was Compiled	08.02.85
68.00	I	288 2930##CDT	08.02.85
69.00	I*	Time Program Was Compiled	08.02.85
70.00	I	294 2990##CTM	08.02.85
71.00	I*	Level of the Compiler	08.02.85
72.00	I	300 303 ##LVL	08.02.85
73.00	I*	Source File Name	08.02.85
74.00	I	304 313 ##SRCN	08.02.85
75.00	I*	Source Library Name	08.02.85
76.00	I	314 323 ##SRCL	08.02.85
77.00	I*	Source File Member Name	08.02.85
78.00	I	324 333 ##SRCM	08.02.85
79.00	I*	Unused	08.02.85
80.00	I	334 429 ##FLR2	09.06.87

## Data Structure — I00SC

98330  
I00SC

.JDFSRC61

J.D. Edwards & Company  
Print Source Code

Date - 27.01.94

Seq No.

Mod Date

1.00	I*****		12.02.88
2.00	I* This is part of a composite common subroutine. In		12.02.88
3.00	I* order for the subroutine to work correctly, the		12.02.88
4.00	I* RPG program must /COPY in the following members:		12.02.88
5.00	I* I00SC, C00SC		12.02.88
6.00	I*		25.04.88
7.00	I* NOTE: The "SRVFDS" file information data structure must		25.04.88
8.00	I* be specified in a continuation record for the display		25.04.88
9.00	I* file (File Description Specification "KINFDS").		25.04.88
10.00	I*		25.04.88
11.00	I*****		12.02.88
12.00	I* PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES		12.02.88
13.00	I* -----		12.02.88
14.00	I*		12.02.88
15.00	II00SC DS		07.01.91
16.00	I*		12.02.88
17.00	I* Function keys 1 thru 32.		17.02.88
18.00	I*		12.02.88
19.00	I 1 32 I00SCF		17.02.88
20.00	I*		25.04.88
21.00	I* Function - End of Job		25.04.88
22.00	I 1 1 #FEOJ		17.02.88
23.00	I*		25.04.88
24.00	I* Function - Clear Screen		25.04.88
25.00	I 2 2 #FCLR		17.02.88
26.00	I*		25.04.88
27.00	I* Function - Help		25.04.88
28.00	I 3 3 #FHELP		17.02.88
29.00	I*		25.04.88
30.00	I* Function - Values List Display		25.04.88
31.00	I 4 4 #FVLST		17.02.88
32.00	I*		25.04.88
33.00	I* Function - Roll Up		25.04.88
34.00	I 5 5 #FROLU		17.02.88
35.00	I*		25.04.88
36.00	I* Function - Roll Down		25.04.88
37.00	I 6 6 #FROLD		17.02.88
38.00	I*		25.04.88
39.00	I* Function - Window Screen Left		25.04.88
40.00	I 7 7 #FWLFT		17.02.88
41.00	I*		25.04.88
42.00	I* Function - Window Screen Right		25.04.88
43.00	I 8 8 #FWRGT		17.02.88
44.00	I*		25.04.88
45.00	I* Function - Question Mark/Cursor Sensitive Help		25.04.88
46.00	I 9 9 #FQMRK		17.02.88
47.00	I*		25.04.88
48.00	I* Function - Display Error Message(s)		25.04.88
49.00	I 10 10 #FERRD		17.02.88
50.00	I*		25.04.88
51.00	I* Function - Exit to Address Book		25.04.88
52.00	I 11 11 #FAB		17.02.88
53.00	I*		25.04.88
54.00	I* Function - Exit to Name Search		25.04.88
55.00	I 12 12 #FNS		17.02.88
56.00	I*		25.04.88

98330		J.D. Edwards & Company	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.94

Seq No.			Mod Date
57.00	I*	Function - Return to Previous Panel/Menu	25.04.88
58.00	I	13 13 #FPRV	17.02.88
59.00	I*		25.04.88
60.00	I*	Function - Display Alternate Panel	25.04.88
61.00	I	14 14 #FALT	17.02.88
62.00	I*		25.04.88
63.00	I*	Function - Exit to Display Valid Function Keys	19.09.89
64.00	I	15 15 #FKEYS	19.09.89
65.00	I*		25.04.88
66.00	I*	Function - Return to Primary Menu	25.04.88
67.00	I	16 16 #FMM	17.02.88
68.00	I*		25.04.88
69.00	I*	Function - Hard Copy Print	25.04.88
70.00	I	17 17 #FPRT	21.04.88
71.00	I*		25.04.88
72.00	I*	Function - Variable by Program (1 thru 15)	25.04.88
73.00	I	18 18 #F01	21.04.88
74.00	I	19 19 #F02	21.04.88
75.00	I	20 20 #F03	21.04.88
76.00	I	21 21 #F04	21.04.88
77.00	I	22 22 #F05	21.04.88
78.00	I	23 23 #F06	21.04.88
79.00	I	24 24 #F07	21.04.88
80.00	I	25 25 #F08	21.04.88
81.00	I	26 26 #F09	21.04.88
82.00	I	27 27 #F10	21.04.88
83.00	I	28 28 #F11	21.04.88
84.00	I	29 29 #F12	21.04.88
85.00	I	30 30 #F13	21.04.88
86.00	I	31 31 #F14	21.04.88
87.00	I	32 32 #F15	21.04.88
88.00	I*		17.02.88
89.00	I*	Selections 1 thru 24.	17.02.88
90.00	I*		17.02.88
91.00	I	33 80 I00SCS	17.02.88
92.00	I*		25.04.88
93.00	I*	Selection - Select/Work With	25.04.88
94.00	I	33 340#SSELC	07.06.88
95.00	I*		25.04.88
96.00	I*	Selection - Change/Revise	25.04.88
97.00	I	35 360#SCHNG	07.06.88
98.00	I*		25.04.88
99.00	I*	Selection - Copy/Hold	25.04.88
100.00	I	37 380#SCOPY	07.06.88
101.00	I*		25.04.88
102.00	I*	Selection - Delete/Cancel	25.04.88
103.00	I	39 400#SDELT	07.06.88
104.00	I*		25.04.88
105.00	I*	Selection - Display/View	25.04.88
106.00	I	41 420#SDSPL	07.06.88
107.00	I*		25.04.88
108.00	I*	Selection - Print/Release	25.04.88
109.00	I	43 440#SPRNT	07.06.88
110.00	I*		25.04.88
111.00	I*	Selection - Rename	25.04.88
112.00	I	45 460#SRENM	07.06.88

# CASE – Computer Aided Software Engineering

98330  
I00SC

.JDFSRC61

J.D. Edwards & Company  
Print Source Code

Date - 27.01.94

Seq No.

Mod Date

113.00	I*			25.04.88
114.00	I*	Selection - Display Attributes		25.04.88
115.00	I		47 480#SDATR	07.06.88
116.00	I*			25.04.88
117.00	I*	Selection - Variable by Program (1 thru 16)		25.04.88
118.00	I		49 500#S01	07.06.88
119.00	I		51 520#S02	07.06.88
120.00	I		53 540#S03	07.06.88
121.00	I		55 560#S04	07.06.88
122.00	I		57 580#S05	07.06.88
123.00	I		59 600#S06	07.06.88
124.00	I		61 620#S07	07.06.88
125.00	I		63 640#S08	07.06.88
126.00	I		65 660#S09	07.06.88
127.00	I		67 680#S10	07.06.88
128.00	I		69 700#S11	07.06.88
129.00	I		71 720#S12	07.06.88
130.00	I		73 740#S13	07.06.88
131.00	I		75 760#S14	07.06.88
132.00	I		77 780#S15	07.06.88
133.00	I		79 800#S16	07.06.88
134.00	I*			22.02.88
135.00	I*	Global JDE Variables		07.01.91
136.00	I*			22.02.88
137.00	I		81 120 I00SCG	07.01.91
138.00	I*	Future use space, room to grow		25.02.91
139.00	I*	-----		07.01.91
140.00	I*			07.01.91
141.00	I*	File Information Data Structure for Panel/Report file.		07.01.91
142.00	I*			07.01.91
143.00	ISRVFDS	DS		22.02.88
144.00	I*			22.02.88
145.00	I*	Internal program file name		22.02.88
146.00	I		1 8 @@IFIL	22.02.88
147.00	I*			22.02.88
148.00	I*	Open indication (1=OPEN)		22.02.88
149.00	I		9 9 @@OPEN	22.02.88
150.00	I*			22.02.88
151.00	I*	End Of File indication (1=End of file)		22.02.88
152.00	I		10 10 @@EOF	22.02.88
153.00	I*			22.02.88
154.00	I*	Status code (09999=I/O Error)		22.02.88
155.00	I		11 150@@STAT	22.02.88
156.00	I*			22.02.88
157.00	I*	Operation code		22.02.88
158.00	I		16 21 @@OPCD	22.02.88
159.00	I*			22.02.88
160.00	I*	Name of RPG routine exception/error occurred		22.02.88
161.00	I		22 29 @@ROUT	22.02.88
162.00	I*			22.02.88
163.00	I*	RPG source statement sequence number		22.02.88
164.00	I		30 37 @@SEQN	22.02.88
165.00	I*			22.02.88
166.00	I*	User-Specified reason for error on *SPECIAL file		22.02.88
167.00	I		38 420@@RESN	22.02.88
168.00	I*			22.02.88

98330		J.D. Edwards & Company	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.94

Seq No.			Mod Date
169.00	I*	Recore format being processed (External file)	22.02.88
170.00	I*	Record ID (Left justified for internal file)	22.02.88
171.00	I	38 45 @@FRMT	22.02.88
172.00	I*		22.02.88
173.00	I*	Machine OR CPF message number	22.02.88
174.00	I	46 52 @@EXNO	22.02.88
175.00	I*		22.02.88
176.00	I*	Machine instruction/Object definition template number	22.02.88
177.00	I	53 56 @@MI	22.02.88
178.00	I*		22.02.88
179.00	I*	UNUSED	22.02.88
180.00	I	57 80 @@FLR1	22.02.88
181.00	I*		22.02.88
182.00	I*	Open data path type (DS-Device DB-Data Base SP-Spool)	22.02.88
183.00	I	81 82 @@ODP	22.02.88
184.00	I*		22.02.88
185.00	I*	Name of file actually opened	22.02.88
186.00	I	83 92 @@FILE	22.02.88
187.00	I*		22.02.88
188.00	I*	Name of library containing file (Blank if spool file)	22.02.88
189.00	I	93 102 @@LIBR	22.02.88
190.00	I*		22.02.88
191.00	I*	Name of spooled file (set only on spool files)	22.02.88
192.00	I	103 112 @@SPNM	22.02.88
193.00	I*		22.02.88
194.00	I*	Name of library where spooled file is located	22.02.88
195.00	I	113 122 @@SPLB	22.02.88
196.00	I*		22.02.88
197.00	I*	Spooled file number (set only on spool files)	22.02.88
198.00	I	B 123 1240@@SPNO	22.02.88
199.00	I*		22.02.88
200.00	I*	Primary record length (bytes transferred at a time)	22.02.88
201.00	I	B 125 1260@@PRCL	22.02.88
202.00	I*		22.02.88
203.00	I*	Secondary record length (bytes transferred at a time)	22.02.88
204.00	I	B 127 1280@@SRCL	22.02.88
205.00	I*		22.02.88
206.00	I*	Member Name:	22.02.88
207.00	I*	. If ODP type is DB, this entry is the	22.02.88
208.00	I*	member name in file named in position	22.02.88
209.00	I*	83 through 92.	22.02.88
210.00	I*	. If ODP type is SP, this entry is the	22.02.88
211.00	I*	member name in the file named in	22.02.88
212.00	I*	positions 103 through 112.	22.02.88
213.00	I*		22.02.88
214.00	I	129 138 @@MBR	22.02.88
215.00	I*		22.02.88
216.00	I*	Input buffer length (zero if no buffer allocated)	22.02.88
217.00	I	B 139 1420@@IBLN	22.02.88
218.00	I*		22.02.88
219.00	I*	Output buffer length (zero if no buffer allocated)	22.02.88
220.00	I	B 143 1460@@OBLN	22.02.88
221.00	I*		22.02.88
222.00	I*	Device Class (supplied only if ODP type is DS or SP)	22.02.88
223.00	I*	1 = Display	22.02.88
224.00	I*	2 = Printer	22.02.88

# CASE – Computer Aided Software Engineering

98330  
I00SC .JDFSRC61

J.D. Edwards & Company  
Print Source Code

Date - 27.01.94

Seq No.			Mod Date
225.00	I*	3 = Card	22.02.88
226.00	I*	4 = Diskette	22.02.88
227.00	I*	5 = Tape	22.02.88
228.00	I		22.02.88
229.00	I*	B 147 1480@@DVCL	22.02.88
230.00	I*	Diskette location(value from 1 to 23 = slot location)	22.02.88
231.00	I	149 151 @@DKLC	22.02.88
232.00	I*		22.02.88
233.00	I*	Number of rows on display screen or lines on a page	22.02.88
234.00	I	B 152 1530@@VDRW	22.02.88
235.00	I*		22.02.88
236.00	I*	Number of columns on display screen or printed line	22.02.88
237.00	I	B 154 1550@@VDCM	22.02.88
238.00	I*		22.02.88
239.00	I*	Number of records in file at time of open	22.02.88
240.00	I	B 156 1590@@RCNT	22.02.88
241.00	I*		22.02.88
242.00	I*	Access type (only supplied if ODP type is DB)	22.02.88
243.00	I*	KU = Keyed, Unique	22.02.88
244.00	I*	KF = Keyed, FIFO W/Duplicate keys	22.02.88
245.00	I*	KI = Keyed, LIFO W/Duplicate keys	22.02.88
246.00	I*	AR = Arrival sequence	22.02.88
247.00	I	160 161 @@ACTY	22.02.88
248.00	I*		22.02.88
249.00	I*	Duplicate key indication (D=Allowed U=Not allowed)	22.02.88
250.00	I	162 162 @@DUK	22.02.88
251.00	I*		22.02.88
252.00	I*	Source file indication (Y=Source file)	22.02.88
253.00	I	163 163 @@SRCI	22.02.88
254.00	I*		22.02.88
255.00	I*	User file control block parameters in effect	22.02.88
256.00	I	164 173 @@FCBP	22.02.88
257.00	I*		22.02.88
258.00	I*	User file control block overrides in effect	22.02.88
259.00	I	174 183 @@FCBO	22.02.88
260.00	I*		22.02.88
261.00	I*	Offset to volume label fields of open feedback	22.02.88
262.00	I*	(Supplied only for tape or diskette)	22.02.88
263.00	I	B 184 1850@@OVL	22.02.88
264.00	I*		22.02.88
265.00	I*	Number of records to be transferred on file open	22.02.88
266.00	I	B 186 1870@@RTFO	22.02.88
267.00	I*		22.02.88
268.00	I*	Overflow line number (printer files only)	22.02.88
269.00	I	B 188 1890@@OFLN	22.02.88
270.00	I*		22.02.88
271.00	I*	UNUSED	22.02.88
272.00	I	190 240 @@FLR2	22.02.88
273.00	I*		22.02.88
274.00	I*	Offset to device dependent feedback information	22.02.88
275.00	I*	(See Appendix D of the CPF Programmer's Guide for	22.02.88
276.00	I*	layout of feedback information for specific	22.02.88
277.00	I*	devices)	22.02.88
278.00	I	B 241 2420@@ODFB	22.02.88
279.00	I*		22.02.88
280.00	I*	Put operation count	22.02.88

98330		J.D. Edwards & Company	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.94

Seq No.			Mod Date
281.00	I	B 243 2460@@PUTC	22.02.88
282.00	I*		22.02.88
283.00	I*	Get operation count	22.02.88
284.00	I	B 247 2500@@GETC	22.02.88
285.00	I*		22.02.88
286.00	I*	PutGet operation count	22.02.88
287.00	I	B 251 2540@@PGC	22.02.88
288.00	I*		22.02.88
289.00	I*	Non-I/O operation count (update of subfile records)	22.02.88
290.00	I	B 255 2580@@NIOC	22.02.88
291.00	I*		22.02.88
292.00	I*	Current operation (Last operation requested)	22.02.88
293.00	I*	X'01' = Get	22.02.88
294.00	I*	X'02' = Get W/Subfile record number	22.02.88
295.00	I*	X'03' = Get by key	22.02.88
296.00	I*	X'05' = Put	22.02.88
297.00	I*	X'06' = PutGet	22.02.88
298.00	I*	X'07' = Update	22.02.88
299.00	I*	X'08' = Delete	22.02.88
300.00	I*	X'09' = Force End of Data	22.02.88
301.00	I*	X'0D' = Release	22.02.88
302.00	I	259 260 @@COPR	22.02.88
303.00	I*		22.02.88
304.00	I*	Name of record format just processed:	22.02.88
305.00	I*	. Specified on the I/O request, or	22.02.88
306.00	I*	. Determined by default processing	22.02.88
307.00	I	261 270 @@CFMT	22.02.88
308.00	I*		22.02.88
309.00	I*	Device Class	22.02.88
310.00	I*	Position 271	22.02.88
311.00	I*	X'00' = Data Base	22.02.88
312.00	I*	X'01' = Keyboard display	22.02.88
313.00	I*	X'02' = Printer	22.02.88
314.00	I*	X'03' = Card	22.02.88
315.00	I*	X'04' = Diskette	22.02.88
316.00	I*	X'05' = Tape	22.02.88
317.00	I*	Position 272 (If position 271 contains X'00')	22.02.88
318.00	I*	X'00' = Nonkeyed file	22.02.88
319.00	I*	X'01' = Keyed file	22.02.88
320.00	I*	Position 272 (If position 271 not X'00')	22.02.88
321.00	I*	X'00' = 5250 Display station, 960 characters	22.02.88
322.00	I*	X'01' = System console, 1024 characters	22.02.88
323.00	I*	X'02' = 5256 Printer	22.02.88
324.00	I*	X'03' = 5211/3262 Printer	22.02.88
325.00	I*	X'04' = MFPU	22.02.88
326.00	I*	X'05' = 3411/3410 Tape	22.02.88
327.00	I*	X'06' = 72M Diskette	22.02.88
328.00	I*	X'07' = 5250 Display station, 1920 characters	22.02.88
329.00	I*	X'08' = Spooled	22.02.88
330.00	I	271 272 @@DCLS	22.02.88
331.00	I*		22.02.88
332.00	I*	Device name (Last completed operation)	22.02.88
333.00	I	273 282 @@DNAM	22.02.88
334.00	I*		22.02.88
335.00	I*	Length of last I/O record processed	22.02.88
336.00	I	B 283 2860@@LIOL	22.02.88

# CASE – Computer Aided Software Engineering

98330  
I00SC .JDFSRC61

J.D. Edwards & Company  
Print Source Code

Date - 27.01.94

Seq No.		Mod Date
337.00	I*	22.02.88
338.00	I*	22.02.88
339.00	I	22.02.88
340.00	I*	22.02.88
341.00	I*	22.02.88
342.00	I	22.02.88
343.00	I*	22.02.88
344.00	I*	22.02.88
345.00	I*	22.02.88
346.00	I*	22.02.88
347.00	I*	22.02.88
348.00	I*	22.02.88
349.00	I*	22.02.88
350.00	I*	22.02.88
351.00	I*	22.02.88
352.00	I*	22.02.88
353.00	I*	09.08.91
354.00	I*	22.02.88
355.00	I*	22.02.88
356.00	I*	22.02.88
357.00	I*	22.02.88
358.00	I*	22.02.88
359.00	I*	22.02.88
360.00	I*	22.02.88
361.00	I*	22.02.88
362.00	I*	22.02.88
363.00	I*	22.02.88
364.00	I*	22.02.88
365.00	I*	22.02.88
366.00	I*	22.02.88
367.00	I*	22.02.88
368.00	I*	22.02.88
369.00	I*	22.02.88
370.00	I*	22.02.88
371.00	I*	22.02.88
372.00	I*	22.02.88
373.00	I*	22.02.88
374.00	I*	22.02.88
375.00	I*	22.02.88
376.00	I*	22.02.88
377.00	I	22.02.88
378.00	I*	22.02.88
379.00	I*	22.02.88
380.00	I	22.02.88
381.00	I*	22.02.88
382.00	I*	22.02.88
383.00	I	22.02.88
384.00	I*	22.02.88
385.00	I*	22.02.88
386.00	I*	22.02.88
387.00	I*	22.02.88
388.00	I*	22.02.88
389.00	I*	22.02.88
390.00	I*	22.02.88
391.00	I*	22.02.88
392.00	I*	22.02.88

Routing data information 287 366 @@RDTA

Current line number within a printer page B 367 3680@@CLNO

AID character indication:

X'F1' = Enter/Rec Adv

X'F5' = Roll up

X'F4' = Roll down

X'F6' = Print

X'F8' = Home

X'BD' = Clear

X'F3' = Help

X'3F' = Auto Enter

X'31' = Command Key 01

X'32' = Command Key 02

X'33' = Command Key 03

X'34' = Command Key 04

X'35' = Command Key 05

X'36' = Command Key 06

X'37' = Command Key 07

X'38' = Command Key 08

X'39' = Command Key 09

X'3A' = Command Key 10

X'3B' = Command Key 11

X'3C' = Command Key 12

X'B1' = Command Key 13

X'B2' = Command Key 14

X'B3' = Command Key 15

X'B4' = Command Key 16

X'B5' = Command Key 17

X'B6' = Command Key 18

X'B7' = Command Key 19

X'B8' = Command Key 20

X'B9' = Command Key 21

X'BA' = Command Key 22

X'BB' = Command Key 23

X'BC' = Command Key 24

369 369 @@AID

Cursor line in hex (display files only) 370 370 @@CURL

Cursor position in hex (display files only) 371 371 @@CURP

Note: By simply defining a 2 byte binary field and moving the cursor line/position field into it right justified you will have the numerical value of the line/position field. Remember the binary field must be set to zero prior to the move.

Number of records transmitted



98330		J.D. Edwards & Company	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.94

Seq No.			Mod Date
393.00	I	B 371 3720@@RTRM	22.02.88
394.00	I*		22.02.88
395.00	I*	UNUSED	22.02.88
396.00	I	373 375 @@FLR4	22.02.88
397.00	I*		22.02.88
398.00	I*	RRN of last subfile record written/updated	22.02.88
399.00	I	B 376 3770@@SRRN	22.02.88
400.00	I*		22.02.88
401.00	I*	RRN of first subfile record on display	22.02.88
402.00	I	B 378 3790@@SRCN	22.02.88
403.00	I*		22.02.88
404.00	I*	UNUSED	22.02.88
405.00	I	380 396 @@FLR5	02.10.89
406.00	I*		22.02.88
407.00	I*	RRN of data base record	22.02.88
408.00	I	B 397 4000@@RRN	22.02.88
409.00	I*		22.02.88
410.00	I*	Data base file key	22.02.88
411.00	I	401 528 @@RKEY	22.02.88
412.00	I*	-----	22.02.88
413.00	I*		30.08.89
414.00	I*	Cursor Sensitive Help Values	30.08.89
415.00	I*		30.08.89
416.00	II00CSR	DS	30.08.89
417.00	I*		30.08.89
418.00	I*	Returned field name.	30.08.89
419.00	I	1 10 ##FLDN	30.08.89
420.00	I*	Returned value.	30.08.89
421.00	I	11 40 ##RVAL	29.09.89
422.00	I*	Returned description.	30.08.89
423.00	I	41 70 ##RDSC	29.09.89
424.00	I*	Returned location: Row.	31.08.89
425.00	I	71 730##RROW	29.09.89
426.00	I*	Returned location: Column.	31.08.89
427.00	I	74 760##RCOL	29.09.89
428.00	I*	Dictionary Field Name (non-blank=override)	03.11.89
429.00	I	77 86 ##DTAI	29.09.89
430.00	I*	Returned Display File Format	29.09.89
431.00	I	87 96 ##RFMT	29.09.89
432.00	I*	RPG Indicator Array	29.09.89
433.00	I	97 195 ##IN	29.09.89
434.00	I*	Override Reporting System (Jargon)	06.10.92
435.00	I	196 199 ###SYR	06.10.92
436.00	I*	-----	30.08.89
437.00	I*		27.11.89
438.00	I*	Hidden Fields for Subfile Attribute Indicators	27.11.89
439.00	I*		27.11.89
440.00	ISHIN	DS	27.11.89
441.00	I	1 1 SHIN01	27.11.89
442.00	I	2 2 SHIN02	27.11.89
443.00	I	3 3 SHIN03	27.11.89
444.00	I	4 4 SHIN04	27.11.89
445.00	I	5 5 SHIN05	27.11.89
446.00	I	6 6 SHIN06	27.11.89
447.00	I	7 7 SHIN07	27.11.89
448.00	I	8 8 SHIN08	27.11.89

# CASE – Computer Aided Software Engineering

---

98330

I00SC

.JDFSRC61

J.D. Edwards & Company

Print Source Code

Date - 27.01.94

Seq No.

Mod Date

449.00	I	9	9	SHIN09	27.11.89
450.00	I	10	10	SHIN10	27.11.89
451.00	I	11	11	SHIN11	27.11.89
452.00	I	12	12	SHIN12	27.11.89
453.00	I	13	13	SHIN13	27.11.89
454.00	I	14	14	SHIN14	27.11.89
455.00	I	15	15	SHIN15	27.11.89
456.00	I	16	16	SHIN16	27.11.89
457.00	I	17	17	SHIN17	27.11.89
458.00	I	18	18	SHIN18	27.11.89
459.00	I	19	19	SHIN19	27.11.89
460.00	I	20	20	SHIN20	27.11.89
461.00	I	21	21	SHIN21	27.11.89
462.00	I	22	22	SHIN22	27.11.89
463.00	I	23	23	SHIN23	27.11.89
464.00	I	24	24	SHIN24	27.11.89
465.00	I	25	25	SHIN25	27.11.89
466.00	I	26	26	SHIN26	27.11.89
467.00	I	27	27	SHIN27	27.11.89
468.00	I	28	28	SHIN28	27.11.89
469.00	I	29	29	SHIN29	27.11.89
470.00	I	30	30	SHIN30	27.11.89
471.00	I	31	31	SHIN31	27.11.89
472.00	I	32	32	SHIN32	27.11.89
473.00	I	33	33	SHIN33	27.11.89
474.00	I	34	34	SHIN34	27.11.89
475.00	I	35	35	SHIN35	27.11.89
476.00	I	36	36	SHIN36	27.11.89
477.00	I	37	37	SHIN37	27.11.89
478.00	I	38	38	SHIN38	27.11.89
479.00	I	39	39	SHIN39	27.11.89
480.00	I	40	40	SHIN40	30.11.89
481.00	I	41	41	SHIN41	27.11.89
482.00	I	42	42	SHIN42	27.11.89
483.00	I	43	43	SHIN43	27.11.89
484.00	I	44	44	SHIN44	27.11.89
485.00	I	45	45	SHIN45	27.11.89
486.00	I	46	46	SHIN46	27.11.89
487.00	I	47	47	SHIN47	27.11.89
488.00	I	48	48	SHIN48	27.11.89
489.00	I	49	49	SHIN49	27.11.89
490.00	I	50	50	SHIN50	27.11.89
491.00	I	51	51	SHIN51	27.11.89
492.00	I	52	52	SHIN52	27.11.89
493.00	I	53	53	SHIN53	27.11.89
494.00	I	54	54	SHIN54	27.11.89
495.00	I	55	55	SHIN55	27.11.89
496.00	I	56	56	SHIN56	27.11.89
497.00	I	57	57	SHIN57	27.11.89
498.00	I	58	58	SHIN58	27.11.89
499.00	I	59	59	SHIN59	27.11.89
500.00	I	60	60	SHIN60	27.11.89
501.00	I	61	61	SHIN61	27.11.89
502.00	I	62	62	SHIN62	27.11.89
503.00	I	63	63	SHIN63	27.11.89
504.00	I	64	64	SHIN64	27.11.89

98330		J.D. Edwards & Company	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.94

Seq No.			Mod Date
505.00	I	65 65 SHIN65	27.11.89
506.00	I	66 66 SHIN66	27.11.89
507.00	I	67 67 SHIN67	27.11.89
508.00	I	68 68 SHIN68	27.11.89
509.00	I	69 69 SHIN69	27.11.89
510.00	I	70 70 SHIN70	27.11.89
511.00	I	71 71 SHIN71	27.11.89
512.00	I	72 72 SHIN72	27.11.89
513.00	I	73 73 SHIN73	27.11.89
514.00	I	74 74 SHIN74	27.11.89
515.00	I	75 75 SHIN75	27.11.89
516.00	I	76 76 SHIN76	27.11.89
517.00	I	77 77 SHIN77	27.11.89
518.00	I	78 78 SHIN78	27.11.89
519.00	I	79 79 SHIN79	27.11.89
520.00	I	80 80 SHIN80	27.11.89
521.00	I	81 81 SHIN81	27.11.89
522.00	I	82 82 SHIN82	27.11.89
523.00	I	83 83 SHIN83	27.11.89
524.00	I	84 84 SHIN84	27.11.89
525.00	I	85 85 SHIN85	27.11.89
526.00	I	86 86 SHIN86	27.11.89
527.00	I	87 87 SHIN87	27.11.89
528.00	I	88 88 SHIN88	27.11.89
529.00	I	89 89 SHIN89	27.11.89
530.00	I	90 90 SHIN90	27.11.89
531.00	I	91 91 SHIN91	27.11.89
532.00	I	92 92 SHIN92	27.11.89
533.00	I	93 93 SHIN93	27.11.89
534.00	I	94 94 SHIN94	27.11.89
535.00	I	95 95 SHIN95	27.11.89
536.00	I	96 96 SHIN96	27.11.89
537.00	I	97 97 SHIN97	27.11.89
538.00	I	98 98 SHIN98	27.11.89
539.00	I	99 99 SHIN99	27.11.89
540.00	I*	-----	27.11.89
541.00	I*		09.06.93
542.00	I*	Hidden Fields for Subfile Mode and Cursor Position	09.06.93
543.00	I*		09.06.93
544.00	II00MDE	DS	09.06.93
545.00	I*	Subfile Mode	09.06.93
546.00	I	1 1 ###MD	09.06.93
547.00	I*	Subfile Relative Record Number	09.06.93
548.00	I	2 60###RNO	09.06.93
549.00	I*	Cursor Location - Record Format	09.06.93
550.00	I	7 16 ###CRC	09.06.93
551.00	I*	Cursor Location - Field Name	09.06.93
552.00	I	17 26 ###CFL	09.06.93
553.00	I*	-----	09.06.93

```

1.00      H/TITLE P928011-Item Master Information
2.00      H*-----
3.00      H*
4.00      H*      Copyright (c) 1993
5.00      H*      J. D. Edwards & Company
6.00      H*
7.00      H*
8.00      H*
9.00      H*
10.00     H*
11.00     H*
12.00     H*
13.00     H*
14.00     H*
15.00     H*-----
16.00     F*
17.00     F*      PROGRAM REVISION LOG
18.00     F*      -----
19.00     F*
20.00     F*
21.00     F*
22.00AUTHRF*
23.00     F*
24.00     F*      B0010      - Standard Maintenance Program Type
25.00     F*      This program provides the standard single cycle
26.00     F*      processing for adding, changing, deleting and
27.00     F*      inquiring into data records as requested.
28.00     F*
29.00     F*-----
30.00     F*
31.00     FP001      IF      E      K      DISK
32.00     FF92801   UP      E      K      DISK
33.00     FV928011 CP      E      WORKSTN      KINFDS      SRVFDS
34.00     F*-----
35.00     F*
36.00     F*      Copy Member for Composite Common Subroutine - C0001
37.00     F*
38.00     F/COPY JDECPY,D0001
39.00     F*-----
40.00     E*
41.00     E*      PROGRAM TABLES AND ARRAYS
42.00     E*      -----
43.00     E*
44.00     E*
45.00     E*
46.00     E*
47.00     E*
48.00     E*
49.00     E*
50.00     E*
51.00     E*      Copy Member for Composite Common Subroutine - C0001
52.00     E*
53.00     E*/COPY JDECPY,E0001
54.00     E*-----
55.00     E*
56.00     E*      Copy Member for Composite Common Subroutine - C0012
57.00     E*
58.00     E/COPY JDECPY,E0012
59.00     E*-----
60.00     E*      Copy Member for Composite Common Subroutine - C997
61.00     E*
62.00     E*
63.00     E/COPY JDECPY,E997
64.00     I*-----
65.00     I*
66.00     I*      PROGRAM INPUT SPECIFICATIONS AND DATA STRUC*URES
67.00     I*      -----
68.00     I*

```

Copyright statement can be changed through the Program Generator

Shows all SARs used to make changes to the program

The Program Generator puts in numeric order. RPG opens from bottom to top so JDE puts more heavily used files at the bottom

Informational data structure for the video

Arrays that handle error messages

Will copy in additional specifications for copy module C0001

```

69.00      I*      Data Structures to Load Video Screen Text
70.00      I*
71.00      IDSTXT      DS      1000
72.00      I          1      18 VTX001
73.00      I*          41      58 VTX002
74.00      I*          81      92 VTX003
75.00      I*         121     138 VTX004
76.00      I*         161     178 VTX005
77.00      I*         201     218 VTX006
78.00      I*         241     258 VTX007
79.00      I*         281     298 VTX008
80.00      I*         321     338 VTX009
81.00      I*         361     378 VTX010
82.00      I*         401     418 VTX011
83.00      I*         441     458 VTX012
84.00      I*         481     498 VTX013
85.00      I*         521     536 VTX014
86.00      I*         561     576 VTX015
87.00      I*         601     616 VTX016
88.00      I*         641     656 VTX017
89.00      I*         681     696 VTX018
90.00      I*         721     736 VTX019
91.00      I*         761     776 VTX020
92.00      I*         801     816 VTX021
93.00      I*         841     856 VTX022
94.00      I*         881     896 VTX023
95.00      I*         921     936 VTX024
96.00      I*         961     976 VTX025
97.00      I*
98.00      I/COPY JDECPY, IOODSINX
99.00      I/COPY JDECPY, IOOPS@@
100.00     I/COPY JDECPY, IOODSPROG
101.00     I*
102.00     I*
103.00     I*
104.00     I*      Copy Member for Composite Common Subroutine - COOSC
105.00     I*
106.00     I/COPY JDECPY, IOOSC
107.00     I*****
108.00     I*
109.00     I*      Copy Member For Server - x0005
110.00     I*
111.00     I/COPY JDECPY, IO005U
112.00     I*****
113.00     I*
114.00     I*      Copy Member For Server - x0006
115.00     I*
116.00     I/COPY JDECPY, IO00661
117.00     I*****
118.00     I*
119.00     I*      Copy Member For Server - x9800E
120.00     I*
121.00     I/COPY JDECPY, I9800e
122.00     I*****
123.00     C*****
124.00     C*      MAINLINE PROGRAM
125.00     C*      -----
126.00     C*
127.00     C*      Process housekeeping.
128.00     C*
129.00     C          EXSR S999
130.00     C          ----
131.00     C*
132.00     C*      If LR on, end program.
133.00     C*
134.00     C          *INLR      CABEQ'1'      EOJ
135.00     C*          -----      ---
136.00     C*
137.00     C*      If automatic inquiry set, process inquiry.
138.00     C*
139.00     C          $AUTO      CASEQ'1'      S003
140.00     C*          -----      ---
141.00     C*          End
142.00     C*
143.00     C*      Begin normal program processing.
144.00     C*      -----
145.00     C*
146.00     C          *INLR      DOWEQ'0'
147.00     C*
148.00     C*      Write video screen.
149.00     C*

```

Each VTX field is 40 long but may not use all 40. Pulls in text from Vocabulary Overrides.

— Data structure for commonly used indexes

— Data structure used with file servers

— Program status data structure

— Data structure for vocabulary overrides and function keys

— Data structure for file server X0005

— One time only functions

24 — If information is passed to this program, it will automatically inquire on the record

```

150.00 C WRITEV9280111
151.00 C MOVE /1/ @@AID
152.00 C EXSR S001
153.00 C* ----- Clears fields
154.00 C*
155.00 C* Load data field dictionary parameters (one cycle only).
156.00 C*
157.00 C $998 CASEQ' ' S998
158.00 C* ----- One time only. Pulls in Data
159.00 C END Dictionary editing information
160.00 C* functions
161.00 C* Begin video screen read processing.
162.00 C*
163.00 C SETOF 999301
164.00 C READ V928011 9998
165.00 C Z-ADDO ##RROW
166.00 C Z-ADDO ##RCOL Used for cursor sensitive help.
167.00 C* Tells where the cursor is.
168.00 C*
169.00 C* If video read timed out, end program.
170.00 C *IN99 CABEQ'1' EOJ LR
171.00 C* -----
172.00 C @@AID CABEQ#FEOJ EOJ LR
173.00 C* -----
174.00 C*
175.00 C* If vaild function key pressed, process and return.
176.00 C*
177.00 C *IN15 IFEQ '1' All function keys are assigned indicator 15 so
178.00 C EXSR SOOEX if 15 is on, a function key has been pressed
179.00 C* -----
180.00 C INLR CABEQ'1' EOJ
181.00 C* -----
182.00 C *IN15 CABEQ'1' END
183.00 C* -----
184.00 C END
185.00 C*
186.00 C* Edit the action code.
187.00 C*
188.00 C EXSR C0001 Edits the action code.
189.00 C* ----- Checks action code security.
190.00 C*
191.00 C* If end of job requested, end program.
192.00 C*
193.00 C @@AID CABEQ#FEOJ EOJ
194.00 C* -----
195.00 C*
196.00 C* If clear screen requested, process and return.
197.00 C*
198.00 C @@AID IFEQ #FCLR
199.00 C EXSR S001
200.00 C* -----
201.00 C GOTO END
202.00 C* -----
203.00 C END
204.00 C*
205.00 C* Load subfile records.
206.00 C*
207.00 C EXSR S003 Sets the file pointer and calls S004
208.00 C* ----- to load the video/report fields
209.00 C*
210.00 C* If add or change, validate all video input.
211.00 C*
212.00 C *IN93 CASEQ'0' S005 If an error has occurred,
213.00 C* ----- validates and edits data
214.00 C END
215.00 C*
216.00 C* If no errors and not inquiry, update file.
217.00 C*
218.00 C *IN93 IFEQ '0'
219.00 C *IN24 CASEQ'0' S010 Updates files
220.00 C* -----
221.00 C END
222.00 C END
223.00 C*
224.00 C* Return for next input.
225.00 C*
226.00 C END TAG
227.00 C* -----
228.00 C*

```

```

229.00 C*      Set correct message in line 24.
230.00 C*
231.00 C      *IN93      IFEQ '1'
232.00 C      MOVLSVL24E      VOL24
233.00 C      ELSE
234.00 C      MOVLSVL24M      VDL24
235.00 C      END
236.00 C*
237.00 C      END
238.00 C*
239.00 C      EOJ      TAG
240.00 C*      ---      ---
241.00 C*
242.00 C*      END MAINLINE PROGRAM
243.00 C*      -----
244.00 C*****
245.00 C*
246.00 C*      Copy Common Subroutine - Edit Action Code
247.00 C*
248.00 C/COPY JDECPY, C0001
249.00 C*****
250.00 C*
251.00 C*      SUBROUTINE SOOEX - Process Function Keys
252.00 C*      -----
253.00 C*
254.00 C      Processing: 1. Determine function key pressed.
255.00 C*                  2. Process function key request.
256.00 C*
257.00 CSR      SOOEX      BEGSR
258.00 C*      -----
259.00 CSR      TOOEXA      Tag
260.00 C*      -----
261.00 C*
262.00 C*      If EOJ requested, exit subroutine.
263.00 C*
264.00 CSR      @@AID      CABEQ#FEOJ      ENDEXE      LR
265.00 C*      -----
266.00 C*
267.00 C*      If Display Keys pressed, exit to help facility and return.
268.00 C*      -----
269.00 C*
270.00 CSR      @@AID      IFEQ #FKEYS
271.00 CSR      CALL 'P9601H'      98
272.00 C*      -----
273.00 CSR      PARM      IOOSC
274.00 CSR      PARM      SRVFDS
275.00 CSR      PARM      IOOCSR
276.00 C*
277.00 CSR      @@AID      CABNE#FKEYS      TOOEXA
278.00 C*      -----
279.00 CSR      GOTO ENDEXE
280.00 C*      -----
281.00 CSR      END
282.00 C*
283.00 C*
284.00 C*      If Cursor Sensitive Help Pressed, exit to CS Help.
285.00 C*      -----
286.00 CSR      @@AID      IFEQ #FQMRK
287.00 CSR      MOVEA*IN      ##IN
288.00 CSR      CALL 'X96CCX'      98
289.00 C*
290.00 CSR      PARM      IGOSC
291.00 CSR      PARM      SRVFDS
292.00 CSR      PARM      IOOCSR
293.00 CSR      PARM ' '      ##CCFF 2
294.00 CSR      PARM      IOGMDE
295.00 C*
296.00 CSR      ##FLDN      IFNE *BLANKS
297.00 CSR      EXSR SOOVL
298.00 C*      -----
299.00 C*      MOVEA##IN      *IN,1
300.00 CSR      END
301.00 CSR      MOVEL*BLANKS      ##DTAI
302.00 CSR      GOTO ENDEXE
303.00 C*      -----
304.00 CSR      END
305.00 C*

```

Sets the message for Line 24

Contains what function key was pressed by the user

External programs start with an X. This is the cursor sensitive help program

Parameters passed identifying where the cursor was when F1 was pressed

```

306.00 C*      If Display errors pressed, exit to error messages
307.00 C*      -----
308.00 C*
309.00 CSR          @@AID      IFEQ #FERD
310.00 CSR          Z-ADD1          #G
311.00 CSR          Z-ADD1          #H
312.00 CSR          #G          DOWLE64
313.00 CSR          @MK,#G      IFEQ '1'
314.00 CSR          MOVE EMK,#G      @ER,#H
315.00 CSR          Add 1          #H
316.00 CSR          END
317.00 CSR          ADD 1          #G
318.00 CSR          END
319.00 CSR          CALL 'POOOOE'          98
320.00 C*          -----
321.00 CSR          PARM          @ER
322.00 CSR          GOTO ENDEXE
323.00 C*          -----
324.00 CSR          END
325.00 C*
326.00 C*      If HELP key pressed, exit to help facility and return.
327.00 C*      -----
328.00 C*
329.00 C*          @@AID      IFEQ #FHELP
330.00 C*          CALL 'POOHELP'          98 Access JDE Help information
331.00 C*          -----
332.00 CSR          PARM          HS@@
333.00 CSR          PARM          HE@@
334.00 CSR          PARM          IOOSC
335.00 CSR          PARM          SRVFDS
336.00 CSR          GOTO ENDEXE
337.00 C*          -----
338.00 CSR          END
339.00 C*
340.00 C*      If Clear screen pressed, clear screen and return.
341.00 C*      -----
342.00 C*
343.00 CSR          @@AID      IFEQ #FCLR
344.00 CSR          EXSR S001
345.00 C*          -----
346.00 CSR          GOTO ENDEXE
347.00 C*          -----
348.00 CSR          END
349.00 C*
350.00 C*      Process roll up and down keys.
351.00 C*      -----
352.00 C*
353.00 CSR          @@AID      IFEQ #FROLU
354.00 CSR          @@AID      OREQ #FROLD
355.00 C*          $SECUR      DOUEQ' '
356.00 CSR          MOVE ' '          $SECUR 1
357.00 C*
358.00 C*      If ROLL UP key pressed, process read next.
359.00 C*      -----
360.00 C*
361.00 CSR          @@AID      IFEQ #FROLU
362.00 C*
363.00 C*      Reset error indicators if roll
364.00 C*
365.00 CSR          MOVEA$RESET      *IN,41
366.00 CSR          MOVE '0'          *IN,40
367.00 CSR          SETOF          818299
368.00 CSR          READ I92801          9981
369.00 CSR          *IN81      IFEQ '1'
370.00 CSR          $RUKEY      SETLLI92801
371.00 CSR          SETOF          8299
372.00 CSR          READ I92801          9982
373.00 C*
374.00 C*      If error on read, set error.
375.00 C*
376.00 CSR          *IN82      IFEQ '1'
377.00 CSR          SETON          9341
378.00 CSR          MOVE '1'          @MK,2
379.00 CSR          GOTO ENDEXE
380.00 C*          -----
381.00 CSR          END
382.00 CSR          END

```



```

384.00      CSR                      END
385.00      C*
386.00      C*      If ROLL DOWN key pressed, process read prior.
387.00      C*      -----
388.00      C*
389.00      CSR          @@AID      IFEQ #FROLD
390.00      C*
391.00      C*      Reset error indicators if roll
392.00      C*
393.00      CSR                      MOVEA$RESET      *IN, 41
394.00      CSR                      MOVE '0'          *IN, 40
395.00      CSR                      SETOF                      818299
396.00      CSR                      READPI92801          9981
397.00      CSR          *IN81      IFEQ '1'
398.00      CSR          $RDKEY      SETLLI92801
399.00      CSR                      SETOF                      8299
400.00      CSR                      READPI92801          9982
401.00      C*
402.00      C*      If error on read, set error.
403.00      C*
404.00      CSR          *IN82      IFEQ '1'
405.00      CSR                      SETON                      9341
406.00      CSR                      MOVE '1'          @MK,2
407.00      CSR                      GOTO ENDEXE
408.00      C*                      ----
409.00      CSR                      END
410.00      CSR                      END
411.00      CSR                      END
412.00      C*
413.00      C*      Load video screen data on roll keys.
414.00      C*      -----
415.00      C*
416.00      CSR          @@AID      IFEQ #FROLU
417.00      CSR          @@AID      OREQ #FROLD
418.00      C*
419.00      C*      Release record lock or report record in use.
420.00      C*
421.00      CSR          *IN99      IFEQ '0'
422.00      CSR                      EXCPTUNLOCK
423.00      CSR                      ELSE
424.00      CSR                      CALL 'P98BLCK'          81
425.00      C*                      -----
426.00      CSR                      PARM                      ##PSDS
427.00      CSR                      SETON                      9341
428.00      CSR                      MOVE '1'          @MK,6
429.00      CSR                      GOTO ENDEXE
430.00      C*                      ----
431.00      CSR                      END
432.00      C*
433.00      C*
434.00      C*      Cost Center security edit.
435.00      C*
436.00      CSR                      MOVE'F92801      ' #FILE
437.00      CSR                      MOVELOXXCC      #MCU
438.00      CSR          #AUT      IFNE '1'
439.00      CSR          #FAUT      ANDNE'1'
440.00      CSR                      EXSR C0000
441.00      C*                      ----
442.00      CSR                      END
443.00      CSR          #AUT      IFNE '1'
444.00      CSR          #FAUT      ANDNE'1'
445.00      CSR          #MAUT      ANDNE'1'
446.00      CSR                      MOVE '1'          $SECUR
447.00      CSR                      END
448.00      CSR          $SECUR      CASEQ' '          S004
449.00      C*                      ----
450.00      CSR                      END
451.00      C*
452.00      CSR                      END
453.00      C*
454.00      CSR                      END
455.00      CSR                      GOTO ENDEXE
456.00      C*                      ----
57.00      CSR                      END
458.00      C*
459.00      CSR          $SAID      IFNE '1'
460.00      CSR                      SETON                      0193
461.00      CSR                      GOTO ENDEXE
462.00      C*                      ----
463.00      CSR                      END
464.00      C*
465.00      CSR                      ENDEXE      ENDSR

```

Program that will display a record lock window when a record in use error is encountered

Could not find a match in the Function Key Definitions for the function key pressed, so program displays *Invalid Function Key* message.

```

466.00 C*****
467.00 C*
469.00 C*   Copy Common Subroutine - Coat Center Security Check
469.00 C*
470.00 C/COPY JDECPY,C0000
471.00 C*****
472.00 C*
473.00 C*   SUBROUTINE SGCVL - Cursor Control Return Values
474.00 C*   -----
475.00 C*
476.00 C*   By format, find the field to upate and move in the
477.00 C*   returned value. If the format is a subfile, the record
478.00 C*   to change is found in @@RRN.
479.00 C*
480.00 CSR           S00VL      BEGSR
481.00 C*           -----
482.00 C*
483.00 CSR           ##RVAL      IFEQ 'BLANK'
484.00 CS           MOVE *BLANK      ##RVAL
485.00 C*           END
486.00 C*
487.00 C*   Return values for fields in format V9280111
488.00 C*
489.00 CSR           ##RFLT      IFEQ 'V9280111'
490.00 C*
491.00 CSR           ##FLDN      IFEQ 'ACTION'
492.00 CSR           MOVE##RVAL      ACTION
493.00 CSR           GOTO ENDOVL
494.00 C*           -----
495.00 CSR           END
496.00 C*
497.00 CSR           ##FLDN      IFEQ 'VDXIT'
498.00 CSR           MOVE##RVAL      VDXIT
499.00 CSR           GOTO ENDOVL
500.00 C*           -----
501.00 CSR           END
502.00 C*
503.00 CSR           ##FLDN      IFEQ 'VDXDS'
504.00 CSR           MOVE##RVAL      VDXDS
505.00 CSR           GOTO ENDOVL
506.00 C*           -----
507.00 CSR           END
508.00 C*
509.00 CSR           ##FLDN      IFEQ 'VDXCC'
510.00 CSR           MOVE##RVAL      VDXCC
511.00 CSR           GOTO ENDOVL
512.00 C*           -----
513.00 CSR           END
514.00 C*
515.00 CSR           ##FLDN      IFEQ 'VDXTY'
516.00 CSR           MOVE##RVAL      VDXTY
517.00 CSR           GOTO ENDOVL
518.00 C*           -----
519.00 CSR           END
520.00 C*
521.00 CSR           ##FLDN      IFEQ 'VDXDT'
522.00 CSR           MOVE##RVAL      VDXDT
523.00 CSR           GOTO ENDOVL
524.00 C*           -----
525.00 CSR           END
526.00 C*
527.00 CSR           ##FLDN      IFEQ 'VDXQT'
528.00 CSR           MOVE##RVAL      VDXQT
529.00 CSR           GOTO ENDOVL
530.00 C*           -----
531.00 CSR           END
532.00 C*
533.00 CSR           ##FLDN      IFEQ 'VDXUM'
534.00 CSR           MOVE##RVAL      VDXUM
535.00 CSR           GOTO ENDOVL
536.00 C*           -----
537.00 CSR           END
538.00 C*
539.00 CSR           ##FLDN      IFEQ 'VDX001'
540.00 CSR           MOVE##RVAL      VDX001
541.00 CSR           GOTO ENDOVL
542.00 C*           -----

```

For cursor sensitive help.

Information was retrieved in program X96CCX. The retrieved information is returned to the video fields in this subroutine.

```

543.00      CSR                      END
544.00      C*
545.00      CSR          ##FLDN      IFEQ 'VDX002      '
546.00      CSR                      MOVE##RVAL      VDX002
547.00      CSR                      GOTO ENDOVL
548.00      C*                      ----
549.00      CSR                      END
550.00      C*
551.00      CSR          #FLDN      IFEQ 'VDX003      '
552.00      CSR                      MOVE##RVAL      VDX003
553.00      CSR                      GOTO ENDOVL
554.00      C*                      ----
555.00      CSR                      END
556.00      C*
557.00      CSR          ##FLDN      IFEQ 'VDX004      '
558.00      CSR                      MOVE##RVAL      VDX004
559.00      CSR                      GOTO ENDOVL
561.00      CSR                      END
562.00      C*
563.00      CSR          ##FLDN      IFEQ 'VDX005      '
564.00      CSR                      MOVE##RVAL      VDX005
565.00      CSR                      GOTO ENDOVL
566.00      C*                      ----
567.00      CSR                      END
568.00      CSR                      END
569.00      C*
570.00      CSR          ENDOVL      ENDSR
571.00      C*****
572.00      C*
573.00      C*      SUBROUTINE S001 - Clear Fields
574.00      C*      -----
575.00      C*
576.00      C*      Processing: 1. Reset all video screen and data file fields
577.00      C*                      for next transaction.
578.00      C*                      2. Clear action code only if requested.
579.00      C*
580.00      CSR          S001      BEGSR
581.00      C*                      ----
582.00      C*
583.00      C*      Reset fields for next transaction.
584.00      C*
585.00      CSR          NOKEY      CLEARI92801
586.00      CSR                      MOVE *BLANK      ###CLF
587.00      CSR                      MOVE *BLANK      ###CRC
588.00      CSR                      Z-ADD*ZERO      ##RCOL
589.00      CSR                      Z-ADD*ZERO      ##RROW
590.00      CSR                      MOVE *BLANK      VDXCC
591.00      CSR                      MOVE *BLANK      VDXDS
592.00      CSR                      MOVE *BLANK      VDXD3
593.00      CSR                      MOVE *BLANK      VDXIT
594.00      CSR                      MOVE *BLANK      VDXQT
595.00      CSR                      MOVE *BLANK      VDXTY
596.00      CSR                      MOVE *BLANK      VDXUM
597.00      CSR                      MOVE *BLANK      VDX001
598.00      CSR                      MOVE *BLANK      VDX002
599.00      CSR                      MOVE *BLANK      VDX003
600.00      CSR                      MOVE *BLANK      VDX004
601.00      CSR                      MOVE *BLANK      VDX005
602.00      CSR                      MOVE$V24M      VDL24
603.00      CSR                      MOVE ' '      @IN37      1
604.00      C*
605.00      C*      Clear action code only if clear screen action.
606.00      C*
607.00      CSR          @@AID      IFEQ #FCLR
608.00      CSR                      MOVE *ALL'0'      $RESET
609.00      CSR                      MOVEA$RESET      *IN,41
610.00      CSR                      MOVE ' '      ACTION      1
611.00      CSR                      Z-ADD*ZERO      QXXIT
612.00      CSR                      MOVE *BLANK      VC0001
613.00      CSR                      MOVE *BLANK      VC0002
614.00      CSR                      MOVE *BLANK      VC0003
615.00      CSR                      MOVE *BLANK      VC0004
616.00      CSR                      MOVE *BLANK      VC0005
617.00      CSR                      MOVE *BLANK      VC0006
618.00      CSR                      MOVE *BLANK      VC0007
619.00      CSR                      MOVE *BLANK      VC0008
620.00      CSR                      Z-ADD*ZERO      $$EDT      60
621.00      CSR                      END
622.00      C*
623.00      CSR                      END001      ENDSR

```

Clears all the fields in the record format for F92801

Clears the video fields

These fields will only be cleared if the user presses the function key to clear the screen. We want to save certain information like key fields and descriptions of they don't get cleared everytime S001 is executed.

```

624.00 C*****
625.00 C*
626.00 C* SUBROUTINE S003 - Edit Key Sets the file pointer and
627.00 C* edit the key
628.00 C*
629.00 C* Processing: 1. Clear error indicators and arrays.
630.00 C* 2. Load input keys.
631.00 C* 3. Validate Master file key.
632.00 C* 4. Release master file record lock.
633.00 C* 5. Load video screen output on inquiry.
634.00 C*
635.00 CSR S003 BEGSR
636.00 C* ----
637.00 C*
638.00 C* Load data field dictionary parameters (one cycle only).
639.00 C*
640.00 CSR $998 CASEQ' ' S998
641.00 C* ----
642.00 CSR END
643.00 C*
644.00 C* Reset error indicators and arrays.
645.00 C*
646.00 CSR MOVE *ALL'0' $RESET 39
647.00 CSR MOVE *BLANK $REST1 63
648.00 CSR MOVEA$RESET *IN, 41
649.00 CSR MOVEA$REST1 @MK, 2
650.00 CSR CLEAR@ER
651.00 C*-----
652.00 C*
653.00 C* Load video input field for - Item ID
654.00 C*
655.00 CSR MOVEAVDXIT @NM
656.00 CSR EXSR C0012
657.00 C* ----
658.00 CSR Z-ADD#NUMR $NBR08 80
659.00 CSR MOVE $NBR08 QXXIT
660.00 C*
661.00 C* Automatic Next Number for - Item ID
662.00 C*
663.00 CSR *IN21 IFEQ '1'
664.00 CSR VDXIT ANDEQ*BLANK
665.00 CSR SETON 81
666.00 CSR *IN81 DOWEQ'1'
667.00 CSR MOVE N@XIT PSIDX 2
668.00 CSR CALL 'X0010' 82
669.00 C* ----
670.00 CSR PARM S@XIT NNSY 4
671.00 CSR PARM PSIDX
672.00 CSR PARM *ZERO #NXTNO 80
673.00 CSR MOVE #NXTNO QXXIT
674.00 CSR MOVE #NXTNO VDXIT
675.00 CSR QXXIT SETLLF92801 8281
676.00 CSR END
677.00 CSR END
678.00 C*-----
679.00 CSR QXKY01 CHAIN192801 9899
680.00 C*
681.00 C* Cost Center security edit.
682.00 C*
683.00 CSR MOVE'F92801 '#FILE
684.00 CSR MOVEQXXCC #MCU
685.00 CSR #AUT IFNE '1'
686.00 CSR #FAUT ANDNE'1'
687.00 CSR EXSR C0000 Checks cost center security
688.00 C* ----
689.00 CSR END
690.00 CSR #AUT IFNE '1'
691.00 CSR #FAUT ANDNE'1'
692.00 C* #MAUT ANDNE'1'
693.00 CSR MOVE '1' $$SECR 1
694.00 CSR END
695.00 C*
696.00 C* If security violation, set error condition.
697.00 C*
698.00 CSR $$SECR IFEQ '1'
699.00 CSR MOVE '1' @MK, 8
700.00 CSR SETON 9341

```

```

701.00      CSR              MOVE ' '          $$SEFCR  1
702.00      CSR              GOTO END003
703.00      C*              -----
704.00      CSR              END
705.00      C*
706.00      C*      Edit result of read and action code.
707.00      C*
708.00      CSR              *IN98      IFEQ '1'
709.00      CSR              *IN21      COMP '0'                      41 *error*
710.00      CSR              ELSE
711.00      CSR              *IN21      COMP '1'                      41 *error*
712.00      CSR              END
713.00      C*
714.00      C*      If indicator 41 on, invalid key for action code.
715.00      C*
716.00      CSR              *IN41      IFEQ '1'
717.00      CSR              MOVE '1'          @MK,2
718.00      CSR              SETON                      93
719.00      CSR              END
720.00      C*
721.00      C*      If indicator 99 on, record in use.
722.00      C*
723.00      CSR              *IN99      IFEQ '1'
724.00      CSR              CALL 'P98RLCK'                      81
725.00      CSR              -----
726.00      CSR              PARM          ##PSDS
727.00      CSR              MOVE '1'          @MK,6
728.00      CSR              SETON                      9341
729.00      CSR              END
730.00      C*-----
731.00      C*
732.00      C*      If not inquiry, skip remainder of subroutine.
733.00      C*
734.00      CSR              *IN24      CABEQ'0'          END003
735.00      CSR              -----
736.00      C*-----
737.00      C*
738.00      C*      Release record lock on master file
739.00      C*
740.00      CSR              *IN98      IFEQ '0'
741.00      CSR              (IN99      ANDEQ'0'
742.00      CSR              EXCPTUNLOCK
743.00      CSR              END
744.00      C*
745.00      CSR      If errors, skip remainder of subroutine.
746.00      C*
747.00      CSR              *IN93      CABEQ'1'          END003
748.00      C*              -----
749.00      C*-----
750.00      C*
751.00      C*      Move data base information to video screen.
752.00      C*
753.00      CSR              EXSR S004
754.00      CSR              -----
755.00      C*-----
756.00      CSR              END003      ENDSR
757.00      C*****
758.00      C*
759.00      C*      Copy Common Subroutine - Right Justify Numeric Fields
760.00      C*
761.00      C/COPY JDECPY, C0012
762.00      C*****
763.00      C*
764.00      C*      SUBROUTINE S004 Load Video Screen Data
765.00      C*      -----
766.00      C*
767.00      C*      Processing 1. Move data base information to video screen.
768.00      C*      All video screen fields re alpha and
769.00      C*      therefore numeric information must be
770.00      C*      processed through subroutine C0014 to set
771.00      C*      proper decimals and provide editing for
772.00      C*      display on screen.
773.00      C*
774.00      C*      Date fields must be converted from their
775.00      C*      internal format of month, day and year or
776.00      C*      Julian to the system format using program
777.00      C*      X0028.

```

JDE uses this or SETLL to release record locks

Moves information to the video/report fields

```

778.00 C*
779.00 CSR          S004  BEGSR
780.00 C*          ----  ----
781.00 C*
782.00 C*
783.00 C*      Move to output -Description for Cost Center
784.00 C*
785.00 CSR          CALL 'X0006'          81
786.00 C*          ----  ----
787.00 CSR          PARM *BLANKS      PSOMOD 1
788.00 CSR          PARM '1'          PSIMOD 1
789.00 CSR          PARM QXXCC        PSMCU 12
790.00 CSR          PARM *BLANKS      PSERRM 4
791.00 CSR          PARM              I0006
792.00 C*
793.00 CSR          MOVE *BLANK      VC0001
794.00 CSR          PDRTRM      IFEQ *BLANK
795.00 CSR          MOVELMCDL01      VC0001
796.00 CSR          END
797.00 C*-----
798.00 C*
799.00 C*      Description display for - Item Type
800.00 C*
801.00 CSR          CLEARI0005U
802.00 CSR          MOVELS@XTY      #USX
803.00 CSR          MOVE R@XTY      #URT
804.00 CSR          MOVE QXXTY      #UKY
805.00 CSR          CALL 'X0005'          81
806.00 C*          ----  ----
807.00 CSR          PARM              *0005U
808.00 CSR          MOVE *BLANK      VC0002
809.00 CSR          #UERR      IFEQ '0'
810.00 CSR          MOVE#UDL01      VC0002
811.00 CSR          END
812.00 C*-----
813.00 C*
814.00 C*      Description display for - Item Unit of Measure
815.00 C*
816.00 CSR          CLEARI0005U
817.00 CSR          MOVELS@XUM      #USY
818.00 CSR          MOVE R@XUM      #URT
819.00 CSR          MOVE QXXUM      #UKY
820.00 CSR          CALL 'X0005'          81
821.00 C*          ----  ----
822.00 CSR          PARM              I0005U
823.00 CSR          MOVE *BLANK      VC0003
824.00 CSR          #UERR      IFEQ '0'
825.00 CSR          MOVE#UDL01      VC0003
826.00 CSR          END
827.00 C*-----
828.00 C*
829.00 C*      Description display for - Item Category Code 001
830.00 C*
831.00 CSR          CLEARI0005U
832.00 CSR          MOVELS@X001      #USY
833.00 CSR          MOVE R@X001      #URT
834.00 CSR          MOVE QXX001      #UKY
835.00 CSR          CALL 'X0005'          81
836.00 C*          ----  ----
837.00 CSR          PARM              I0005U
838.00 CSR          MOVE *BLANK      VC0004
839.00 CSR          #UERR      IFEQ '0'
840.00 CSR          MOVE#UDL01      VC0004
841.00 CSR          END
842.00 C*-----
843.00 C*
844.00 C*      Description display for - Item Category Code 002
845.00 C*
846.00 CSR          CLEARI0005U
847.00 CSR          MOVELS@X002      #USY
848.00 CSR          MOVE R@X002      #URT
849.00 CSR          MOVE QXX002      #UKY
850.00 CSR          CALL 'X0005'          81
851.00 C*          ----  ----
852.00 CSR          PARM              I0005U
853.00 CSR          MOVE *BLANK      VC0005
854.00 CSR          #UERR      IFEQ '0'

```

File server for user defined codes

```

855.00      CSR              MOVE#UDL01      VC0005
856.00      CSR              END
857.00      C*-----
858.00      C*
859.00      C*      Description display for - Item Category Code 003
860.00      CSR
861.00      CSR              CLEARI0005U
862.00      CSR              MOVELS@X003      #USY
863.00      CSR              MOVE R@X003      #URT
864.00      CSR              MOVE QXX003      #UKY
865.00      C*              CALL 'X0005'      81
866.00      CSR              -----
867.00      CSR              PARM              I0005U
868.00      CSR              MOVE *BLANK      VC0006
869.00      CSR              #UERR      IFEQ '0'
870.00      CSR              MOVE#UDL01      VC0005
871.00      CSR              END
872.00      C*-----
873.00      C*
874.00      C*      Description display for - Item Category Code 004
875.00      C*
876.00      CSR              CLEARI0005U
877.00      CSR              MOVELS@X004      #USY
878.00      CSR              MOVE R@X004      #URT
879.00      CSR              MOVE QXX004      #UKY
880.00      C*              CALL 'X0005'      81
881.00      CSR              -----
882.00      CSR              PARM              I0005U
883.00      CSR              MOVE *BLANK      VC0007
884.00      CSR              #UERR      IFEQ '0'
885.00      CSR              MOVE#UDL01      VC0007
886.00      CSR              END
887.00      C*-----
888.00      C*
889.00      C*      Description display for - Item Category Code 005
890.00      C*
891.00      CSR              CLEARI0005U
892.00      CSR              MOVELS@X005      #USY
893.00      CSR              MOVE R@X005      #URT
894.00      CSR              MOVE QXX005      #UKY
895.00      C*              CALL 'X0005'      81
896.00      CSR              -----
897.00      CSR              PARM              I0005U
898.00      CSR              MOVE *BLANK      VC0008
899.00      CSR              #UERR      IFEQ '0'
900.00      CSR              MOVE#UDL01      VC0008
901.00      CSR              END
902.00      C*-----
903.00      C*
904.00      C*      Move to output - Cost Center
905.00      C*
906.00      CSR              MOVE *BLANK      #SINBR
907.00      CSR              MOVE#QXXCC      #SINBR
908.00      CSR              MOVE T@XCC      #DTYP
909.00      CSR              MOVE W@XCC      #EWRD
910.00      CSR              MOVE E@XCC      #EC
911.00      CSR              MOVE F@XCC      #DSPD
912.00      CSR              MOVE G@XCC      #DATD
913.00      CSR              MOVE J@XCC      #ALR
914.00      CSR              MOVE ' '      #ECOR
915.00      CSR              MOVE ' '      #DCOR
916.00      CSR              EXSR C00161
917.00      CSR              -----
918.00      CSR              #ALR      IFEQ 'L'
919.00      CSR              MOVE#SINBR      VDXCC
920.00      CSR              ELSE
921.00      CSR              MOVE #SINBR      VDXCC
922.00      CSR              END
923.00      C*-----
924.00      C*
925.00      C*      Move to output - Description
926.00      C*
927.00      CSR              MOVE#QXXDS      VDXDS
928.00      C*-----
929.00      C*
930.00      C*      Move to Output - Date Last Ship
931.00      C*

```

Editing information retrieved in S998

Copy module to edit field for use on screen/report

```

932.00  CSR          MOVE QXXDT      #SIDAT  6
933.00  CSR          MOVE *BLANK     #EDAT   8
934.00  CSR          MOVEL '*JUL     '#FFMT   7
935.00  CSR          MOVEL '*SYSVAL   '#TFMT   7
936.00  CSR          MOVEL '*SYSVAL   '#SKP    7
937.00  CSR          MOVE ' ' '      $KRTST   7
938.00  CSR          CALL 'X0028     '      81
939.00  C*          -----
940.00  CSR          PARM             #SIDAT
941.00  CSR          PARM             #EDAT
942.00  CSR          PARM             #FFMT
943.00  CSR          PARM             #TFMT
944.00  CSR          PARM             #SKP
945.00  CSR          PARM             $KRTST
946.00  C*          MOVEL#EDAT       VDXDT
947.00  C*          -----
949.00  C*
949.00  C*          Move to output - Item ID
950.00  C*
951.00  CSR          MOVE *BLANK     #SINBR
952.00  CSR          MOVELQXXIT      #SINBR
953.00  CSR          MOVE T@XIT      #DTYP
954.00  CSR          MOVE W@XIT      #EWRD
955.00  CSR          MOVE E@XIT      #EC
956.00  CSR          MOVE F@XIT      #DSPD
957.00  CSR          MOVE G@XIT      #DATD
958.00  CSR          MOVE J@XIT      #ALR
959.00  CSR          MOVE ' ' '      #ECOR
960.00  CSR          MOVE ' ' '      #DCOR
961.00  CSR          EXSR C00161
962.00  C*          -----
963.00  CSR          #ALR          IFEQ 'L'
964.00  CSR          MOVEL#SINBR      VDXIT
965.00  CSR          ELSE
966.00  CSR          MOVE #SINBR      VDXIT
967.00  CSR          END
969.00  C*          -----
969.00  C*
970.00  C*          Move to output - Quantity - On hand
971.00  C*
972.00  CSR          MOVE *BLANK     #SINBR
973.00  CSR          MOVELQXXQT      #SINBR
974.00  CRR          MOVE T@XQT      #DTYP
975.00  CSR          MOVE W@XQT      #RWRD
976.00  CSR          MOVE E@XQT      #EC
977.00  CSR          MOVE F@XQT      #DSPD
978.00  CSR          MOVE G@XQT      #DATD
979.00  CSR          MOVE J@XQT      #ALR
980.00  CSR          MOVE ' ' '      #ECOR
981.00  CSR          MOVE ' ' '      #DCOR
982.00  CSR          EXSR C00161
983.00  C*          -----
984.00  CSR          #ALR          IFEQ 'L'
985.00  CSR          MOVEL#SINBR      VDXQT
986.00  CSR          ELSE
987.00  CSR          MOVE #SINBR      VDXQT
989.00  CSR          END
989.00  C*          -----
990.00  C*
991.00  C*          Move to output - Item Type
992.00  C*
993.00  CSR          MOVELQXXTY      VDXTY
994.00  C*          -----
995.00  C*
996.00  C*          Move to output - Item Unit of Measure
997.00  C*
999.00  CSR          MOVELQXXUM      VDXUM
999.00  C*          -----
1000.00 C*
1001.00 C*          Move to output - Item Category Code 001
1002.00 C*
1003.00 CSR          MOVE *BLANK     #SINBR
1004.00 CSR          MOVELQXX001     #SINBR
1005.00 CSR          MOVE T@X001     #DTYP
1006.00 CSR          MOVE W@X001     #EWRD
1007.00 CSR          MOVE E@X001     #EC
1009.00 CSR          MOVE G@X001     #DATD

```

External program used to  
edit dates.



```

1010.00      CSR                MOVE J@X001      #ALR
1011.00      CSR                MOVE ' '         #ECOR
1012.00      CSR                MOVE ' '         #DCOR
1013.00      CSR                EXSR C00161
1014.00      C*                -----
1015.00      CSR                #ALR      IFEQ 'L'
1016.00      CSR                MOVE#SINBR      VDX0001
1017.00      CSR                ELSE
1018.00      CSR                MOVE #SINBR      VDXIT
1019.00      CSR                END
1020.00      C*-----
1021.00      C*
1022.00      C*      Move to output - Item Category Code 002
1023.00      C*
1024.00      CSR                MOVE *BLANK      #SINBR
1025.00      CSR                MOVE#QXX002     #SINBR
1026.00      CSR                MOVE T@X002     #DTYP
1027.00      CSR                MOVE W@X002     #RWRD
1028.00      CSR                MOVE E@X002     #EC
1029.00      CSR                MOVE F@X002     #DSPD
1030.00      CSR                MOVE G@X002     #DATD
1031.00      CSR                MOVE J@X002     #ALR
1032.00      CSR                MOVE ' '         #ECOR
1033.00      CSR                MOVE ' '         #DCOR
1034.00      CSR                EXSR C00161
1035.00      C*                -----
1036.00      CSR                #ALR      IFEQ 'L'
1037.00      CSR                MOVE#SINBR      VDX002
1038.00      CSR                ELSE
1039.00      CSR                MOVE #SINBR      VDX002
1040.00      CSR                END
1041.00      C*-----
1042.00      C*
1043.00      C*      Move to output - Item Category Code 003
1044.00      C*
1045.00      CSR                MOVE *BLANK      #SINBR
1046.00      CSR                MOVE#QXX003     #SINBR
1047.00      CSR                MOVE T@X003     #DTYP
1048.00      CSR                MOVE W@X003     #EWRD
1049.00      CSR                MOVE E@X003     #EC
1050.00      CSR                MOVE F@X003     #DSPD
1051.00      CSR                MOVE G@X003     #DATD
1052.00      CSR                MOVE J@X003     #ALR
1053.00      CSR                MOVE ' '         #ECOR
1054.00      CSR                MOVE ' '         #DCOR
1055.00      CSR                EXSR C00161
1056.00      C*                -----
1057.00      CSR                #ALR      IFEQ 'L'
1058.00      CSR                MOVE#SINBR      VDX003
1059.00      CSR                ELSE
1060.00      CSR                MOVE #SINBR      VDX003
1061.00      CSR                END
1062.00      C*-----
1063.00      C*
1064.00      C*      Move to output - Item Category Code 004
1065.00      C*
1066.00      CSR                MOVE *BLANK      #SINBR
1067.00      CSR                MOVE#QXX004     #SINBR
1068.00      CSR                MOVE T@X004     #DTYP
1069.00      CSR                MOVE W@X004     #EWRD
1070.00      CSR                MOVE E@X004     #EC
1071.00      CSR                MOVE F@X004     #DSPD
1072.00      CSR                MOVE G@X004     #DATD
1072.00      CSR                MOVE J@X004     #ALR
1074.00      CSR                MOVE ' '         #ECOR
1075.00      CSR                MOVE ' '         #DCOR
1076.00      CSR                EXSR C00161
1077.00      C*                -----
1078.00      CSR                #ALR      IFEQ 'L'
1079.00      CSR                MOVE#SINBR      VDX004
1080.00      CSR                ELSE
1081.00      CSR                MOVE #SINBR      VDX004
1082.00      CSR                END
1083.00      C*-----
1084.00      C*
1085.00      C*      Move to output - Item Category Code 005
1086.00      C*

```

```

1087.00      CSR                MOVE *BLANK          #SINBR
1088.00      CSR                MOVELQXX005         #SINBR
1089.00      CRR                MOVE T@X005          #DTYP
1090.00      CSR                MOVE W@X005          #EWRD
1091.00      CSR                MOVE E@X005          #EC
1092.00      CSR                MOVE F@X005          #DSPD
1093.00      CSR                MOVE G@X005          #DATD
1094.00      CSR                MOVE J@X005          #ALR
1095.00      CSR                MOVE ' '             #ECOR
1096.00      CSR                MOVE ' '             #DCOR
1097.00      CSR                EXSR C00161
1098.00      C*                ----
1099.00      CSR                #ALR      IFEQ 'L'
1100.00      CSR                MOVEL#SINBR          VDX005
1101.00      CSR                ELSE
1102.00      CSR                MOVE #SINBR          VDX005
1103.00      CSR                END
1104.00      C*-----
1105.00      CSR                END004      ENDSR
1106.00      C*****
1107.00      C*
1108.00      C*      Copy Common Subroutine - Format Numeric Fields for Output with Override
1109.00      C*
1110.00      C/COPY JDECPY,C00161
1111.00      C*****
1112.00      C*
1113.00      C*      SUBROUTINE S005 - Scrub Input
1114.00      C*      -----
1115.00      C*
1116.00      C*      Processing: 1.  Validate all video input.
1117.00      C*                      All numeric fields must be processed
1118.00      C*                      through subroutines C0012 and C0015 in order
1119.00      C*                      to scrub the alpha input field and convert
1120.00      C*                      15 digits and 0 decimals.
1121.00      C*
1122.00      C*                      Date fields must be converted from system
1123.00      C*                      format to their internal format of month,
1124.00      C*                      day and year or julian using program X0028.
1125.00      C*      2.  Update data record fields from video.
1126.00      C*
1127.00      CSR                S005      BEGSR
1128.00      C*                ----
1129.00      C*
1130.00      C*      If not addition or change, bypass subroutine
1131.00      C*
1132.00      CSR                *IN21      IFEQ '0'
1133.00      CSR                *IN22      ANDEQ '0'
1134.00      CSR                GOTO END005
1135.00      C*                ----
1136.00      CSR                END
1137.00      C*
1138.00      C*
1139.00      C*
1140.00      C*      Scrub and edit - Cost Center
1141.00      C*
1142.00      CSR                CALL 'X0006'          99
1143.00      C*                ----
1144.00      CSR                PARM '1'              PSOMOD 1
1145.00      CSR                PARM '1'              PSIMOD 1
1146.00      CSR                PARM VDXCC            PSMCU 12
1147.00      CSR                PARM *BLANKS          PSERRM 4
1148.00      CSR                PARM                  I0006
1149.00      C*
1150.00      CSR                PSERRM      IFNE *BLANK
1151.00      CSR                SETON
1152.00      CSR                MOVELPSERRM          EMK,10      4393
1153.00      CSR                MOVE '1'              @MK,10
1154.00      CSR                END
1155.00      CSR                MOVE PSMCU          QXXCC
1156.00      C*-----
1157.00      C*
1158.00      C*      Scrub and edit - Description
1159.00      C*
1160.00      CSR                MOVELVDXDS          QXXDS
1161.00      C*
1162.00      C*      Set default value - Description
1163.00      C*

```

Validates and edits data  
entered by the user

Only performs this  
subroutine if a record is  
added or changed

```

1165.00 CSR QXXDS UFEQ *BLANK
1166.00 CSR D@XDS IFNE *BLANK
1167.00 CSR MOVEAD@XDS @DV
1168.00 CSR MOVE@DV QXXDS
1169.00 CSR @DV,1 IFEQ ' ' ' '
1170.00 CSR MOVE ' ' ' ' @DV,1
1171.00 CSR Z-ADD2 #M
1172.00 CSR #M DOWLE40
1173.00 CSR @DV,#M IFEQ ' ' ' '
1174.00 CSR MOVE ' ' ' ' @DV,#m
1175.00 CSR END
1176.00 CSR ADD 1 #M
1177.00 CSR END
1178.00 CSR MOVEA@DV,2 QXXDS
1179.00 CSR END
1180.00 CSR END
1181.00 CSR END
1182.00 C*
1183.00 C* Edit allowed values - Description
1184.00 C*
1185.00 CSR A@XDS IFEQ '*NB'
1186.00 CSR QXXDS ANDEQ*BLANK
1187.00 CSR MOVE '1' @MK,03
1188.00 CSR SETON 4293
1189.00 CSR END
1190.00 C*-----
1191.00 C*
1192.00 C* Scrub and edit - Date Last Ship
1193.00 C*
1194.00 CSR MOVEAVDXDT @NM A
1195.00 CSR EXSR C0012
1196.00 C*
1197.00 CSR Z-ADD#NUME NBR6 60
1198.00 CSR MOVE $NBR6 QXXDT
1199.00 C*
1200.00 C* Edit julian date - Date Last Ship
1201.00 C*
1202.00 CSR VDXDT IFNE * BLANK
1203.00 CSR MOVE QXXDT #SIDAT 6
1204.00 CSR MOVE *BLANK #EDAT 8
1205.00 CSR MOVE' *SYSVAL ' #FFMT 7
1206.00 CSR MOVE' *JUL ' #TFMT 7
1207.00 CSR MOVE' *NONE ' #SKP 7
1208.00 CSR MOVE' ' ' $ERTST 1
1209.00 CSR CALL 'X0028 ' 99
1210.00 C*
1211.00 CSR PARM #SIDAT
1212.00 CSR PARM #EDAT
1213.00 CSR PARM #FFMT
1214.00 CSR PARM #TFMT
1215.00 CSR PARM #SKP
1216.00 CSR PARM $KRTST
1217.00 CSR MOVE#SIDAT QXXDT
1218.00 CSR $ERTST IFEW '1'
1219.00 CSR MOVE '1' @MK,04
1220.00 CSR SETON 4593
1221.00 CSR END
1222.00 CSR END
1223.00 C*-----
1224.00 C*
1225.00 C* Scrub and edit - Item ID
1226.00 C*
1227.00 CSR MOVEAVDXIT @NM
1228.00 CSR EXSR C0012
1229.00 C*
1230.00 CSR MOVE F@XIT #DSPD
1231.00 CSR MOVE G@XIT #DATD
1232.00 CSR EXSR C00151
1233.00 C*-----
1234.00 CSR MOVE #NUMBR QXXIT
1235.00 C*
1236.00 C* Set default value - Item ID
1237.00 C*
1238.00 CSR VDXIT IFEQ *BLANK
1239.00 CSR D@XIT ANDNE*BLANK
1240.00 CSR MOVEAD@XIT @NM
1241.00 CSR EXSR C0012

```

Work fields used in the  
RPG program begin with \$

Work fields used in a copy  
module begin with #

```

1242.00 C*          -----
1243.00 CSR          MOVE F@XIT      #DSPD
1244.00 CSR          MOVE G@XIT      #DATD
1245.00 CSR          EXSR C00151
1246.00 C*          -----
1247.00 CSR          MOVE #NUMBR      QXXIT
1248.00 CSR          END
1249.00 C*
1250.00 C*          Edit upper and lower range - Item ID
1251.00 C*
1252.00 CSR          L@XIT      IFNE *BLANK
1253.00 CSR          MOVE *BLANK      X@XIT  15
1254.00 CSR          MOVE '1'        $ERTST  1
1255.00 CSR          MOVE LQXXIT      X@XIT
1256.00 CSR          X@XIT      IFEG L@XIT
1257.00 CSR          X@XIT      ANDLEU@XIT
1258.00 CSR          MOVE ' '        $ERTST
1259.00 CSR          END
1260.00 CSR          $ERTST      IFEQ '1'
1261.00 CSR          MOVE '1'        @MK,07
1262.00 CSR          SETON                      4193
1263.00 CSR          END
1264.00 CSR          END
1265.00 C**-----
1266.00 C*
1267.00 C*          Scrub and edit - Quantity - On Hand
1268.00 C*
1269.00 CSR          MOVEAVDXQT      @NM
1270.00 CSR          EXSR C0012
1271.00 C*          -----
1272.00 CSR          MOVE F@XQT      #DSPD
1273.00 CSR          MOVE G@XQT      #DATD
1274.00 CSR          EXSR C00151
1275.00 C*          -----
1276.00 CSR          MOVE #NUMBR      QXXQT
1277.00 C*
1278.00 C*          Set default value - Quantity - On Hand
1279.00 C*
1280.00 CSR          VDXQT      IFEQ *BLANK
1281.00 CSR          D@XQT      ANDNE*BLANK
1282.00 CSR          MOVEAD@XQT      @NM
1283.00 CSR          EXSR C0012
1284.00 C*          -----
1285.00 CSR          MOVE F@XQT      #DSPD
1286.00 CSR          MOVE G@XQT      #DATD
1287.00 CSR          EXSR C00151
1288.00 C*          -----
1289.00 CSR          MOVE #NUMBR      QXXQT
1290.00 CSR          END
1291.00 C*
1292.00 C*          Edit upper and lower range - Quantity - On Hand
1293.00 C*
1294.00 CSR          L@XQT      IFNE *BLANK
1295.00 CSR          MOVE *BLANK      X@XQT  15
1296.00 CSR          MOVE '1'        $ERTST  1
1297.00 CSR          MOVE LQXXQT      X@XQT
1298.00 CSR          X@XQT      IFEG L@XQT
1299.00 CSR          X@XQT      ANDLEU@XQT
1300.00 CSR          MOVE ' '        $ERTST
1301.00 CSR          END
1302.00 CSR          $ERTST      IFEQ '1'
1303.00 CSR          MOVE '1'        @MK,07
1304.00 CSR          SETON                      4693
1305.00 CSR          END
1306.00 CSR          END
1307.00 C**-----
1308.00 C*
1309.00 C*          Scrub and edit - Item Type
1310.00 C*
1311.00 CSR          MOVE LVDXTY      QXXTY
1312.00 C*
1313.00 C*          Set default value - Item Type
1314.00 C*
1315.00 CSR          QXXTY      IFEQ *BLANK
1316.00 CSR          D@XTY      IFNE *BLANK
1317.00 CSR          MOVEAD@XTY      @40
1318.00 CSR          MOVEA@40      QXXTY

```

Default value from Data Dictionary

Upper and lower ranges from Data Dictionary

```

1319.00   CSR      @40,1      IFEQ ' ' ' '
1320.00   CSR      MOVE ' '      @40,1
1321.00   CSR      Z-ADD2      #M
1322.00   CSR      #M          DOWLE40
1323.00   CSR      @40,#M     IFEQ ' ' ' '
1324.00   CSR      MOVE ' '      @40,#M
1325.00   CSR      END
1326.00   CSR      ADD 1      #M
1327.00   CSR      END
1328.00   CSR      MOVEA@40,2    QXXTY
1329.00   CSR      END
1330.00   CSR      END
1331.00   CSR      END
1332.00   C*
1333.00   C*      Edit allowed values - Item Type
1334.00   C*
1335.00   CSR      A@XTY      IFNE *BLANK
1336.00   CSR      A@XTY      IFEQ '*NB'
1337.00   CSR      QXXTY      ANDEQ*BLANK
1338.00   CSR      MOVE '1'      @MK,03
1339.00   CSR      SETON                      4493
1340.00   CSR      ELSE
1341.00   CSR      MOVEAA@XTY      @40
1342.00   CSR      MOVE *HIVAL      @AV
1343.00   CSR      EXSR C997
1344.00   C*      ----
1345.00   CSR      MOVE ' '      $ERTST 1
1346.00   CSR      MOVE *BLANK      $WRK10 10
1347.00   CSR      MOVE LQXXTY      $WRK10
1348.00   CSR      @AV,1          IFNE *HIVAL
1349.00   CSR      $WRK10         LOKPUP@AV                      81
1350.00   CSR      *IN81          IFEQ '0'
1351.00   CSR      MOVE '1'      $ERTST
1352.00   CSR      END
1353.00   CSR      $ERTST         IFEQ '1'
1354.00   C*      MOVE '1'      O*,07
1355.00   CSR      SETON                      4493
1356.00   CSR      END
1357.00   CSR      END
1358.00   CSR      END
1359.00   CSR      END
1360.00   C*
1361.00   C*      Edit upper and lower range - Item Type
1362.00   C*
1363.00   CSR      LQXTY      IFNE *BLANK
1364.00   CSR      MOVE '1'      $ERTST
1365.00   CSR      QXXTY      IFGE L@XTY
1366.00   CSR      QXXTY      ANDLEU@XXTY
1367.00   CSR      MOVE ' '      $ERTST
1368.00   CSR      END
1369.00   CSR      $ERTST         IFEQ '1'
1370.00   CSR      MOVE '1'      @MK,07
1371.00   CSR      SETON                      4493
1372.00   CSR      END
1373.00   CSR      END
1374.00   C*
1375.00   C*      Edit from User Defined Codes - Item Type
1376.00   C*
1377.00   CSR      R@XTY      IFNE *BLANK
1378.00   CSR      CLEARZ0005U
1379.00   C*      MOVE LS@XTY      #USY
1380.00   CSR      MOVE R@XTY      #URT
1381.00   CSR      MOVE QXXTY      #UKR
1382.00   CSR      CALL 'X0005'                      81
1383.00   C*
1384.00   CSR      #UERR        PARM          I0005U
1385.00   CSR      IFEQ '1'
1386.00   CSR      MOVE '1'      @MK,09
1387.00   CSR      SETON                      4493
1388.00   CSR      END
1389.00   CSR      END
1390.00   C*-----
1391.00   C*
1392.00   C*      Scrub and edit - Item Unit of Measure
1393.00   C*
1394.00   CSR      MOVE LVDXUN      QXXUM
1395.00   C*

```

```

1396.00 C*      Set default value - Item Unit of Measure
1397.00 C*
1398.00 CSR      QXXUM      IFEQ *BLANK
1399.00 CSR      E!XUM      IFNE *BLANK
1400.00 CSR      MOVEAD@XUM      @40
1401.00 CSR      MOVEA@40      QXXUM
1402.00 CSR      @40,1      IFEQ ' ' ' '
1403.00 CSR      MOVE ' '      @40,1
1404.00 CSR      Z-ADD2      #M
1405.00 CSR      #M      DOWLE40
1406.00 C*      @40,#M      IFEQ ' ' ' '
1407.00 CSR      MOVE ' '      @40,#MN
1408.00 CSR      END
1409.00 CSR      ADD 1      #M
1410.00 CSR      END
1411.00 CSR      MOVEA@40,2      QXXUM
1412.00 CSR      END
1413.00 CSR      END
1414.00 CSR      END
1415.00 C*
1416.00 C*      Edit allowed values - Item Unit of Measure
1417.00 C*
1418.00 CSR      A@XUM      IFNE *BLANK
1419.00 CSR      A@XUM      IFEQ '*NB'
1420.00 CSR      QXUM      ANDEQ*BLANK
1421.00 CSR      MOVE '1'      @MK,03
1422.00 CSR      SETON      4793
1423.00 CSR      ELSE
1424.00 CSR      MOVEAA@XUM      @40
1425.00 CSR      MOVE *HIVAL      @AV
1426.00 CSR      EXSR C997
1427.00 C*      ----
1428.00 CSR      MOVE ' '      $ERTST 1
1429.00 CSR      MOVE *BLANK      $WRK10 10
1430.00 CSR      MOVELOQXXUM      $WRK10
1431.00 CSR      @AV,1      IFNE *HIVAL
1432.00 CSR      $WRK10      LOKUP@AV      81
1433.00 CSR      *IN8I      IFEQ '0'
1434.00 CSR      MOVE '1'      $ERTST
1435.00 CSR      END
1436.00 C*      $ERTST      IFEQ '1'
1437.00 CSR      MOVE '1'      @MK,07
1438.00 CSR      SETON      4793
1439.00 CSR      END
1440.00 CSR      END
1441.00 CSR      END
1442.00 CSR      END
1443.00 C*
1444.00 C*      Edit upper and lower range - Item Unit of Measure
1445.00 C*
1446.00 CSR      L@XUN      IFNE *BLANK
1447.00 CSR      MOVE '1'      $ERTST
1448.00 CSR      QXXUM      IFGE L@XUM
1449.00 CSR      OXTON      kNO~U0X0N
1450.00 CSR      MOVE ' '      $ERTST
1451.00 CSR      END
1452.00 CSR      $ERTST      IFEQ '1'
1453.00 C*      MOVE '1'      @MK,07
1454.00 CSR      SETON      4793
1455.00 CSR      END
1456.00 C*      END
1457.00 C*
1458.00 C*      Edit from User Defined Codes - Item Unit of Measure
1459.00 C*
1460.00 CSR      R@XUM      IFNE *BLANK

```

```

1473.00 C*-----
1474.00 C*
1475.00 C*      Scrub and edit - Item Category Code 001
1476.00 C*
1477.00 CSR              MOVELVDX001      QXX001
1478.00 C*
1479.00 C*      Set default value - Item Category Code 001
1480.00 C*
1481.00 CSR              QXX001      IFEQ *BLANK
1482.00 CSR              D@X001      IFNE *BLANK
1483.00 CSR              MOVEAD@X001      @40
1484.00 CSR              MOVEA@40      QXX001
1485.00 CSR              @40,1      IFEQ ' ' ' '
1486.00 CSR              MOVE ' ' ' '      @40,1
1487.00 CSR              Z-ADD2      #M
1488.00 CSR              #M      DOWLE40
1489.00 CSR              @40,#m      IFEQ
1490.00 CSR              MOVE ' ' ' '      @40,#M
1491.00 CSR              END
1492.00 CSR              ADD 1      #M
1493.00 CSR              END
1494.00 CSR              MOVEA@40,2      QXX001
1495.00 CSR              END
1496.00 CSR              END
1497.00 CSR              END
1498.00 C*
1499.00 C*      Edit allowed values - Item Category Code 001
1500.00 C*
1501.00 CSR              A@X001      IFNE *BLANK
1502.00 CSR              A@X001      IFEQ '*NB'
1503.00 CSR              QXX001      ANDEQ*BLANK
1504.00 CSR              MOVE '1'      @Mk,03      4893
1505.00 CSR              SETON
1506.00 CSR              ELSE
1507.00 CSR              MOVEAA@X001      @40
1508.00 CSR              MOVE *HIVAL      @AV
1509.00 CSR              EXSR C997
1510.00 C*
1511.00 CSR              MOVE ' ' ' '      $ERTST 1
1512.00 CSR              MOVE *BLANK      $WRK10 10
1513.00 CSR              MOVELVQXX001      $WRK10
1514.00 CSR              @AV,1      IFNE *HIVAL
1515.00 CSR              $WRK10      LOKUP@AV      81
1516.00 CSR              *IN81      IFEQ '0'
1517.00 CSR              MOVE '1'      $ERTST
1518.00 CSR              END
1519.00 CSR              $ERTST      IFEQ '1'
1520.00 CSR              MOVE '1'      @MK,07      4893
1521.00 CSR              SETON
1522.00 CSR              END
1523.00 CSR              END
1524.00 CSR              END
1525.00 CSR              END
1526.00 C*
1527.00 C*      Edit upper and lower range - Item Category Code 001
1528.00 C*
1529.00 CSR              L@X001      IFNE *BLANK
1530.00 CSR              MOVE '1'      $ERTST
1531.00 CSR              QXX001      IFGE L@X001
1532.00 CSR              QXX001      ANDLEU@X001
1533.00 CSR              MOVE ' ' ' '      $ERTST
1534.00 CSR              END
1535.00 CSR              $ERTST      IFEQ '1'
1536.00 CSR              MOVE '1'      @MK,07      4893
1537.00 CSR              SETON
1538.00 CSR              END
1539.00 CSR              END
1540.00 C*
1541.00 C*      Edit from User Defined Codes - Item Category Code 001
1542.00 C*
1543.00 CSR              R@X001      IFNE *BLANK
1544.00 CSR              CLEARI0005U
1545.00 CSR              MOVELS@X001      #USY
1546.00 CSR              MOVE R@X001      #URT
1547.00 CSR              MOVE QXX001      #UKY
1548.00 CSR              CALL 'X0005'      81
1549.00 C*

```

```

1550.00  CSR          PARM          I0005U
1551.00  CSR          IFEQ '1'
1552.00  CSR          #UERR        MOVE '1'          @MK,09
1553.00  CSR          SETON
1554.00  CSR          END
1555.00  CSR          END
1556.00  C*-----
1557.00  C*
1558.00  C*      Scrub and edit - Item Category Code 002
1559.00  C*
1560.00  CSR          MOVELVDX002    QXX002
1561.00  C*
1562.00  C*      Set default value - Item Category Code 002
1563.00  C*
1564.00  CSR          QXX002        IFEQ *BLANK
1565.00  CSR          D@X002        IFNE *BLANK
1566.00  CSR          MOVEADX002     @40
1567.00  CSR          MOVEA@40       QXX002
1568.00  CSR          @40,1         IFEQ ' '
1569.00  CSR          MOVE ' '       @40,1
1570.00  CSR          Z-ADD2         #M
1571.00  CSR          #M            DOWLE40
1572.00  CSR          @40,#M        IFEQ ' '
1573.00  CSR          MOVE ' '       @40,#M
1574.00  CSR          END
1575.00  CSR          ADD 1          #M
1576.00  CSR          END
1577.00  CSR          MOVEA@40,2     QXX002
1578.00  CSR          END
1579.00  CSR          END
1580.00  CSR          END
1581.00  C*
1582.00  C*      Edit allowed values - Item Category Code 002
1583.00  C*
1584.00  CSR          A@X002        IFNE *BLANK
1585.00  CSR          A@X002        IFEQ '*NB'
1586.00  CSR          QXX002        ANDEQ*BLANK
1587.00  CSR          MOVE '1'       @MK,03
1588.00  CSR          SETON
1589.00  CSR          ELSE
1590.00  CSR          MOVEAAX002     @40
1591.00  CSR          MOVE *HIVAL     @AV
1592.00  CSR          EXSR C997
1593.00  C*      ----
1594.00  CSR          MOVE ' '       $ERTST 1
1595.00  CSR          MOVE *BLANK     $WRK10 10
1596.00  CSR          MOVELQXX002     $WRK10
1597.00  CSR          @AV,1          IFNE *HIVAL
1598.00  CSR          $WFRK10        LOKUP@AV
1599.00  CSR          *IN81          IFEQ '0'
1600.00  CSR          MOVE '1'       $ERTST
1601.00  CSR          END
1602.00  CSR          $ERTST         IFEQ '1'
1603.00  CSR          MOVE '1'       @MK,07
1604.00  CSR          SETON
1605.00  CSR          END
1606.00  CSR          END
1607.00  CSR          END
1608.00  CSR          END
1609.00  C*
1610.00  C*      Edit upper and lower range - Item Category Code 002
1611.00  C*
1612.00  CSR          L@X002        IFNE *BLANK
1613.00  CSR          MOVE '1'       $ERTST
1614.00  CSR          QXX002        IFGE L@X002
1615.00  CSR          QXX002        ANDLEU@X002
1616.00  CSR          MOVE ' '       $ERTST
1617.00  CSR          END
1618.00  CSR          $ERTST         IFEQ '1'
1619.00  CSR          MOVE '1'       @MK,07
1620.00  CSR          SETON
1621.00  CSR          END
1622.00  CSR          END
1623.00  C*
1624.00  C*      Edit from User Defined Codes - Item Category Code 002
1625.00  C*
1626.00  CSR          R@X002        IFNE *BLANK

```



```

1627.00 CSR CLEARI0005U
1628.00 CSR MOVELS@X002 #USY
1629.00 CSR MOVE R@X002 #URT
1630.00 CSR MOVE QXX002 OUKr
1631.00 CSR CALL 'X0005' 81
1632.00 C* -----
1633.00 CSR PARM I0005U
1634.00 CSR #UERR IFEQ '1'
1635.00 CSR MOVE '1' @MK,09
1636.00 CSR SETON 4993
1637.00 CSR END
1638.00 CSR END
1639.00 C*-----
1640.00 C*
1641.00 C* Scrub and edit - Item Category Code 003
1642.00 C*
1643.00 CSR MOVELVDX003 QXX003
1644.00 C*
1645.00 C* Set default value - Item Category Code 003
1646.00 C*
1647.00 CSR QXX003 IFEQ *BLANK
1648.00 CSR D@X003 IFNE *BLANK
1649.00 CSR MOVEAD@X003 @40
1650.00 CSR MOVEA@40 QXX003
1651.00 CSR @40,1 IFEQ ' ' '
1652.00 CSR MOVE ' ' @40,1
1653.00 CSR Z-ADD2 #M
1654.00 CSR #M DOWLE40
1655.00 CSR @40,#M IFEQ ' ' '
1656.00 CSR MOVE ' ' @40,#M
1657.00 CSR END
1658.00 CSR ADD 1 #M
1659.00 CSR END
1660.00 CSR MOVEA@40,2 QXX003
1661.00 CSR END
1662.00 CSR END
1663.00 CSR END
1664.00 C*
1665.00 C* Edit allowed values - Item Category Code 003
1666.00 C*
1667.00 CSR A@X003 IFNE *BLANK
1668.00 CSR A@X003 IFEQ '*NB'
1669.00 CSR QXX003 ANDEQ*BLANK
1670.00 CSR MOVE '1' @MK,03
1671.00 CSR SETON 5093
1672.00 CSR ELSE
1673.00 CSR MOVEAA@003 @40
1674.00 CSR MOVE *HIVAL @AV
1675.00 CSR EXSR C997
1676.00 C* -----
1677.00 CSR MOVE ' ' $ERTST 1
1678.00 CSR MOVE *BLANK $WRK10 10
1679.00 CSR MOVELQXX003 $WRK10
1680.00 CSR @AV,1 IFNE *HIVAL
1681.00 CSR $WRK10 LOKUP@AV 81
1682.00 CSR *IN81 IFEQ '0'
1683.00 CSR MOVE '1' $ERTST
1684.00 CSR END
1685.00 CSR $ERTST IFEQ '1'
1686.00 CSR MOVE '1' @MK,07
1687.00 CSR SETON 5093
1688.00 CSR END
1689.00 CSR END
1690.00 CSR END
1691.00 CSR END
1692.00 C*
1693.00 C* Edit upper and lower range - Item Category Code 003
1694.00 C*
1695.00 CSR L@X003 IFNE *BLANK
1696.00 CSR MOVE '1' $ERTST
1697.00 CSR QXX003 IFGE L@X003
1698.00 CSR QXX003 ANDLEU@X003
1699.00 CSR MOVE ' ' $ERTST
1700.00 CSR END
1701.00 CSR $ERTST IFEQ '1'
1702.00 CSR MOVE '1' @MK,07
1703.00 CSR SETON 5093

```

```

1704.00  CSR                      END
1705.00  CSR                      END
1706.00  C*
1707.00  C*      Edit from User Defined Codes - Item Category Code 003
1708.00  C*
1709.00  CSR          R@X003      IFNE *BLANK
1710.00  CSR                      CLEARI0005U
1711.00  CSR                      MOVELS@X003      #USY
1712.00  CSR                      MOVE R@X003      #URT
1713.00  CSR                      MOVE QXX003      #UKY
1714.00  CSR                      CALL 'X0005'      81
1715.00  C*                      ----
1716.00  CSR                      PARM                      I0005U
1717.00  CSR          #UERR      IFEQ '1'
1718.00  CSR                      MOVE '1'      @MK,09
1719.00  CSR                      SETON                      5093
1720.00  CSR                      END
1721.00  CSR                      END
1722.00  C*-----
1723.00  C*
1724.00  C*      Scrub and edit - Item Category Code 004
1725.00  C*
1726.00  CSR                      MOVELVDX004      QXX004
1727.00  C*
1728.00  C*      Set default value - Item Category Code 004
1729.00  C*
1730.00  CSR          QXX004      IFEQ *BLANK
1731.00  CSR          D@X004      IFNE *BLANK
1732.00  CSR                      MOVEAD@X004      @40
1733.00  CSR                      MOVEA@40      QXX004
1734.00  CSR          @40,1      IFEQ ' '
1735.00  CSR                      MOVE ' '      @40,1
1736.00  CSR                      Z-ADD2      #M
1737.00  CSR          #M      DOWLE40
1738.00  CSR          @40,#M      IFEQ ' '
1739.00  CSR                      MOVE ' '      @40,#M
1740.00  CSR                      END
1741.00  CSR                      ADD 1      #M
1742.00  CSR                      END
1743.00  CSR                      MOVEA@40,2      QXX004
1744.00  CSR                      END
1745.00  CSR                      END
1746.00  CSR                      END
1747.00  C*
1748.00  C*      Edit allowed values - Item Category Code 004
1749.00  C*
1750.00  CSR          A@X004      IFNE *BLANK
1751.00  CSR          A@X004      IFEQ '*NB'
1752.00  CSR          QXX004      ANDEQ*BLANK
1753.00  CSR                      MOVE '1'      @MK,03
1754.00  CSR                      SETON                      5193
1755.00  CSR                      ELSE
1756.00  CSR                      MOVEAA@X004      @40
1757.00  CSR                      MOVE *HIYAL      @AV
1758.00  CSR                      EXSR C997
1759.00  C*                      ----
1760.00  CSR                      MOVE ' '      $ERTST
1761.00  CSR                      MOVE *BLANK      $WRK10 10
1762.00  CSR                      MOVELQXX004      $WRK10
1763.00  CSR          @AV,1      IFNE *HIVAL
1764.00  CSR          $WRK10      LOKUP@AV      81
1765.00  CSR          *IN81      IFEQ '0'
1766.00  CSR                      MOVE '1'      $ERTST
1767.00  CSR                      END
1768.00  CSR          $ERTST      IFEQ '1'
1769.00  CSR                      MOVE '1'      @MK,07
1770.00  CSR                      SETON                      5193
1771.00  CSR                      END
1772.00  CSR                      END
1773.00  CSR                      END
1774.00  CSR                      END
1775.00  C*
1776.00  C*      Edit upper and lower range - Item Category Code 004
1777.00  C*
1778.00  CSR          L@X004      IFNE *BLANK
1779.00  CSR                      MOVE '1'      $ERTST
1780.00  CSR          QXX004      IFGE L@X004

```

```

1781.00 CSR          QXX004  ANDLEU@X004
1782.00 CSR          MOVE '          $ERTST
1783.00 CSR          END
1704.00 CSR          $ERTST  IFEQ '1'
1785.00 CSR          MOVE '1'          @MK,07
1786.00 CSR          SETON                      5193
1797.00 CSR          END
1788.00 CSR          END
1789.00 C*
1790.00 C*      Edit from User Defined Codes - Item Category Code 004
1791.00 C*
1792.00 CSR          R@X004  IFNE *BLANK
1793.00 CSR          CLEARI0005U
1794.00 CSR          MOVELS@X004          #USY
1795.00 CSR          MOVE R@X004          #URT
1796.00 CSR          MOVE QXX004          #UKY
1797.00 CSR          CALL 'X0005'                      81
1798.00 C*          -----
1799.00 CSR          PARM                      I0005U
1800.00 CSR          #UERR  IFEQ '1'
1801.00 CSR          MOVE '1'          @MK,09
1802.00 CSR          SETON                      5193
1803.00 CSR          END
1804.00 CSR          END
1805.00 C*-----
1806.00 C*
1807.00 C*      Scrub and edit - Item Category Code 005
1808.00 C*
1809.00 CSR          MOVELVXOOS          QXX005
1810.00 C*
1811.00 C*      Set default value - Item Category Code 005
1812.00 C*
1813.00 CSR          QXX005  IFEQ *BLANK
1814.00 CSR          D@XOOS  IFNE *BLANK
1815.00 CSR          MOVEAD@XOOS          @40
1816.00 CSR          MOVEA@40          QXX005
1817.00 CSR          @40,1  IFEQ ' ' ' '
1818.00 CSR          MOVE ' ' ' '          @40,1
1819.00 CSR          Z-ADD2          #M
1820.00 CSR          #M  DOWLE40
1821.00 CSR          @40, #M  IFEQ ' ' ' '
1822.00 CSR          MOVE ' ' ' '          @40, #M
1823.00 CSR          END
1824.00 CSR          ADD 1          #M
1825.00 CSR          END
1826.00 CSR          MOVEA@40,2          QXX005
1827.00 CSR          END
1828.00 CSR          END
1829.00 CSR          END
1830.00 C*
1831.00 C*      Edit allowed values - Item Category Code 005
1832.00 C*
1833.00 CSR          A@X005  IFNE *BLANK
1834.00 CSR          A@X005  IFEQ '*NB'
1835.00 CSR          QXX005  ANDEQ*BLANK
1836.00 CSR          MOVE '1'          @MK,03
1837.00 CSR          SETON                      5293
1838.00 CSR          ELSE
1839.00 CSR          MOVEAA@X005          @40
1840.00 CSR          MOVE *HIVAL          @AV
1841.00 CSR          EXSR C997
1842.00 C*          -----
1843.00 CSR          MOVE ' ' ' '          $ERTST 1
1844.00 CSR          MOVE *BLANK          $WRK10 10
1845.00 CSR          MOVELQXX005          $WRK10
1846.00 CSR          @AV,1  IFNE *HIVAL
1847.00 CSR          $WRK10  LOKUP@AV                      81
1848.00 CSR          *IN81  IFEQ '0'
1849.00 CSR          MOVE '1'          $ERTST
1850.00 CSR          END
1851.00 CSR          $ERTST  IFEQ '1'
1852.00 CSR          MOVE '1'          @MX.07
1853.00 CSR          SETON                      5293
1854.00 CSR          END
1855.00 CSR          END
1856.00 CSR          END
1857.00 CSR          END

```

```

1858.00 C*
1859.00 C*      Edit upper and lower range - Item Category Code 005
1860.00 C*
1861.00 CSR      L@X005      IFNE *BLANK
1862.00 CSR      MOVE '1'      $ERTST
1863.00 CSR      QXX005      IFGE L@X005
1864.00 CSR      QXX005      ANDLEU@X005
1865.00 CSR      MOVE ' '      $ERTST
1866.00 CSR      END
1867.00 CSR      $ERTST      IFEQ '1'
1868.00 CSR      MOVE '1'      @MK,07
1869.00 CSR      SETON
1870.00 CSR      END
1871.00 CSR      END
1872.00 C*
1873.00 C*      Edit from User Defined Codes - Item Category Code 005
1874.00 C*
1875.00 CSR      R@X005      IFNE *BLANK
1876.00 CSR      CLEARI0005U
1877.00 CSR      MOVELS@X005      #USY
1878.00 CSR      MOVE R@X005      #URT
1879.00 CSR      MOVE QXX005      #UKY
1880.00 CSR      CALL 'X0005'
1881.00 C*
1882.00 CSR      PARM      I0005U
1883.00 CSR      #UERR      IFEQ '1'
1884.00 CSR      MOVE '1'      @MK,09
1885.00 CSR      SETON
1886.00 CSR      END
1887.00 CSR      END
1888.00 C*-----
1889.00 CSR      END005      ENDSR
1890.00 C*****
1891.00 C*
1892.00 C*      Copy Common Subroutine - Currency - Translate Video Fields to Data Base
1893.00 C*
1894.00 C/COPY JDECPY,C00151
1895.00 C*****
1896.00 C*
1897.00 C*      Copy Common Subroutine - Build Allowed Values Work Array
1898.00 C*
1899.00 C/COPY JDECPY,C997
1900.00 C*****
1901.00 C*
1902.00 C*      Subroutine S010 - Update Data Base
1903.00 C*
1904.00 C*
1905.00 C*      Processing: 1. Update data base file based upon valid
1906.00 C*                  action codes.
1907.00 C*
1908.00 CSR      S010      BEGSR
1909.00 C*
1910.00 C*
1911.00 C*      If add action, add record.
1912.00 C*
1913.00 CSR      *IN21      IFEQ '1'
1914.00 CSR      WRITEI92801
1915.00 CSR      END
1916.00 C*
1917.00 C*      If change action, update record.
1918.00 C*
1919.00 CSR      *IN22      IFEQ '1'
1920.00 CSR      UPDATI92801
1921.00 CSR      END
1922.00 C*
1923.00 C*      If delete action, delete record.
1924.00 C*
1925.00 CSR      *IN23      IFEQ '1'
1926.00 CSR      DELETI92801
1927.00 CSR      END
1928.00 C*

```

```

1929.00 C* Clear data field for next transaction
1930.00 C*
1931.00 CSR MOVE #PCLR @@AID
1932.00 CSR EXSR S001
1933.00 C* -----
1934.00 CSR END010 ENDSR
1935.00 C*****
1935.00 C*
1936.00 C* SUBROUTINE S998 - Load dictionary parameters.
1937.00 C* -----
1938.00 C*
1939.00 CSR S998 BEGSR
1940.00 C* -----
1941.00 C*
1942.00 C*
1943.00 C*
1944.00 C* Dictionary parameters for - Cost Center
1945.00 C*
1946.00 CSR MOVE *BLANK FRDTAI
1947.00 CSR MOVE 'XCC' FRDTAI
1948.00 CSR CALL 'X9800E' 81
1949.00 C* -----
1950.00 CSR PARM I9800E
1951.00 CSR FRERR IFRQ '0'
1952.00 CSR MOVE FRDSCR B@XCC 40
1953.00 CSR MOVE FRDTAT T@XCC 1
1954.00 CSR MOVE FREC E@XCC 1
1955.00 CSR MOVE FRDTAS C@XCC 50
1956.00 CSR MOVE FROTAD G@XCC 20
1957.00 CSR MOVE FRCDEC F@XCC 1
1958.00 CSR MOVE LFRSY S@XCC 4
1959.00 CSR MOVE FRRT R@XCC 2
1960.00 CSR MOVE FRDVAL D@XCC 40
1961.00 CSR MOVE FRVAL A@XCC 40
1962.00 CSR MOVE FRLVAL L@XCC 40
1963.00 CSR MOVE FRUVAL U@XCC 40
1964.00 CSR MOVE FREDWR W@XCC 30
1965.00 CSR MOVE FRLR J@XCC 1
1966.00 CSR MOVE FRNNIX N@XCC 20
1967.00 CSR Z-ADD1 #@XCC 110
1968.00 CSR MOVE F@XCC #A
1969.00 CSR DO #A
1970.00 CSR MULT 10 #@XCC
1971.00 CSR END
1972.00 CSR END
1973.00 C*-----
1974.00 C*
1975.00 C* Dictionary parameters for - Description
1976.00 C*
1977.00 CSR MOVE *BLANK FRDTAI
1978.00 CSR MOVE 'XDS' FRDTAI
1979.00 CSR CALL 'X9800E' 81
1980.00 C* -----
1981.00 CSR PARM I9800E
1982.00 CSR FRERR IFEQ '0'
1983.00 CSR MOVE FRDSCR B@XDS 40
1984.00 CSR MOVE FRDTAT T@XDS 1
1985.00 CSR MOVE FREC E@XDS 1
1986.00 CSR MOVE FRDTAS C@XDS 50
1987.00 CSR MOVE FRDTAD G@XDS 20
1988.00 CSR MOVE FRCDEC F@XDS 1
1989.00 CSR MOVE LFRSY S@XDS 4
1990.00 CSR MOVE FRRT R@XDS 2
1991.00 CSR MOVE FRDVAL D@XDS 40
1992.00 CSR MOVE FRVAL A@XDS 40
1993.00 CSR MOVE FRLVAL L@XDS 40
1994.00 CSR MOVE FRUVAL U@XDS 40
1995.00 CSR MOVE FREDWR W@XDS 30
1996.00 CSR MOVE FRLR J@XDS 1
1997.00 CSR MOVE FRNNIX N@XDS 20
1998.00 CSR Z-1DD1 #@XDS 110
1999.00 CSR MOVE F@XDS #A
2000.00 CSR DO #A
2001.00 CSR MULT 10 #@XDS
2002.00 CSR END
2003.00 CSR END
2004.00 C*-----
2005.00 C*

```

Forces clear of everything before processing next record. Simulates user pressing the *Clear Screen* function key.

Retrieves all of the Data Dictionary editing parameters for necessary data items used in the program and moves the information into constant fields

Data Dictionary file server

```

2006.00 C* Dictionary parameters for - Date Last Ship
2007.00 C*
2008.00 CSR MOVE *BLANK FRDTAI
2009.00 CSR MOVE 'XDT' FRDTAI
2010.00 CSR CALL 'X9800E' 81
2011.00 C* -----
2012.00 CSR PARM I9800E
2013.00 CSR FRERR IFRQ '0'
2014.00 CSR MOVE FRDSCR B@XDT 40
2015.00 CSR MOVE FRDTAT T@XDT 1
2016.00 CSR MOVE FREC E@XDT 1
2017.00 CSR MOVE FRDTAS C@XDT 50
2018.00 CSR MOVE FROTAD G@XDT 20
2019.00 CSR MOVE FRCDEC F@XDT 1
2020.00 CSR MOVE LFRSY S@XDT 4
2021.00 CSR MOVE FRRT R@XDT 2
2022.00 CSR MOVE FRDVAL D@XDT 40
2023.00 CSR MOVE FRVAL A@XDT 40
2024.00 CSR MOVE FRLVAL L@XDT 40
2025.00 CSR MOVE FRUVAL U@XDT 40
2026.00 CSR MOVE FREDWR W@XDT 30
2027.00 CSR MOVE FRLR J@XDT 1
2028.00 CSR MOVE FRNNIX N@XDT 20
2029.00 CSR Z-ADD1 #@XDT 110
2030.00 CSR MOVE F@XDT #A
2031.00 CSR DO #A
2032.00 CSR MULT 10 #@XDT
2033.00 CSR END
2034.00 CSR END
2035.00 C* -----
2036.00 C*
2037.00 C* Dictionary parameters for - Item ID
2038.00 C*
2039.00 CSR MOVE *BLANK FRDTAI
2040.00 CSR MOVE 'XIT' FRDTAI
2041.00 CSR CALL 'X9800E' 81
2042.00 C* -----
2043.00 CSR PARM I9800E
2044.00 CSR FRERR IFEQ '0'
2045.00 CSR MOVE FRDSCR B@XIT 40
2046.00 CSR MOVE FRDTAT T@XIT 1
2047.00 CSR MOVE FREC E@XIT 1
2048.00 CSR MOVE FRDTAS C@XIT 50
2049.00 CSR MOVE FRDTAD G@XIT 20
2050.00 CSR MOVE FRCDEC F@XIT 1
2051.00 CSR MOVE LFRSY S@XIT 4
2052.00 CSR MOVE FRRT R@XIT 2
2053.00 CSR MOVE FRDVAL fT 40
2054.00 CSR MOVE FRVAL A@XIT 40
2055.00 CSR MOVE FRLVAL L@XIT 40
2056.00 CSR MOVE FRUVAL U@XIT 40
2057.00 CSR MOVE FREDWR W@XIT 30
2058.00 CSR MOVE FRLR J@XIT 1
2059.00 CSR MOVE FRNNIX N@XIT 20
2060.00 CSR Z-1DD1 #@XIT 110
2061.00 CSR MOVE F@XZT #A
2062.00 CSR DO #A
2063.00 CSR MULT 10 #@XIT
2064.00 CSR END
2065.00 CSR END
2066.00 C* -----
2067.00 C*
2068.00 C* Dictionary parameters for - gnanntity On Hand
2069.00 C*
2070.00 CSR MOVE *BLANK FRDTAI
2071.00 CSR MOVE 'XQT' FRDTAI
2072.00 CSR CALL 'X9800E' 81
2073.00 C* -----
2074.00 CSR PARM I9800E
2075.00 CSR FRERR IFEQ '0'
2076.00 CSR MOVE FRDSCR B@XOT 40
2077.00 CSR MOVE FRDTAT T@XQT 1
2078.00 CSR MOVE FREC E@XQT 1
2079.00 CSR MOVE FRDTAS C@XQT 50
2080.00 CSR MOVE FRDTAD G@XQT 20
2081.00 CSR MOVE FRCDEC F@XQT 1
2082.00 CSR MOVE LFRSY @SXQT 4

```

```

2083.00    CSR                MOVE FRRT      RXQT      2
2084.00    CSR                MOVE FRDVAL     D@XQT     40
2085.00    CSR                MOVE FRVAL      A@XQT     40
2086.00    CSR                MOVE FRLVAL     L@XQT     40
2087.00    CSR                MOVE FRUVAL     U@XQT     40
2088.00    CSR                MOVE FREDWR     W@XQT     30
2089.00    CSR                MOVE FRLR       J@XQT      1
2090.00    CSR                MOVE FRNNIX     N@XQT     20
2091.00    CSR                Z-ADD1         #@XQT     110
2092.00    CSR                MOVE F@XQT      #A
2093.00    CSR                DO #A
2094.00    CSR                MULT 10         #@XQT
2095.00    CSR                END
2096.00    CSR                END
2097.00    C*-----
2098.00    C*
2099.00    C*      Dictionary parameters for - Item Type
2100.00    C*
2101.00    CSR                MOVE *BLANK      FRDTAI
2102.00    CSR                MOVE 'XTY'      FRDTAI
2103.00    CSR                CALL 'X9800E'
2104.00    C*                -----
2105.00    CSR                PARM              I9800E
2106.00    CSR      FRERR    IFEQ '0'
2107.00    CSR                MOVE FRDSCR     B@XTY     40
2108.00    CSR                MOVE FRDTAT     T@XTY      1
2109.00    CSR                MOVE FREC       E@XTY      1
2110.00    CSR                MOVE FRDTAS     C@XTY     50
2111.00    CSR                MOVE FRDTAT     G@XTY     20
2112.00    CSR                MOVE FRCDEC     F@XTY      1
2113.00    CSR                MOVE LFRSY      S@XTY      4
2114.00    CSR                MOVE FRRT       R@XTY      2
2115.00    CSR                MOVE FRDVAL     D@XTY     40
2116.00    CSR                MOVE FRVAL      A@XTY     40
2117.00    CSR                MOVE FRLVAL     L@XTY     40
2118.00    CSR                MOVE FRUVAL     U@XTY     40
2119.00    CSR                MOVE FREDWR     W@XTY     30
2120.00    CSR                MOVE FRLR       J@XTY      1
2121.00    CSR                MOVE FRNNIX     N@XTY     20
2122.00    CSR                Z-ADD1         #@XTY     110
2123.00    CSR                MOVE F@XTY      #A
2124.00    CSR                DO #A
2125.00    CSR                MULT 10         #@XTY
2126.00    CSR                END
2127.00    CSR                END
2128.00    C*-----
2129.00    C*
2130.00    C*      Dictionary parameters for - Item Unit of Measure
2131.00    C*
2132.00    CSR                MOVE *BLANK      FRDTAI
2133.00    CSR                MOVE 'XUM'      FRDTAI
2134.00    CSR                CALL 'X9800E'
2135.00    C*                -----
2136.00    CSR                PARM              I9800E
2137.00    CSR      FRERR    IFEQ '0'
2138.00    CSR                MOVE FRDSCR     B@XUM     40
2139.00    CSR                MOVE FRDTAT     T@XUM      1
2140.00    CSR                MOVE FREC       E@XUM      1
2141.00    CSR                MOVE FRDTAS     C@XUM     50
2142.00    CSR                MOVE FRDTAD     G@XUM     20
2143.00    CSR                MOVE FRCDEC     F@XUM      1
2144.00    CSR                MOVE LFRSY      S@XUM      4
2145.00    CSR                MOVE FRRT       R@XUM      2
2146.00    CSR                MOVE FRDVAL     D@XDM     40
2147.00    CSR                MOVE FRVAL      A@XUM     40
2148.00    CSR                MOVE FRLVAL     L@XUM     40
2149.00    CSR                MOVE FRUVAL     U@XUM     40
2150.00    CSR                MOVE FREDWR     W@XUM     30
2151.00    CSR                MOVE FRLR       J@XUM      1
2152.00    CSR                MOVE FRNNIX     N@XUM     20
2153.00    CSR                Z-ADD1         #@XUM     110
2154.00    CSR                MOVE F@XUM      #A
2155.00    CSR                DO #A
2156.00    CSR                MULT 10         #@XUM
2157.00    CSR                END
2158.00    CSR                END
2159.00    C*-----

```

```

2160.00 C*
2161.00 C* Dictionary parameters for - Item Category Code 001
2162.00 C*
2163.00 CSR MOVE *BLANK FRDTAI
2164.00 CSR MOVE 'X001' FRDTAI
2165.00 CSR CALL 'X9800E' 81
2166.00 C*
2167.00 CSR PARM I9800E
2168.00 CSR FRERR IFEQ '0'
2169.00 CSR MOVE FRDSCR B@X001 40
2170.00 CSR MOVE FRDTAT T@X001 1
2171.00 CSR MOVE FREC E@X001 1
2172.00 CSR MOVE FRDTAS C@X001 50
2173.00 CSR MOVE FRDTAD G@X001 20
2174.00 CSR MOVE FRCDEC F@X001 1
2175.00 CSR MOVE FRST S@X001 4
2176.00 CSR MOVE FRRT R@X001 2
2177.00 CSR MOVE FRDVAL D@X001 40
2178.00 CSR MOVE FRVAL A@X001 40
2179.00 CSR MOVE FRLVAL L@X001 40
2180.00 CSR MOVE FRDVAL U@X001 40
2181.00 CSR MOVE FREDWR W@X001 30
2182.00 CSR MOVE FRLR J@X001 1
2183.00 CSR MOVE FRNNIX N@X001 20
2184.00 CSR Z-ADD1 #@X001 110
2185.00 CSR MOVE F@X001 #A
2186.00 CSR DO #A
2187.00 CSR MULT 10 #@X001
2188.00 CSR END
2189.00 CSR END
2190.00 C*-----
2191.00 C*
2192.00 C* Dictionary parameters for - Item Category Code 002
2193.00 C*
2194.00 CSR MOVE *BLANK FRDTAI
2195.00 CSR MOVE 'X002' FRDTAI
2196.00 CSR CALL 'X9800E' 81
2197.00 C*
2198.00 CSR PARM I9800E
2199.00 CSR FRERR IFEQ '0'
2200.00 CSR MOVE FRDSCR B@X002 40
2201.00 CSR MOVE FRDTAT T@X002 1
2202.00 CSR MOVE FREC E@X002 1
2203.00 CSR MOVE FRDTAS C@X002 50
2204.00 CSR MOVE FRDTAD G@X002 20
2205.00 CSR MOVE FRCDEC F@X002 1
2206.00 CSR MOVE FRST S@X002 4
2207.00 CSR MOVE FRRT R@X002 2
2208.00 CSR MOVE FRDVAL D@X002 40
2209.00 CSR MOVE FRVAL A@X002 40
2210.00 CSR MOVE FRLVAL L@X002 40
2211.00 CSR MOVE FRDVAL U@X002 40
2212.00 CSR MOVE FREDWR W@X002 30
2213.00 CSR MOVE FRLR J@X002 1
2214.00 CSR MOVE FRNNIX N@X002 20
2215.00 CSR Z-ADD1 #@X002 110
2216.00 CSR MOVE F@X002 #A
2217.00 CSR DO #A
2218.00 CSR MULT 10 #@X002
2219.00 CSR END
2220.00 CSR END
2221.00 C*-----
2222.00 C*
2223.00 C* Dictionary parameters for - Item Category Code 003
2224.00 C*
2225.00 CSR MOVE *BLANK FRDTAI
2226.00 CSR MOVE 'X003' FRDTAI
2227.00 CSR CALL 'X9800E' 81
2228.00 C*
2229.00 CSR PARM I9800E
2230.00 CSR FRERR IFEQ '0'
2231.00 CSR MOVE FRDSCR B@X003 40
2232.00 CSR MOVE FRDTAT T@X003 1
2233.00 CSR MOVE FREC E@X003 1
2234.00 CSR MOVE FRDTAS C@X003 50
2235.00 CSR MOVE FRDTAD G@X003 20
2236.00 CSR MOVE FRCDEC F@X003 1

```



```

2237.00  CSR      MOVEFRSY      S@X003      4
2238.00  CSR      MOVE FRRT      R@X003      2
2239.00  CSR      MOVE FRDVAL     D@X003     40
2240.00  CSR      MOVE FRVAL      A@X003     40
2241.00  CSR      MOVE FRLVAL     L@X003     40
2242.00  CSR      MOVE FRDVAL     U@X003     40
2243.00  CSR      MOVE FREDWR     W@X003     30
2244.00  CSR      MOVE FRLR       J@X003      1
2245.00  CSR      MOVE FRNNIX     N@X003     20
2246.00  CSR      Z-ADD1         #@X003    110
2247.00  CSR      MOVE F@X003     #A
2248.00  CSR      DO      #A
2249.00  CSR      MULT 10         #@X003
2250.00  CSR      END
2251.00  CSR      END
2252.00  C*-----
2253.00  C*
2254.00  C*      Dictionary parameters for - Item Category Code 004
2255.00  C*
2256.00  CSR      MOVE *BLANK      FRDTAI
2257.00  CSR      MOVE 'X004'      FRDTAI
2258.00  CSR      CALL 'X9800E'
2259.00  C*-----
2260.00  CSR      PARM              I9800E
2261.00  CSR      IFEQ '0'
2262.00  CSR      FRERR      MOVE FRDSCR     B@X004     40
2263.00  CSR      MOVE FRDTAT     T@X004      1
2264.00  CSR      MOVE FREC       E@X004      1
2265.00  CSR      MOVE FRDTAS     C@X004     50
2266.00  CSR      MOVE FRDTAD     G@X004     20
2267.00  CSR      MOVE FRCDEC     F@X004      1
2268.00  CSR      MOVEFRSY      S@X004      4
2269.00  CSR      MOVE FRRT      R@X004      2
2270.00  CSR      MOVE FRDVAL     D@X004     40
2271.00  CSR      MOVE FRVAL      A@X004     40
2272.00  CSR      MOVE FRLVAL     L@X004     40
2273.00  CSR      MOVE FRDVAL     U@X004     40
2274.00  CSR      MOVE FREDWR     W@X004     30
2275.00  CSR      MOVE FRLR       J@X004      1
2276.00  CSR      MOVE FRNNIX     N@X004     20
2277.00  CSR      Z-ADD1         #@X004    110
2278.00  CSR      MOVE F@X004     #A
2279.00  CSR      DO      #A
2280.00  CSR      MULT 10         #@X004
2281.00  CSR      END
2282.00  CSR      END
2283.00  C*-----
2284.00  C*
2285.00  C*      Dictionary parameters for - Item Category Code 005
2286.00  C*
2287.00  CSR      MOVE *BLANK      FRDTAI
2288.00  CSR      MOVE 'X005'      FRDTAI
2289.00  CSR      CALL 'X9800E'
2290.00  C*-----
2291.00  CSR      PARM              I9800E
2292.00  CSR      IFEQ '0'
2293.00  CSR      FRERR      MOVE FRDSCR     B@X005     40
2294.00  CSR      MOVE FRDTAT     T@X005      1
2295.00  CSR      MOVE FREC       E@X005      1
2296.00  CSR      MOVE FRDTAS     C@X005     50
2297.00  CSR      MOVE FRDTAD     G@X005     20
2298.00  CSR      MOVE FRCDEC     F@X005      1
2299.00  CSR      MOVEFRSY      S@X005      4
2300.00  CSR      MOVE FRRT      R@X005      2
2301.00  CSR      MOVE FRDVAL     D@X005     40
2302.00  CSR      MOVE FRVAL      A@X005     40
2303.00  CSR      MOVE FRLVAL     L@X005     40
2304.00  CSR      MOVE FRDVAL     U@X005     40
2305.00  CSR      MOVE FREDWR     W@X005     30
2306.00  CSR      MOVE FRLR       J@X005      1
2307.00  CSR      MOVE FRNNIX     N@X005     20
2308.00  CSR      Z-ADD1         #@X005    110
2309.00  CSR      MOVE F@X005     #A
2310.00  CSR      DO      #A
2311.00  CSR      MULT 10         #@X005
2312.00  CSR      END
2313.00  CSR      END

```

```

2314.00 C*
2315.00 C*
2316.00 C*   Set subroutine execution flag.
2317.00 C*
2318.00 CSR           MOVE '1'           $998           1
2319.00 C*
2320.00 CSR           END998           ENDSR
2321.00 C*****
2322.00 C*
2323.00 C*   SUBROUTINE S999- Housekeeping
2324.00 C*
2325.00 C*
2326.00 C*   Processing:  1.   Load video screen text.
2327.00 C*                  2.   Retrieve screen title data area, test
2328.00 C*                      for unauthorized access, center video
2329.00 C*                      title and save to video screen.
2330.00 C*                  3.   Initialize key list.
2331.00 C*                  4.   Load roll keys.
2332.00 C*                  5.   Passed parameters.
2333.00 C*                  6.   Load error message array.
2334.00 C*
2335.00 CSR           S999           BEGSR
2336.00 C*           -----
2337.00 C*
2338.00 C*   Required program parameters.
2339.00 C*
2340.00 CSR           *ENTRY           PLIST
2341.00 C*
2342.00 C*   Passed Parameter - Item ID
2343.00 C*
2344.00 CSR           PARM           $$XIT           8
2345.00 C*
2346.00 C*   Move to internal reference - Item ID
2347.00 C*
2348.00 CSR           MOVE ##XIT           vDXIT
2349.00 C*
2350.00 C*   Test for auto inquiry function.
2351.00 C*
2352.00 CSR           VDXIT           IFNE *BLANK
2353.00 CSR           MOVE '1'           $AUTO           1
2354.00 CSR           END
2355.00 C*-----
2356.00 C*
2357.00 C*   Load video screen text
2358.00 C*
2359.00 CSR           MOVE@@FILE           PSKEY           10
2360.00 CSR           Z-ADD025           PSVTX#           30
2361.00 C/COPY JDECPY,C00SC
2362.00 C*-----
2363.00 C*
2364.00 C*   Key list for - Cost Center Security
2365.00 C*
2366.00 CSR           MSKY01           KLIST
2367.00 CSR           KFLD           MSUSER
2368.00 CSR           KFLD           MSFILE
2369.00 CSR           KFLD           MSMCUT
2370.00 C*-----
2371.00 C*
2372.00 C*   Key list for -SDM Item Master File
2373.00 C*
2374.00 CSR           ZXKY01           KLIST
2375.00 CSR           KFLD           QXXIT
2376.00 C*-----
2377.00 C*
2378.00 C*   Load roll key uppr and lower key values.
2379.00 C*
2380.00 CSR           *LIKE           DEFM QXXIT           $RUKEY
2381.00 CSR           *LIKE           DEFN $RUKEY           $RDKEY
2382.00 CSR           MOVE *LOVAL           $RUKEY
2383.00 CSR           MOVE *ALL'9'           $RDKEY
2384.00 C*-----
2385.00 C*

```

Assures S998 will only be executed once

Parameters passed to program

Set auto-inquiry if information is passed

Retrieves vocabulary overrides

Only loads these VTX fields displayed on the video instead of all 144

Composite keys are defined here

Using \*LIKE more and more, especially for work fields.

```

2386.00 C*          Load error messages array.
2387.00 C*
2388.00 CSR          MOVE '0001'      EMK,01      Inv Action
2389.00 CSR          MOVE '0002'      EMK,02      Inv Key
2390.00 CSR          MOVE '0003'      EMK,03      Inv Blanks
2391.00 CSR          MOVE '0004'      EMK,04      Inv Date
2392.00 CSR          MOVE '0005'      EMK,05      Inv Next Nbr
2393.00 CSR          MOVE '0007'      EMK,06      In Use
2394.00 CSR          MOVE '0025'      EMK,07      Inv Values
2395.00 CSR          MOVE '0026'      EMK,08      Inv MCU
2396.00 CSR          MOVE '0027'      EMK,09      Inv Desc Ttl
2397.00 CSR          MOVE '0052'      EMK,10
2398.00 C*-----
2399.00 C*
2400.00 C*          Load invalid action code array.
2401.00 C*
2402.00 CSR          MOVEA '          ' @NAC
2403.00 C*-----
2404.00 C*
2405.00 Ct          Load system date.
2406.00 C*
2407.00 CSR          TIME $WRK12 120
2408.00 CSR          MOVE $WRK12      $$EDT      60
2409.00 CSR          MOVE $$EDT      $SIDAT      6
2410.00 CSR          MOVE ' *SYSVAL ' #FFMT      7
2411.00 CSR          MOVE *BLANKS     #EDAT      8
2412.00 CSR          MOVE ' *JUL      ' #TFMT      7
2413.00 CSR          MOVE ' *NONE     ' #SKP      7
2414.00 CSR          MOVE ' '         $ERTST     1
2415.00 CSR          CALL 'X0028      '
2416.00 C*-----
2417.00 CSR          PARM              $SIDAT
2418.00 CSR          PARM              #EDAT
2419.00 CSR          PARM              #FFMT
2420.00 CSR          PARM              #TFMT
2421.00 CSR          PARM              #SKP
2422.00 CSR          PARM              $ERTST
2423.00 CSR          MOVE #SIDAT      $$UPMJ      60
2424.00 C*-----
2425.00 CSR          END999      ENDSR
2426.00 C*****
2427.00 C*****
2428.00 0I92801 E          UNLOCK

```

Error message numbers from Data Dictionary

Lockout action code function used with the Program Generator

Use the TIME feature to allow for all date formats

Method of releasing master file record locks



## Appendix E – J.D. Edwards Subroutines and Flows

---

### Internal RPG Subroutines Within J.D. Edwards Programs

- Standard names make program maintenance easier.
- Called primarily from Mainline.

The table below describes internal RPG subroutines within J.D. Edwards programs:

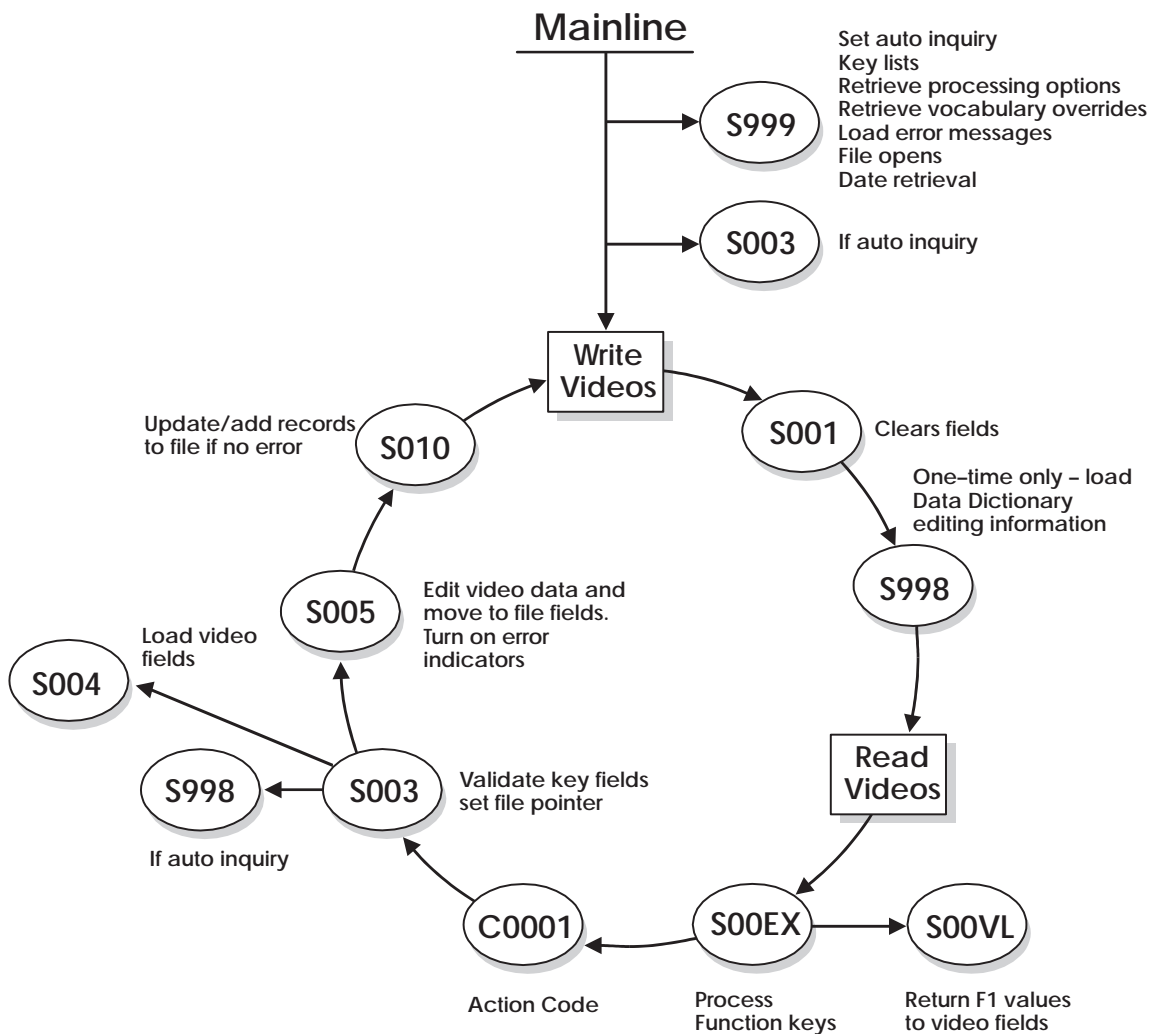
Routine	Description
S00EX	Processes all function key exits. <ul style="list-style-type: none"><li>• Calls P9601H if F24 was pressed</li><li>• Calls X96CCX if F1 was pressed</li><li>• Calls subroutine S00VL if F1 was pressed after X96CCX was called</li><li>• Calls P0000E if F7 was pressed</li><li>• Calls P00HELP if the HELP key was pressed</li><li>• Calls subroutine S001 if F22 was pressed</li><li>• Calls all programs to process all user defined function keys</li></ul>
S00VL	Values returned with Cursor Sensitive Help.  Is called from the subroutine S00EX after the program X96CCX is called
S00OP	Subfile Selection Exits (Options).
S001	Clears all database and video fields. <ul style="list-style-type: none"><li>• Usually only clears key fields and VC0 fields if F22 (Clear) is pressed</li></ul>
S002	Checks for level breaks for reports. <ul style="list-style-type: none"><li>• Turns on level break flags.</li><li>• Retrieves total line description</li></ul>

Routine	Description
S003	Validates the key fields. Calls S998 subroutine if auto inquire was invoked Sets the file pointer. <ul style="list-style-type: none"><li>• Performs a SETLL and CHAIN if a single record maintenance program</li><li>• Performs a SETLL for subfile programs</li></ul> Calls a subroutine S004 to load video/report fields Monitors for no subfile records loaded if a subfile Loads unused subfile records with blanks
S004	Display/load video/report fields.
S005	Scrubs and edits video/report fields. <ul style="list-style-type: none"><li>• Moves video data to database fields</li><li>• Turns on error indicators if a field is in error</li><li>• Updates/writes records to the database file if a subfile</li><li>• Updates the subfile</li></ul>
S010	For reports with level breaks it: <ul style="list-style-type: none"><li>• Prints the total</li><li>• Clears the level break totals</li><li>• Prints the grand total (if it has reached the end of the file)</li><li>• Prints the detail</li><li>• Adds to the new level break totals</li></ul> Calls subroutine S020 if it is a report with subheadings If it is <i>not</i> a report, it updates, adds, or deletes records from the database file <ul style="list-style-type: none"><li>• Turns on F22 (Clear) to force S001 to be executed to clear the buffer before reading another record.</li></ul>
S020	Print Report Subheadings.
S998	Loads Data Dictionary values. (One time only) <ul style="list-style-type: none"><li>• Retrieves row description for level breaks and subheadings, if applicable</li></ul>

<b>Routine</b>	<b>Description</b>
S999	Housekeeping. (One time only) <ul style="list-style-type: none"><li>• Sets auto inquiry</li><li>• Defines key lists</li><li>• Retrieves processing options and level breaks, if applicable</li><li>• Retrieves vocabulary overrides</li><li>• Loads error messages</li><li>• Performs file opens</li><li>• Current date retrieval</li><li>• Work fields defined using *LIKE</li><li>• Prints cover page and Helps in a report</li><li>• Performed only one time</li></ul>

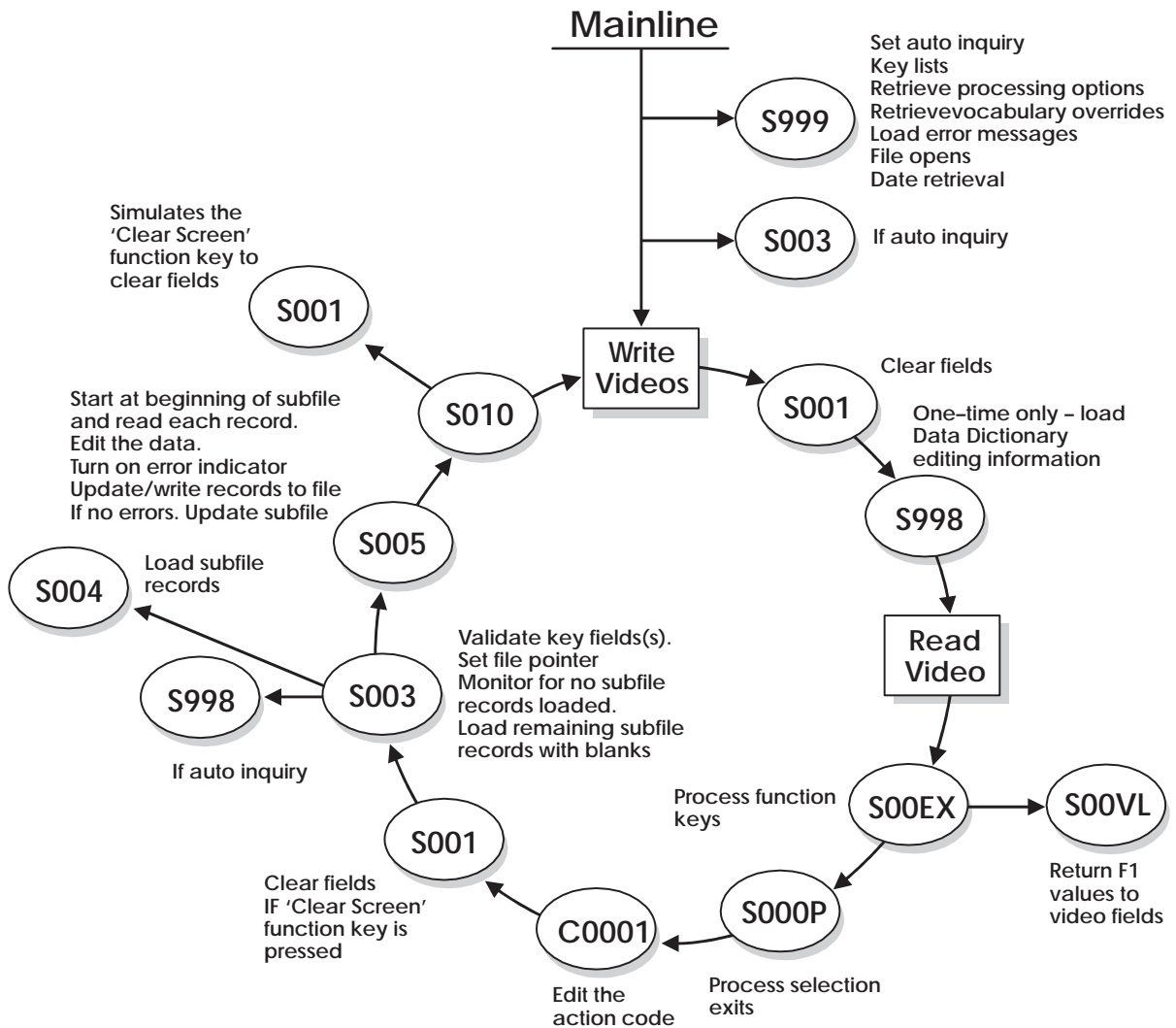
---

## Interactive Non-Subfile Program

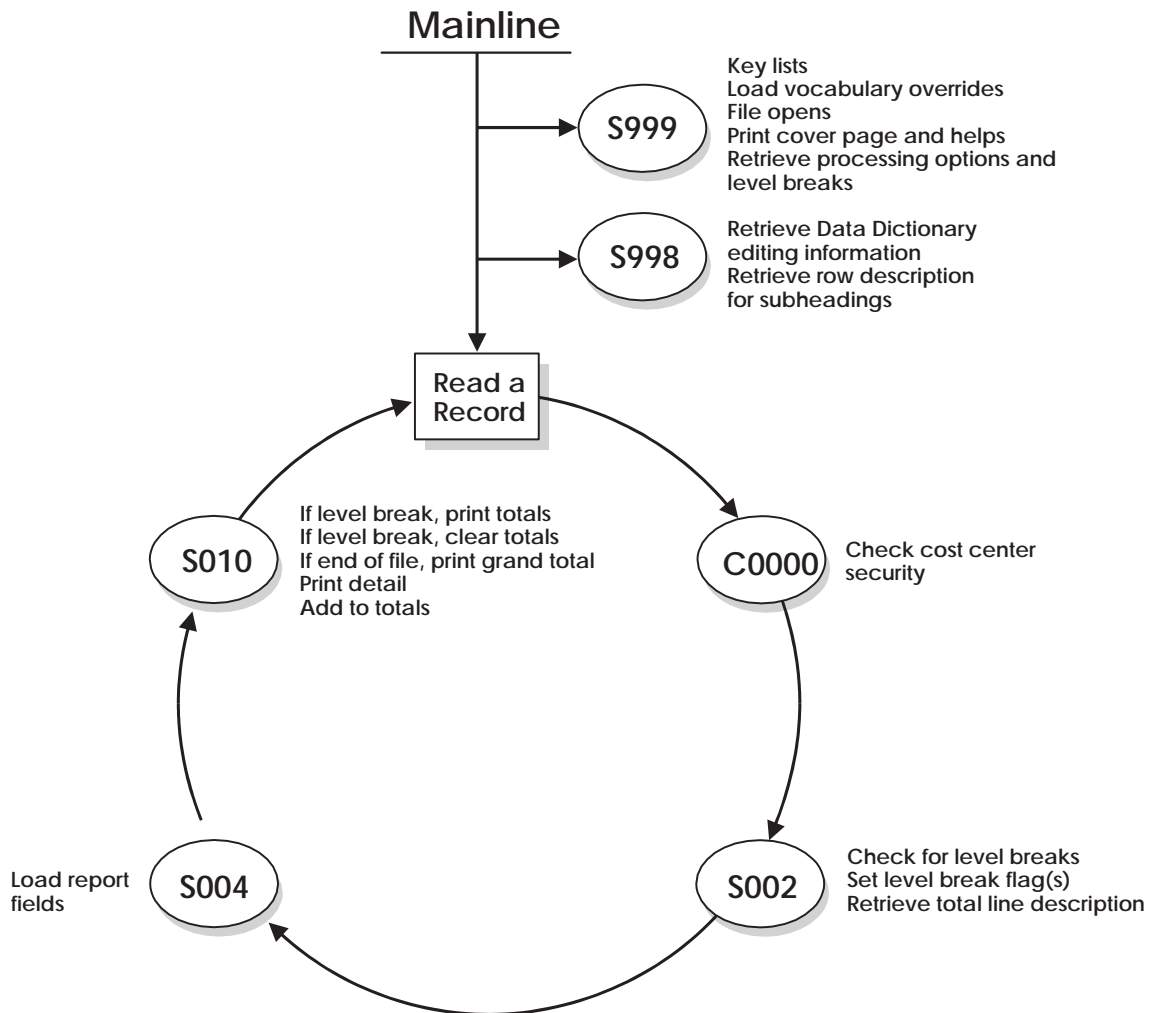




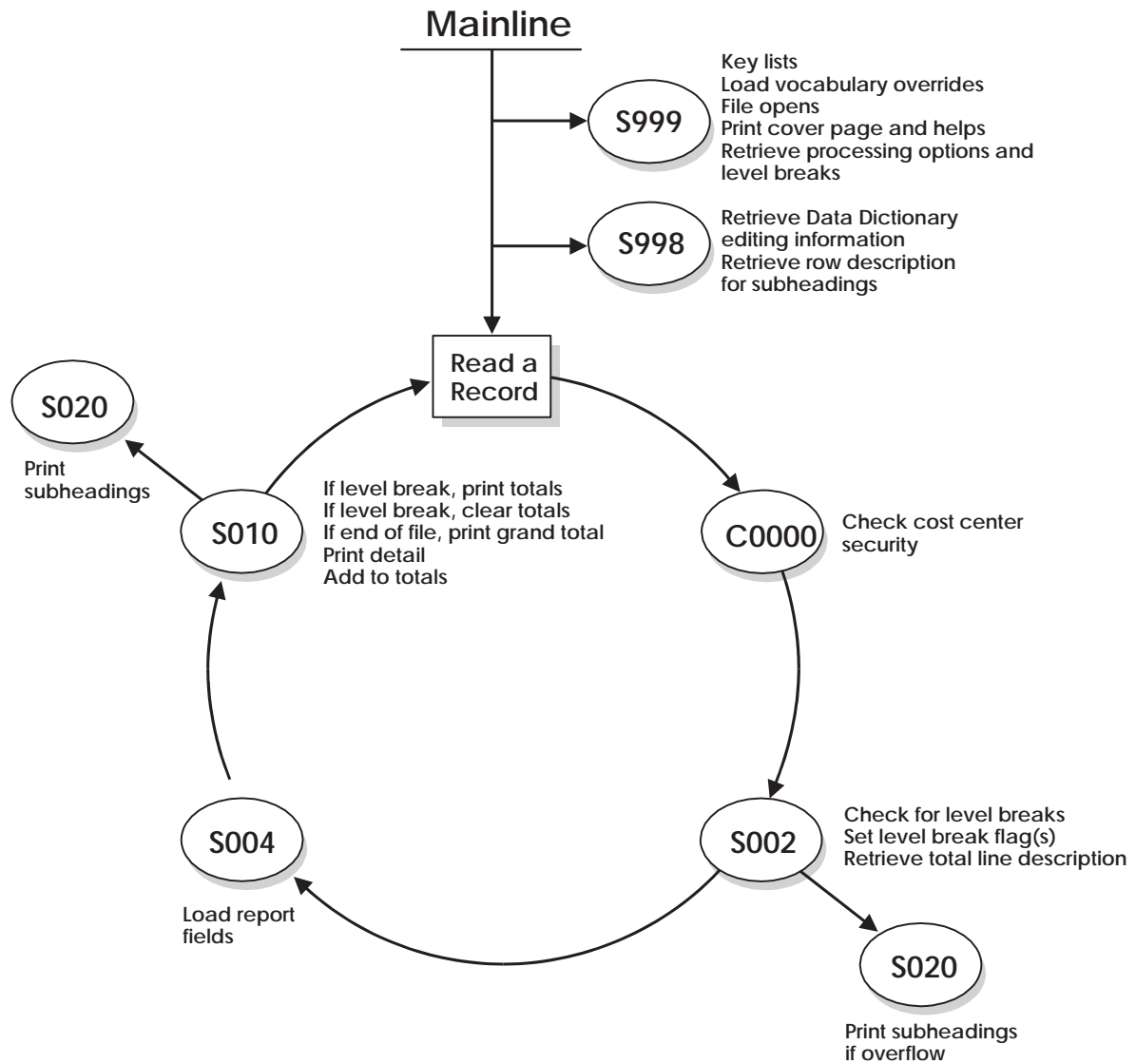
## Subfile Program With Selection Exits



## Report Program Without Subheadings



## Report Program With Subheadings





## Appendix F – Putting It All Together

### Program Type B0010

```
93001                                Create/Modify Program Types
Action Code. . . . . I
Program Type . . . . . B0010      STD/M  - Action Code
Seq  Prim Modul Glossary K
1.00 FILEDEFN01      File Specification
2.00 FILEEXTN0      Tables & Arrays      - STD Video
3.00 INPUT1          Data Structures     - STD Video
4.00 MAINLINE        Mainline           - Video
5.00 S00EX-1         Exits Subroutine    - STD Video
6.00 S00OP           Options Subroutine
6.50 S00VL-1         Return Values Subr  - Standard
7.00 S001-1          Clear Subroutine    - STD Video
8.00 S003-1          Edit Key            - STD Video
9.00 S004-1          Load Display Subr   - STD Video
10.00 S005-1         Edit Subroutine     - STD Video
11.00 S010-1         Update Subroutine    - STD Video
12.00 S999-1         Housekeeping Subr   - STD Video
_____
_____                                F24=More
```

- These are all the pieces required to create program type B0010.

The following pages will show the basic shell for this program type and we will follow through the shell to see how the generator will create the RPG source code.

## Program Type B0010

```
R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
TITLEH/TITLE
H* -----
H*
H*      Copyright (c) 1994
H*      J. D. Edwards & Company
H*
H*      This unpublished material is proprietary to
H*      J. D. Edwards & Company. All rights reserved.
H*      The methods and techniques described herein are
H*      considered trade secrets and/or confidential.
H*      Reproduction or distribution, in whole or in part,
H*      is forbidden except by express written permission
H*      of J. D. Edwards & Company.
H* -----
F*
F*      PROGRAM REVISION LOG
F* -----
F*
F*      Date      Programmer      Nature of Revision
F* -----
AUTHRF*          SAR #          (AS/400  A/G)
F*
DESC F*
F*
F* *****
F*
FILESF*
COPY F* *****
E* *****
E*      PROGRAM TABLES AND ARRAYS
E* -----
E*
E*      EMK      64  4      Error Msg
E*      @MK      64  1      Error Msg
E*      @ER      64  4      Error Msg
E*      @DV      40  1      Dflt Wrk
E*      @C      256 1      Literal Work
COPY E*
I* *****
I*      PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES
I* -----
I*
I*      Data Structure to Load Video Screen Text
I*
VTS IDSTXT      DS
VTX I          1  40 VTX001
VTX I          41  80 VTX002
VTX I          81 120 VTX003
VTX I          121 160 VTX004
VTX I          161 200 VTX005
VTX I          201 240 VTX006
VTX I          241 280 VTX007
VTX I          281 320 VTX008
VTX I          321 360 VTX009
VTX I          361 400 VTX010
VTX I          401 440 VTX011
VTX I          441 480 VTX012
VTX I          481 520 VTX013
R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
VTX I          521 560 VTX014
VTX I          561 600 VTX015
VTX I          601 640 VTX016
VTX I          641 680 VTX017
VTX I          681 720 VTX018
VTX I          721 760 VTX019
VTX I          761 800 VTX020
VTX I          801 840 VTX021
VTX I          841 880 VTX022
VTX I          881 920 VTX023
VTX I          921 960 VTX024
VTX I          9611000 VTX025
VTX I          10011040 VTX026
VTX I          10411080 VTX027
VTX I          10811120 VTX028
VTX I          11211160 VTX029
VTX I          11611200 VTX030
VTX I          12011240 VTX031
VTX I          12411280 VTX032
VTX I          12811320 VTX033
VTX I          13211360 VTX034
VTX I          13611400 VTX035
VTX I          14011440 VTX036
VTX I          14411480 VTX037
VTX I          14811520 VTX038
VTX I          15211560 VTX039
VTX I          15611600 VTX040
VTX I          16011640 VTX041
FILEDEFN01 001000000000
FILEDEFN01 002000000000
FILEDEFN01 003000000000
FILEDEFN01 004000000000
FILEDEFN01 005000000000
FILEDEFN01 006000000000
FILEDEFN01 007000000000
FILEDEFN01 008000000000
FILEDEFN01 009000000000
FILEDEFN01 010000000000
FILEDEFN01 011000000000
FILEDEFN01 012000000000
FILEDEFN01 013000000000
FILEDEFN01 014000000000
FILEDEFN01 015000000000
FILEDEFN01 016000000000
FILEDEFN01 017000000000
FILEDEFN01 018000000000
FILEDEFN01 019000000000
FILEDEFN01 020000000000
FILEDEFN01 021000000000
FILEDEFN01 022000000000
FILEDEFN01 023000000000
FILEDEFN01 024000000000
FILEDEFN01 025000000000
FILEDEFN01 026000000000
FILEDEFN01 027000000000
FILEDEFN01 028000000000
FILEDEFN01 029000000000
FILEEXTN0 001000000000
FILEEXTN0 002000000000
FILEEXTN0 003000000000
FILEEXTN0 004000000000
FILEEXTN0 005000000000
FILEEXTN0 006000000000
FILEEXTN0 007000000000
FILEEXTN0 008000000000
FILEEXTN0 011100000000
FILEEXTN0 012000000000
INPUT1 001000000000
INPUT1 002000000000
INPUT1 003000000000
INPUT1 004000000000
INPUT1 005000000000
INPUT1 006000000000
INPUT1 007000000000
INPUT1 008000000000
INPUT1 009000000000
INPUT1 010000000000
INPUT1 011000000000
INPUT1 012000000000
INPUT1 013000000000
INPUT1 014000000000
INPUT1 015000000000
INPUT1 016000000000
INPUT1 017000000000
INPUT1 018000000000
INPUT1 019000000000
INPUT1 020000000000
INPUT1 021000000000
INPUT1 022000000000
INPUT1 023000000000
INPUT1 024000000000
INPUT1 025000000000
INPUT1 026000000000
INPUT1 027000000000
INPUT1 028000000000
INPUT1 029000000000
INPUT1 030000000000
INPUT1 031000000000
INPUT1 032000000000
INPUT1 033000000000
INPUT1 034000000000
INPUT1 035000000000
INPUT1 036000000000
INPUT1 037000000000
INPUT1 038000000000
INPUT1 039000000000
INPUT1 040000000000
INPUT1 041000000000
INPUT1 042000000000
INPUT1 043000000000
INPUT1 044000000000
INPUT1 045000000000
INPUT1 046000000000
INPUT1 047000000000
INPUT1 048000000000
```

VTX	I	16411680	VTX042	INPUT1	049000000000
VTX	I	16811720	VTX043	INPUT1	050000000000
VTX	I	17211760	VTX044	INPUT1	051000000000
VTX	I	17611800	VTX045	INPUT1	052000000000
VTX	I	18011840	VTX046	INPUT1	053000000000
VTX	I	18411880	VTX047	INPUT1	054000000000
VTX	I	18811920	VTX048	INPUT1	055000000000
VTX	I	19211960	VTX049	INPUT1	056000000000
VTX	I	19612000	VTX050	INPUT1	057000000000
VTX	I	20012040	VTX051	INPUT1	058000000000
VTX	I	20412080	VTX052	INPUT1	059000000000
VTX	I	20812120	VTX053	INPUT1	060000000000
VTX	I	21212160	VTX054	INPUT1	061000000000
VTX	I	21612200	VTX055	INPUT1	062000000000
VTX	I	22012240	VTX056	INPUT1	063000000000
VTX	I	22412280	VTX057	INPUT1	064000000000
VTX	I	22812320	VTX058	INPUT1	065000000000
VTX	I	23212360	VTX059	INPUT1	066000000000
VTX	I	23612400	VTX060	INPUT1	067000000000
VTX	I	24012440	VTX061	INPUT1	068000000000
VTX	I	24412480	VTX062	INPUT1	069000000000
VTX	I	24812520	VTX063	INPUT1	070000000000
VTX	I	25212560	VTX064	INPUT1	071000000000
VTX	I	25612600	VTX065	INPUT1	072000000000
VTX	I	26012640	VTX066	INPUT1	073000000000
VTX	I	26412680	VTX067	INPUT1	074000000000
VTX	I	26812720	VTX068	INPUT1	075000000000
VTX	I	27212760	VTX069	INPUT1	076000000000
VTX	I	27612800	VTX070	INPUT1	077000000000
VTX	I	28012840	VTX071	INPUT1	078000000000
VTX	I	28412880	VTX072	INPUT1	079000000000

R93950	B0010	- STD/M	- Action Code		DATE - 2/02/94
VTX I		28812920	VTX073	INPUT1	080000000000
VTX I		29212960	VTX074	INPUT1	081000000000
VTX I		29613000	VTX075	INPUT1	082000000000
VTX I		30013040	VTX076	INPUT1	083000000000
VTX I		30413080	VTX077	INPUT1	084000000000
VTX I		30813120	VTX078	INPUT1	085000000000
VTX I		31213160	VTX079	INPUT1	086000000000
VTX I		31613200	VTX080	INPUT1	087000000000
VTX I		32013240	VTX081	INPUT1	088000000000
VTX I		32413280	VTX082	INPUT1	089000000000
VTX I		32813320	VTX083	INPUT1	090000000000
VTX I		33213360	VTX084	INPUT1	091000000000
VTX I		33613400	VTX085	INPUT1	092000000000
VTX I		34013440	VTX086	INPUT1	093000000000
VTX I		34413480	VTX087	INPUT1	094000000000
VTX I		34813520	VTX088	INPUT1	095000000000
VTX I		35213560	VTX089	INPUT1	096000000000
VTX I		35613600	VTX090	INPUT1	097000000000
VTX I		36013640	VTX091	INPUT1	098000000000
VTX I		36413680	VTX092	INPUT1	099000000000
VTX I		36813720	VTX093	INPUT1	100000000000
VTX I		37213760	VTX094	INPUT1	101000000000
VTX I		37613800	VTX095	INPUT1	102000000000
VTX I		38013840	VTX096	INPUT1	103000000000
VTX I		38413880	VTX097	INPUT1	104000000000
VTX I		38813920	VTX098	INPUT1	105000000000
VTX I		39213960	VTX099	INPUT1	106000000000
VTX I		39614000	VTX100	INPUT1	107000000000
VTX I		40014040	VTX101	INPUT1	108000000000
VTX I		40414080	VTX102	INPUT1	109000000000
VTX I		40814120	VTX103	INPUT1	110000000000
VTX I		41214160	VTX104	INPUT1	111000000000
VTX I		41614200	VTX105	INPUT1	112000000000
VTX I		42014240	VTX106	INPUT1	113000000000
VTX I		42414280	VTX107	INPUT1	114000000000
VTX I		42814320	VTX108	INPUT1	115000000000
VTX I		43214360	VTX109	INPUT1	116000000000
VTX I		43614400	VTX110	INPUT1	117000000000
VTX I		44014440	VTX111	INPUT1	118000000000
VTX I		44414480	VTX112	INPUT1	119000000000
VTX I		44814520	VTX113	INPUT1	120000000000
VTX I		45214560	VTX114	INPUT1	121000000000
VTX I		45614600	VTX115	INPUT1	122000000000
VTX I		46014640	VTX116	INPUT1	123000000000
VTX I		46414680	VTX117	INPUT1	124000000000
VTX I		46814720	VTX118	INPUT1	125000000000
VTX I		47214760	VTX119	INPUT1	126000000000
VTX I		47614800	VTX120	INPUT1	127000000000
VTX I		48014840	VTX121	INPUT1	128000000000
VTX I		48414880	VTX122	INPUT1	129000000000
VTX I		48814920	VTX123	INPUT1	130000000000
VTX I		49214960	VTX124	INPUT1	131000000000
VTX I		49615000	VTX125	INPUT1	132000000000
VTX I		50015040	VTX126	INPUT1	133000000000
VTX I		50415080	VTX127	INPUT1	134000000000
VTX I		50815120	VTX128	INPUT1	135000000000
VTX I		51215160	VTX129	INPUT1	136000000000
VTX I		51615200	VTX130	INPUT1	137000000000
VTX I		52015240	VTX131	INPUT1	138000000000



```

R93950                                B0010      - STD/M      - Action Code                                DATE - 2/02/94
VTX I                                52415280 VTX132                                INPUT1 139000000000
VTX I                                52815320 VTX133                                INPUT1 140000000000
VTX I                                53215360 VTX134                                INPUT1 141000000000
VTX I                                53615400 VTX135                                INPUT1 142000000000
VTX I                                54015440 VTX136                                INPUT1 143000000000
VTX I                                54415480 VTX137                                INPUT1 144000000000
VTX I                                54815520 VTX138                                INPUT1 145000000000
VTX I                                55215560 VTX139                                INPUT1 146000000000
VTX I                                55615600 VTX140                                INPUT1 147000000000
VTX I                                56015640 VTX141                                INPUT1 148000000000
VTX I                                56415680 VTX142                                INPUT1 149000000000
VTX I                                56815720 VTX143                                INPUT1 150000000000
VTX I                                57215760 VTX144                                INPUT1 151000000000
I*                                     INPUT1 152000000000
I/COPY JDECPY,I00DSINX                INPUT1 153000000000
I/COPY JDECPY,I00PS@@                INPUT1 153100000000
INFDSI/COPY JDECPY,I00DSPROG          INPUT1 154000000000
DATESI*                                INPUT1 155000000000
COPY I*                                INPUT1 156000000000
C*****                                MAINLINE 001000000000
C*      MAINLINE PROGRAM                MAINLINE 002000000000
C*      -----                        MAINLINE 003000000000
C*                                     MAINLINE 004000000000
C*      Process housekeeping.            MAINLINE 005000000000
C*                                     MAINLINE 006000000000
C*                                     EXSR S999                MAINLINE 007000000000
C*      -----                        MAINLINE 008000000000
C*                                     MAINLINE 009000000000
C*      If LR on, end program.            MAINLINE 010000000000
C*                                     MAINLINE 011000000000
C*      *INLR      CABEQ'1'      EOJ      MAINLINE 012000000000
C*      -----      ---
C*                                     MAINLINE 013000000000
C*                                     MAINLINE 014000000000
C*      If automatic inquiry set, process inquiry. MAINLINE 015000000000
C*                                     MAINLINE 016000000000
C*      $AUTO      CASEQ'1'      S003      24                MAINLINE 017000000000
C*      -----      ---
C*                                     MAINLINE 018000000000
C*      END                                MAINLINE 019000000000
C*      Begin normal program processing.  MAINLINE 020000000000
C*      -----                        MAINLINE 021000000000
C*                                     MAINLINE 022000000000
C*      *INLR      DOWEQ'0'              MAINLINE 023000000000
C*                                     MAINLINE 024000000000
/*                                     MAINLINE 025000000000
/*      If #SFRNO field, do subfile record number validation MAINLINE 026000000000
/*                                     MAINLINE 027000000000
+FLDNC*      #SFRNO      ZSFLRCDNO        MAINLINE 028000000000
/*                                     MAINLINE 029000000000
/*      If SFLCLR is used, process *in38 accordingly        MAINLINE 030000000000
/*                                     MAINLINE 031000000000
+FLDNC*      ?SFLCLR      ZSFLCLR          MAINLINE 032000000000
C*                                     MAINLINE 033000000000
C*      Write video screen.                MAINLINE 034000000000
C*                                     MAINLINE 035000000000
/*                                     MAINLINE 036000000000
/*      If not a subfile display, just write format1        MAINLINE 037000000000
/*                                     MAINLINE 038000000000
-FLDNC*      ?SFL      ZWRITE              MAINLINE 039000000000
/*                                     MAINLINE 040000000000

```

```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
/*              If a subfile display, write format1 and formatc
/*
+FLDNC*          ?SFL              ZWRITESFL
C              MOVE '1'          @@AID
C              EXSR S001
C              ----
C*              Load data field dictionary parameters (one cycle only).
C*
C              $998      CASEQ' '      S998
C              ----
C              END
C*              Begin video screen read processing.
C*
C              SETOF                      999301
DSPF C          READ &01FILE          9998
C              Z-ADD0          ##RROW
C              Z-ADD0          ##RCOL
C*              If video read timed out, end program.
C*
C              *IN99      CABEQ'1'      EOJ      LR
C              ----
C              @@AID      CABEQ#FEOJ      EOJ      LR
C              ----
C*              If valid function key pressed, process and return.
C*
C              *IN15      IFEQ '1'
C              EXSR S00EX
C              ----
C              *INLR      CABEQ'1'      EOJ
C              ----
C              *IN15      CABEQ'1'      END
C              ----
C              END
/*
/*              If any selection exits, exsr S000P
/*
+DTAIC*          SELC              ZS000P
/*
/*              If action code then exsr C0001
/*
+FLDNC*          ACTION          ZACTION
C*
C*              Load subfile records.
C*
C              EXSR S003
C              ----
C*
/*              If any update files then exsr S005
/*
/*
+FILEC*          *ANY      DB      ZS005      @
/*
/*              If any update files and action code then do S010
/*
+FILEC*          *ANY      DB      *AND      @
-FILEC*          *ANY      DB      *AND      2

```

```

MAINLINE 041000000000
MAINLINE 042000000000
MAINLINE 043000000000
MAINLINE 044000000000
MAINLINE 045000000000
MAINLINE 046000000000
MAINLINE 047000000000
MAINLINE 048000000000
MAINLINE 049000000000
MAINLINE 050000000000
MAINLINE 051000000000
MAINLINE 052000000000
MAINLINE 053000000000
MAINLINE 054000000000
MAINLINE 055000000000
MAINLINE 056000000000
MAINLINE 057000000000
MAINLINE 058000000000
MAINLINE 059000000000
MAINLINE 060000000000
MAINLINE 061000000000
MAINLINE 062000000000
MAINLINE 063000000000
MAINLINE 064000000000
MAINLINE 065000000000
MAINLINE 066000000000
MAINLINE 067000000000
MAINLINE 068000000000
MAINLINE 069000000000
MAINLINE 070000000000
MAINLINE 071000000000
MAINLINE 072000000000
MAINLINE 073000000000
MAINLINE 074000000000
MAINLINE 075000000000
MAINLINE 076000000000
MAINLINE 077000000000
MAINLINE 078000000000
MAINLINE 079000000000
MAINLINE 080000000000
MAINLINE 081000000000
MAINLINE 082000000000
MAINLINE 083000000000
MAINLINE 084000000000
MAINLINE 085000000000
MAINLINE 086000000000
MAINLINE 087000000000
MAINLINE 088000000000
MAINLINE 089000000000
MAINLINE 090000000000
MAINLINE 091000000000
MAINLINE 092000000000
MAINLINE 093000000000
MAINLINE 094000000000
MAINLINE 095000000000
MAINLINE 096000000000
MAINLINE 097000000000
MAINLINE 098000000000
MAINLINE 098500000000

```

```

R93950          B0010      - STD/M  - Action Code          DATE - 2/02/94
+FLDNC*          ACTION    ZS010A
/*
/*      If a Master File 2 exists, then do S011.
/*
+FILEC*          *ANY      DB    *AND      @
+FILEC*          *ANY      DB    *AND      2
+FLDNC*          ACTION    ZS011
C*
C*      Return for next input.
C*
C*          END          TAG
C*          ---          ---
C*
C*      Set correct message in line 24.
C*
C*          *IN93      IFEQ '1'
C*                      MOVELSVL24E      VDL24
C*                      ELSE
C*                      MOVELSVL24M      VDL24
C*                      END
C*
C*                      END
C*
C*          EOJ          TAG
C*          ---          ---
C*
C*      END MAINLINE PROGRAM
C*      -----
COPY C*****
C*
C*      SUBROUTINE S00EX - Process Function Keys
C*      -----
C*
C*      Processing:  1. Determine function key pressed.
C*                  2. Process function key request.
C*
CSR      S00EX      BEGSR
C*      -----
+FLDNC*      #SFRNO      Z@@SRCN
CSR      T00EXA      TAG
C*      -----
C*
C*      If EOJ requested, exit subroutine.
C*
CSR      @@AID      CABEQ#FEOJ      ENDEXE      LR
C*      -----
C*
C*      If Display Keys pressed, exit to help facility and return.
C*      -----
CSR      @@AID      IFEQ #FKEYS
CSR      CALL 'P9601H'      98
C*      -----
CSR      PARM      I00SC
CSR      PARM      SRVFDS
CSR      PARM      I00CSR
C*
CSR      @@AID      CABNE#FKEYS      T00EXA
C*      -----

```

```

MAINLINE 099000000000
MAINLINE 100000000000
MAINLINE 101000000000
MAINLINE 102000000000
MAINLINE 103000000000
MAINLINE 103500000000
MAINLINE 104000000000
MAINLINE 105000000000
MAINLINE 106000000000
MAINLINE 107000000000
MAINLINE 108000000000
MAINLINE 109000000000
MAINLINE 110000000000
MAINLINE 111000000000
MAINLINE 112000000000
MAINLINE 113000000000
MAINLINE 114000000000
MAINLINE 115000000000
MAINLINE 116000000000
MAINLINE 117000000000
MAINLINE 118000000000
MAINLINE 119000000000
MAINLINE 120000000000
MAINLINE 121000000000
MAINLINE 122000000000
MAINLINE 123000000000
MAINLINE 124000000000
MAINLINE 125000000000
MAINLINE 126000000000
S00EX-1 001000000000
S00EX-1 002000000000
S00EX-1 003000000000
S00EX-1 004000000000
S00EX-1 005000000000
S00EX-1 006000000000
S00EX-1 007000000000
S00EX-1 008000000000
S00EX-1 009000000000
S00EX-1 009500000000
S00EX-1 010000000000
S00EX-1 011000000000
S00EX-1 012000000000
S00EX-1 013000000000
S00EX-1 014000000000
S00EX-1 015000000000
S00EX-1 016000000000
S00EX-1 017000000000
S00EX-1 018000000000
S00EX-1 019000000000
S00EX-1 020000000000
S00EX-1 021000000000
S00EX-1 022000000000
S00EX-1 023000000000
S00EX-1 024000000000
S00EX-1 025000000000
S00EX-1 026000000000
S00EX-1 027000000000
S00EX-1 028000000000
S00EX-1 029000000000

```

```
R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
CSR              GOTO ENDEXE
C*              -----
CSR              END
C*
C*      If Cursor Sensitive Help Pressed, exit to CS Help.
C*      -----
CSR              @@AID      IFEQ #FQMRK
CSR              MOVEA*IN      ##IN
CSR              CALL 'X96CCX'          98
C*              -----
CSR              PARM              I00SC
CSR              PARM              SRVFDS
CSR              PARM              I00CSR
CSR              PARM ' '          ##CCFF 2
CSR              PARM              I00MDE
C*
CSR              ##FLDN      IFNE *BLANKS
CSR              EXSR S00VL
C*              -----
CSR              MOVEA##IN      *IN,1
CSR              END
CSR              MOVEL*BLANKS      ##DTAI
CSR              GOTO ENDEXE
C*              -----
CSR              END
C*
C*      If Display errors pressed, exit to error messages.
C*      -----
CSR              @@AID      IFEQ #FERRD
CSR              Z-ADD1      #G
CSR              Z-ADD1      #H
CSR              #G          DOWLE64
CSR              @MK,#G      IFEQ '1'
CSR              MOVE EMK,#G      @ER,#H
CSR              ADD 1          #H
CSR              END
CSR              ADD 1          #G
CSR              END
CSR              CALL 'P0000E'          98
C*              -----
CSR              PARM              @ER
CSR              GOTO ENDEXE
C*              -----
CSR              END
C*
C*      If HELP key pressed, exit to help facility and return.
C*      -----
CSR              @@AID      IFEQ #FHHELP
CSR              CALL 'P00HELP'          98
C*              -----
CSR              PARM              HS@@
CSR              PARM              HE@@
CSR              PARM              I00SC
CSR              PARM              SRVFDS
CSR              PARM              I00CSR
CSR              GOTO ENDEXE
S00EX-1      030000000000
S00EX-1      031000000000
S00EX-1      032000000000
S00EX-1      033000000000
S00EX-1      034000000000
S00EX-1      035000000000
S00EX-1      036000000000
S00EX-1      037000000000
S00EX-1      038000000000
S00EX-1      039000000000
S00EX-1      040000000000
S00EX-1      041000000000
S00EX-1      042000000000
S00EX-1      043000000000
S00EX-1      044000000000
S00EX-1      044100000000
S00EX-1      045000000000
S00EX-1      046000000000
S00EX-1      047000000000
S00EX-1      048000000000
S00EX-1      049000000000
S00EX-1      050000000000
S00EX-1      051000000000
S00EX-1      052000000000
S00EX-1      053000000000
S00EX-1      054000000000
S00EX-1      055000000000
S00EX-1      056000000000
S00EX-1      057000000000
S00EX-1      058000000000
S00EX-1      059000000000
S00EX-1      060000000000
S00EX-1      061000000000
S00EX-1      062000000000
S00EX-1      063000000000
S00EX-1      064000000000
S00EX-1      065000000000
S00EX-1      066000000000
S00EX-1      067000000000
S00EX-1      068000000000
S00EX-1      069000000000
S00EX-1      070000000000
S00EX-1      071000000000
S00EX-1      072000000000
S00EX-1      073000000000
S00EX-1      074000000000
S00EX-1      075000000000
S00EX-1      076000000000
S00EX-1      077000000000
S00EX-1      078000000000
S00EX-1      079000000000
S00EX-1      080000000000
S00EX-1      081000000000
S00EX-1      082000000000
S00EX-1      083000000000
S00EX-1      084000000000
S00EX-1      085000000000
S00EX-1      086000000000
S00EX-1      087000000000
```

R93950		B0010	- STD/M	- Action Code	DATE - 2/02/94
C*		-----			S00EX-1 088000000000
CSR		END			S00EX-1 089000000000
C*					S00EX-1 090000000000
C*		If Clear screen pressed, clear screen and return.			S00EX-1 091000000000
C*		-----			S00EX-1 092000000000
C*					S00EX-1 093000000000
CSR	@@AID	IFEQ #FCLR			S00EX-1 094000000000
CSR		EXSR S001			S00EX-1 095000000000
C*		-----			S00EX-1 096000000000
CSR		GOTO ENDEXE			S00EX-1 097000000000
C*		-----			S00EX-1 098000000000
EXITCSR		END			S00EX-1 099000000000
C*					S00EX-1 100000000000
C*		Process roll up and down keys.			S00EX-1 101000000000
C*		-----			S00EX-1 102000000000
C*					S00EX-1 103000000000
CSR	@@AID	IFEQ #FROLU			S00EX-1 104000000000
CSR	@@AID	OREQ #FROLD			S00EX-1 105000000000
CSR	\$SECUR	DOUEQ ' '			S00EX-1 107000000000
CSR		MOVE ' ' \$SECUR 1			S00EX-1 108000000000
C*					S00EX-1 109000000000
C*		If ROLL UP key pressed, process read next.			S00EX-1 110000000000
C*		-----			S00EX-1 111000000000
C*					S00EX-1 112000000000
CSR	@@AID	IFEQ #FROLU			S00EX-1 113000000000
C*					S00EX-1 114000000000
C*		Reset error indicators if roll			S00EX-1 115000000000
C*					S00EX-1 116000000000
CSR		MOVEA\$RESET *IN,41	818299		S00EX-1 117000000000
CSR		MOVE '0' *IN,40			S00EX-1 118000000000
CSR		SETOF			S00EX-1 119000000000
MF CSR	%	READ &01FORMAT	9981		S00EX-1 120000000000
CSR	*IN81	IFEQ '1'			S00EX-1 121000000000
MF CSR	\$RUKEY	SETLL&01FORMAT			S00EX-1 122000000000
CSR		SETOF	8299		S00EX-1 123000000000
MF CSR	%	READ &01FORMAT	9982		S00EX-1 124000000000
C*					S00EX-1 125000000000
C*		If error on read, set error.			S00EX-1 126000000000
C*					S00EX-1 127000000000
CSR	*IN82	IFEQ '1'			S00EX-1 128000000000
CSR		SETON	9341		S00EX-1 129000000000
CSR		MOVE '1' @MK,2			S00EX-1 130000000000
CSR		GOTO ENDEXE			S00EX-1 131000000000
C*		-----			S00EX-1 132000000000
CSR		END			S00EX-1 133000000000
CSR		END			S00EX-1 134000000000
CSR		END			S00EX-1 135000000000
C*					S00EX-1 136000000000
C*		If ROLL DOWN key pressed, process read prior.			S00EX-1 137000000000
C*		-----			S00EX-1 138000000000
C*					S00EX-1 139000000000
CSR	@@AID	IFEQ #FROLD			S00EX-1 140000000000
C*					S00EX-1 141000000000
C*		Reset error indicators if roll			S00EX-1 142000000000
C*					S00EX-1 143000000000
CSR		MOVEA\$RESET *IN,41			S00EX-1 144000000000
CSR		MOVE '0' *IN,40			S00EX-1 145000000000
CSR		SETOF	818299		S00EX-1 146000000000
MF CSR	%	READP&01FORMAT	9981		S00EX-1 147000000000

```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
CSR             *IN81      IFEQ '1'
MF CSR          $RDKEY      SETLL&01FORMAT
CSR             SETOF              8299
MF CSR          %          READP&01FORMAT          9982
C*
C*      If error on read, set error.
C*
CSR             *IN82      IFEQ '1'
CSR             SETON              9341
CSR             MOVE '1'          @MK,2
CSR             GOTO ENDEXE
C*      -----
CSR             END
CSR             END
CSR             END
C*
C*      Load video screen data on roll keys.
C*      -----
CSR             @@AID      IFEQ #FROLU
CSR             @@AID      OREQ #FROLD
/*
/*      Include record lock logic if update files exist.
/*
+FILEC*          *ANY      DB      ZUNLOCK      @
C*
MCU01C*
MCU01C*      Cost Center security edit.
MCU01C*
MCU01CSR          MOVEL&01(FILE )#FILE
MCU01CSR          MOVEL&01KEY      #MCU
MCU01CSR          #AUT      IFNE '1'
MCU01CSR          #FAUT      ANDNE'1'
MCU01CSR          EXSR C0000
MCU01C*          -----
MCU01CSR          END
MCU01CSR          #AUT      IFNE '1'
MCU01CSR          #FAUT      ANDNE'1'
MCU01CSR          #MAUT      ANDNE'1'
MCU01CSR          MOVE '1'          $SECUR
MCU01CSR          END
CSR             $SECUR      CASEQ' '          S004
C*          -----
CSR             END
C*
CSR             END
C*
CSR             END
C*
CSR             END
CSR             GOTO ENDEXE
C*      -----
CSR             END
C*
CSR             @@AID      IFNE '1'
CSR             SETON              0193
CSR             GOTO ENDEXE
C*      -----
CSR             END
C*
CSR             ENDEXE      ENDSR

```

```

S00EX-1      148000000000
S00EX-1      149000000000
S00EX-1      150000000000
S00EX-1      151000000000
S00EX-1      152000000000
S00EX-1      153000000000
S00EX-1      154000000000
S00EX-1      155000000000
S00EX-1      156000000000
S00EX-1      157000000000
S00EX-1      158000000000
S00EX-1      159000000000
S00EX-1      160000000000
S00EX-1      161000000000
S00EX-1      162000000000
S00EX-1      163000000000
S00EX-1      164000000000
S00EX-1      165000000000
S00EX-1      166000000000
S00EX-1      167000000000
S00EX-1      168000000000
S00EX-1      169000000000
S00EX-1      169100000000
S00EX-1      169200000000
S00EX-1      169300000000
S00EX-1      169400000000
S00EX-1      169900000000
S00EX-1      170000000000
S00EX-1      171000000000
S00EX-1      172000000000
S00EX-1      173000000000
S00EX-1      174000000000
S00EX-1      175000000000
S00EX-1      176000000000
S00EX-1      177000000000
S00EX-1      178000000000
S00EX-1      179000000000
S00EX-1      180000000000
S00EX-1      181000000000
S00EX-1      182000000000
S00EX-1      183000000000
S00EX-1      184000000000
S00EX-1      185000000000
S00EX-1      186000000000
S00EX-1      187000000000
S00EX-1      188000000000
S00EX-1      189000000000
S00EX-1      190000000000
S00EX-1      191000000000
S00EX-1      192000000000
S00EX-1      193000000000
S00EX-1      194000000000
S00EX-1      195000000000
S00EX-1      196000000000
S00EX-1      197000000000
S00EX-1      198000000000
S00EX-1      199000000000
S00EX-1      200000000000
S00EX-1      201000000000

```

```

R93950                                B0010      - STD/M      - Action Code                                DATE - 2/02/94
COPY C*****
/*
/*      If the display file has the selection option field,
/*      include the S00OP subroutine to process selection options.
/*
+FLDNC**      VDSELC      *AND
-FLDNC**      SFSELC      S00OP-1
/*
+FLDNC**      SFSELC      S00OP-2
C*
C*      SUBROUTINE S00VL - Cursor Control Return Values
C*      -----
C*
C*      By format, find the field to update and move in the
C*      returned value. If the format is a subfile, the record
C*      to change is found in @@RRN.
C*
CSR          S00VL      BEGSR
C*      -----
C*
CSR          ##RVAL      IFEQ ' *BLANK'
CSR          MOVE *BLANK      ##RVAL
CSR          END
S00VLC*
C*
CSR          END0VL      ENDSR
COPY C*****
C*
C*      SUBROUTINE S001 - Clear Fields
C*      -----
C*
C*      Processing: 1. Reset all video screen and data file fields
C*                  for next transaction.
C*                  2. Clear action code only if requested.
C*
CSR          S001      BEGSR
C*      -----
C*
C*      Reset fields for next transaction.
MF CSR          *NOKEY      CLEAR&01FORMAT
CLRY C*
CSR          MOVE SVL24M      VDL24
CSR          MOVE ' '      @IN37 1
C*
C*      Clear action code only if clear screen action.
C*
CSR          @@AID      IFEQ #FCLR
CSR          MOVE *ALL'0'      $RESET
CSR          MOVEA$RESET      *IN,41
CSR          MOVE ' '      ACTION 1
CLRN C*
CSR          END
C*
CSR          END001      ENDSR
COPY C*****
C*
C*      SUBROUTINE S003 - Edit Key
C*      -----

```

```

S00EX-1      202000000000
S00OP        000100000000
S00OP        000200000000
S00OP        000300000000
S00OP        000400000000
S00OP        001000000000
S00OP        001100000000
S00OP        001200000000
S00OP        001300000000
S00VL-1      001000000000
S00VL-1      002000000000
S00VL-1      003000000000
S00VL-1      004000000000
S00VL-1      005000000000
S00VL-1      006000000000
S00VL-1      007000000000
S00VL-1      008000000000
S00VL-1      009000000000
S00VL-1      010000000000
S00VL-1      011000000000
S00VL-1      012000000000
S00VL-1      013000000000
S00VL-1      014000000000
S00VL-1      015000000000
S00VL-1      016000000000
S00VL-1      017000000000
S00VL-1      018000000000
S001-1       001000000000
S001-1       002000000000
S001-1       003000000000
S001-1       004000000000
S001-1       005000000000
S001-1       006000000000
S001-1       007000000000
S001-1       008000000000
S001-1       009000000000
S001-1       010000000000
S001-1       011000000000
S001-1       012000000000
S001-1       013000000000
S001-1       013100000000
S001-1       014000000000
S001-1       015000000000
S001-1       016000000000
S001-1       017000000000
S001-1       018000000000
S001-1       019000000000
S001-1       020000000000
S001-1       021000000000
S001-1       022000000000
S001-1       023000000000
S001-1       024000000000
S001-1       025000000000
S001-1       026000000000
S001-1       027000000000
S001-1       028000000000
S003-1       001000000000
S003-1       002000000000
S003-1       003000000000

```

```
R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
C*
C*      Processing:  1.  Clear error indicators and arrays.
C*                  2.  Load input keys.
C*                  3.  Validate master file key.
C*                  4.  Release master file record lock.
C*                  5.  Load video screen output on inquiry.
C*
CSR          S003      BEGSR
C*          ----      -----
C*
C*      Load data field dictionary parameters (one cycle only).
C*
CSR          $998      CASEQ' '      S998
C*          ----      -----
CSR          END
C*
C*      Reset error indicators and arrays.
C*
CSR          MOVE *ALL'0'      $RESET 39
CSR          MOVE *BLANK      $REST1 63
CSR          MOVEA$RESET      *IN,41
CSR          MOVEA$REST1      @MK,2
CSR          CLEAR@ER
C*-----
KEYS C*
C*-----
MF CSR          CHAIN&01FORMAT      9899
MCU01C*
MCU01C*      Cost Center security edit.
MCU01C*
MCU01CSR          MOVE&01(FILE )#FILE
MCU01CSR          MOVE&01KEY      #MCU
MCU01CSR          #AUT      IFNE '1'
MCU01CSR          #FAUT      ANDNE'1'
MCU01CSR          EXSR C0000
MCU01C*          ---- -----
MCU01CSR          END
MCU01CSR          #AUT      IFNE '1'
MCU01CSR          #FAUT      ANDNE'1'
MCU01CSR          #MAUT      ANDNE'1'
MCU01CSR          MOVE '1'      $$SECR 1
MCU01CSR          END
C*
C*      If security violation, set error condition.
C*
CSR          $$SECR      IFEQ '1'
CSR          MOVE '1'      @MK,8
CSR          SETON
CSR          MOVE ' '      $$SECR 1      9341
CSR          GOTO END003
C*          ---- -----
CSR          END
C*
C*      Edit result of read and action code.
C*
CSR          *IN98      IFEQ '1'
CSR          *IN21      COMP '0'      41 *error*
CSR          ELSE
CSR          *IN21      COMP '1'      41 *error*
S003-1      004000000000
S003-1      005000000000
S003-1      006000000000
S003-1      007000000000
S003-1      008000000000
S003-1      009000000000
S003-1      010000000000
S003-1      011000000000
S003-1      012000000000
S003-1      012100000000
S003-1      012200000000
S003-1      012300000000
S003-1      012400000000
S003-1      012500000000
S003-1      012600000000
S003-1      013000000000
S003-1      014000000000
S003-1      015000000000
S003-1      016000000000
S003-1      016100000000
S003-1      017000000000
S003-1      018000000000
S003-1      019000000000
S003-1      020000000000
S003-1      021000000000
S003-1      022000000000
S003-1      023000000000
S003-1      024000000000
S003-1      025000000000
S003-1      026000000000
S003-1      027000000000
S003-1      028000000000
S003-1      029000000000
S003-1      030000000000
S003-1      031000000000
S003-1      032000000000
S003-1      033000000000
S003-1      034000000000
S003-1      035000000000
S003-1      036000000000
S003-1      037000000000
S003-1      038000000000
S003-1      039000000000
S003-1      040000000000
S003-1      041000000000
S003-1      042000000000
S003-1      043000000000
S003-1      044000000000
S003-1      045000000000
S003-1      046000000000
S003-1      047000000000
S003-1      048000000000
S003-1      049000000000
S003-1      050000000000
S003-1      051000000000
S003-1      052000000000
S003-1      053000000000
S003-1      054000000000
S003-1      055000000000
```



```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
CSR              END
C*
C*      If indicator 41 on, invalid key for action code.
C*
CSR      *IN41      IFEQ '1'
CSR              MOVE '1'          @MK,2
CSR              SETON              93
CSR              END
C*
C*      If indicator 99 on, record in use.
C*
CSR      *IN99      IFEQ '1'
CSR              CALL 'P98RLCK'      81
C*              -----
CSR              PARM              ##PSDS
CSR              MOVE '1'          @MK,6
CSR              SETON              9341
CSR              END
C*-----
C*
C*      If not inquiry, skip remainder of subroutine.
C*
CSR      *IN24      CABEQ'0'      END003
C*              -----
C*-----
C*
C*      Release record lock on master file.
C*
CSR      *IN98      IFEQ '0'
CSR      *IN99      ANDEQ'0'
CSR              EXCPTUNLOCK
CSR              END
C*
C*      If errors, skip remainder of subroutine.
C*
CSR      *IN93      CABEQ'1'      END003
C*              -----
C*-----
C*
C*      Move data base information to video screen.
C*
CSR              EXSR S004
C*              -----
C*-----
CSR      END003      ENDSR
COPY C*****
C*
C*      SUBROUTINE S004 - Load Video Screen Data
C*      -----
C*
C*      Processing: 1. Move data base information to video screen.
C*                  All video screen fields are alpha and
C*                  therefore numeric information must be
C*                  processed through subroutine C0014 to set
C*                  proper decimals and provide editing for
C*                  display on screen.
C*
C*                  Date fields must be converted from their
C*                  internal format of month, day and year or

```

```

S003-1 056000000000
S003-1 057000000000
S003-1 058000000000
S003-1 059000000000
S003-1 060000000000
S003-1 061000000000
S003-1 062000000000
S003-1 063000000000
S003-1 064000000000
S003-1 065000000000
S003-1 066000000000
S003-1 067000000000
S003-1 067100000000
S003-1 067200000000
S003-1 067300000000
S003-1 068000000000
S003-1 069000000000
S003-1 070000000000
S003-1 071000000000
S003-1 072000000000
S003-1 073000000000
S003-1 074000000000
S003-1 075000000000
S003-1 076000000000
S003-1 077000000000
S003-1 078000000000
S003-1 079000000000
S003-1 079100000000
S003-1 079200000000
S003-1 080000000000
S003-1 081000000000
S003-1 081100000000
S003-1 082000000000
S003-1 083000000000
S003-1 084000000000
S003-1 085000000000
S003-1 086000000000
S003-1 087000000000
S003-1 088000000000
S003-1 089000000000
S003-1 090000000000
S003-1 091000000000
S003-1 092000000000
S003-1 093000000000
S003-1 094000000000
S003-1 095000000000
S004-1 001000000000
S004-1 002000000000
S004-1 003000000000
S004-1 004000000000
S004-1 005000000000
S004-1 006000000000
S004-1 007000000000
S004-1 008000000000
S004-1 009000000000
S004-1 010000000000
S004-1 011000000000
S004-1 012000000000
S004-1 013000000000

```

```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
C*              julian to the system format using program
C*              X0028.
C*
CSR            S004      BEGSR
C*            ----      -----
DSP1 C*
CSR            END004     ENDSR
COPY C*****
C*
C*      SUBROUTINE S005 - Scrub Input
C*      -----
C*
C*      Processing: 1.  Validate all video input.
C*                    All numeric fields must be processed
C*                    thru subroutines C0012 and C0015 in order
C*                    to scrub the alpha input field and convert
C*                    back to internal numeric representation of
C*                    15 digits and 0 decimals.
C*
C*                    Date fields must be converted from system
C*                    format to their internal format of month,
C*                    day and year or julian using program X0028.
C*      2.  Update data record fields from video.
C*
CSR            S005      BEGSR
C*            ----      -----
C*
C*      If not addition or change, bypass subroutine
C*
CSR            *IN21      IFEQ '0'
CSR            *IN22      ANDEQ '0'
CSR            GOTO END005
C*            ----      -----
CSR            END
C*
FIELD C*
CSR            END005     ENDSR
COPY C*****
C*
C*      SUBROUTINE S010 - Update Data Base
C*      -----
C*
C*      Processing: 1.  Update data base file based upon valid
C*                    action codes.
C*
CSR            S010      BEGSR
C*            ----      -----
AC*
AC*      If add action, add record.
AC*
ACSR          *IN21      IFEQ '1'
MF ACSR          %        WRITE&01FORMAT          99
ACSR          END
CC*
CC*      If change action, update record.
CC*
CCSR          *IN22      IFEQ '1'
MF CCSR          %        UPDAT&01FORMAT          99
CCSR          END

```

```

S004-1 014000000000
S004-1 015000000000
S004-1 016000000000
S004-1 017000000000
S004-1 018000000000
S004-1 025000000000
S004-1 026000000000
S004-1 027000000000
S005-1 001000000000
S005-1 002000000000
S005-1 003000000000
S005-1 004000000000
S005-1 005000000000
S005-1 006000000000
S005-1 007000000000
S005-1 008000000000
S005-1 009000000000
S005-1 010000000000
S005-1 011000000000
S005-1 012000000000
S005-1 013000000000
S005-1 014000000000
S005-1 015000000000
S005-1 016000000000
S005-1 017000000000
S005-1 018000000000
S005-1 019000000000
S005-1 020000000000
S005-1 021000000000
S005-1 022000000000
S005-1 023000000000
S005-1 024000000000
S005-1 025000000000
S005-1 026000000000
S005-1 028000000000
S005-1 029000000000
S005-1 030000000000
S005-1 031000000000
S010-1 001000000000
S010-1 002000000000
S010-1 003000000000
S010-1 004000000000
S010-1 005000000000
S010-1 006000000000
S010-1 007000000000
S010-1 008000000000
S010-1 009000000000
S010-1 010000000000
S010-1 011000000000
S010-1 012000000000
S010-1 013000000000
S010-1 014000000000
S010-1 015000000000
S010-1 016000000000
S010-1 017000000000
S010-1 018000000000
S010-1 019000000000
S010-1 020000000000
S010-1 021000000000

```

R93950	B0010	- STD/M	- Action Code	DATE - 2/02/94
DC*				S010-1 022000000000
DC*	If delete action, delete record.			S010-1 023000000000
DC*				S010-1 024000000000
DCSR	*IN23	IFEQ '1'		S010-1 025000000000
MF DCSR	%	DELET&01FORMAT	99	S010-1 026000000000
DCSR		END		S010-1 027000000000
C*				S010-1 028000000000
C*	Clear data field for next transaction			S010-1 029000000000
C*				S010-1 030000000000
CSR		MOVE #FCLR	@@AID	S010-1 031000000000
CSR		EXSR S001		S010-1 032000000000
C*		----	----	S010-1 033000000000
CSR	END010	ENDSR		S010-1 034000000000
COPY	C*****			S010-1 035000000000
C*				S999-1 001000000000
C*	SUBROUTINE S998	- Load dictionary parameters.		S999-1 002000000000
C*	-----			S999-1 003000000000
C*				S999-1 004000000000
CSR	S998	BEGSR		S999-1 005000000000
C*	----	----		S999-1 006000000000
DPARMC*				S999-1 007000000000
C*				S999-1 008000000000
C*	Set subroutine execution flag.			S999-1 009000000000
C*				S999-1 010000000000
CSR		MOVE '1'	\$998 1	S999-1 011000000000
C*				S999-1 012000000000
CSR	END998	ENDSR		S999-1 013000000000
C*	C*****			S999-1 014000000000
C*				S999-1 015000000000
C*	SUBROUTINE S999	- Housekeeping		S999-1 016000000000
C*	-----			S999-1 017000000000
C*				S999-1 018000000000
C*	Processing: 1.	Load video screen text.		S999-1 019000000000
C*	2.	Retrieve screen title data area, test		S999-1 020000000000
C*		for unauthorized access, center video		S999-1 021000000000
C*		title and move to video screen.		S999-1 022000000000
C*	3.	Initialize key list.		S999-1 023000000000
C*	4.	Load roll keys.		S999-1 024000000000
C*	5.	Passed parameters.		S999-1 025000000000
C*	6.	Load error message array.		S999-1 026000000000
C*				S999-1 027000000000
CSR	S999	BEGSR		S999-1 028000000000
C*	----	----		S999-1 029000000000
C*				S999-1 030000000000
C*	Required program parameters.			S999-1 031000000000
C*				S999-1 032000000000
ENTRYCSR	*ENTRY	PLIST		S999-1 033000000000
AUTOIC*				S999-1 034000000000
C*	C*-----			S999-1 035000000000
C*				S999-1 036000000000
C*	Load video screen text.			S999-1 037000000000
C*				S999-1 038000000000
CSR		MOVE@@FILE	PSKEY 10	S999-1 039000000000
VTXI	C*			S999-1 040000000000
	C/COPY JDECPY,C00SC			S999-1 041000000000
	C*-----			S999-1 042000000000
	/*			S999-1 043000000000
	/* If processing options exist, load processing options			S999-1 044000000000
	/*			S999-1 045000000000

# CASE – Computer Aided Software Engineering

```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/94
+FLDNC*          *OPTION    ZOPTIONX
KLISTC*
C*
C*      Load roll key upper and lower key values.
C*
MF  CSR          *LIKE      DEFN &01KEYFLD $RUKEY          S999-1      046000000000
CSR          *LIKE      DEFN $RUKEY $RDKEY          S999-1      047000000000
CSR          MOVE *LOVAL $RUKEY          S999-1      048000000000
CSR          MOVE *ALL'9' $RDKEY          S999-1      049000000000
C*-----
C*
C*      Load error messages array.
C*
CSR          MOVE '0001' EMK,01      Inv Action          S999-1      050000000000
CSR          MOVE '0002' EMK,02      Inv Key              S999-1      051000000000
CSR          MOVE '0003' EMK,03      Inv Blanks           S999-1      052000000000
CSR          MOVE '0004' EMK,04      Inv Date             S999-1      053000000000
CSR          MOVE '0005' EMK,05      Inv Next Nbr         S999-1      054000000000
CSR          MOVE '0007' EMK,06      In Use               S999-1      055000000000
CSR          MOVE '0025' EMK,07      Inv Values           S999-1      056000000000
CSR          MOVE '0026' EMK,08      Inv MCU              S999-1      057000000000
EMK CSR          MOVE '0027' EMK,09      Inv Desc Ttl      S999-1      058000000000
C*-----
C*
C*      Load invalid action code array.
C*
ACTN CSR          MOVEA'      '      @NAC          S999-1      059000000000
C*-----
C*
C*      Load system date.
C*
CSR          TIME          $WRK12 120          S999-1      060000000000
CSR          MOVE $WRK12 $SED 60          S999-1      061000000000
CSR          MOVE $SED #SIDAT 6          S999-1      062000000000
CSR          MOVE #SYSVAL 'FFMT 7          S999-1      063000000000
CSR          MOVE #BLANKS #EDAT 8          S999-1      064000000000
CSR          MOVE #JUL 'TFMT 7          S999-1      065000000000
CSR          MOVE #NONE 'SEP 7          S999-1      066000000000
CSR          MOVE ' ' #ERTST 1          S999-1      067000000000
CSR          CALL 'X0028 '          S999-1      069000000000
C*-----
CSR          PARM          #SIDAT          S999-1      070000000000
CSR          PARM          #EDAT          S999-1      071000000000
CSR          PARM          #FFMT          S999-1      072000000000
CSR          PARM          #TFMT          S999-1      073000000000
CSR          PARM          #SEP          S999-1      074000000000
CSR          PARM          $ERTST          S999-1      075000000000
CSR          MOVE #SIDAT $$UPMJ 60          S999-1      076000000000
C*-----
CSR          END999      ENDSR          S999-1      077000000000
C*****
/*
/*      If processing options exist, include copy module
/*
+FLDNC*          *OPTION    ZOPTIONC          S999-1      078000000000
COPY C*****          S999-1      079000000000
MF  O&01FMT E          UNLOCK          S999-1      080000000000
          S999-1      081000000000
          S999-1      082000000000
          S999-1      083000000000
          S999-1      084000000000
          S999-1      085000000000
          S999-1      086000000000
          S999-1      087000000000
          S999-1      088000000000
          S999-1      089000000000
          S999-1      090000000000
          S999-1      091000000000
          S999-1      092000000000
          S999-1      093000000000
          S999-1      094000000000
          S999-1      095000000000
          S999-1      096000000000
          S999-1      097000000000
          S999-1      098000000000
          S999-1      099000000000
          S999-1      100000000000
          S999-1      101000000000
          S999-1      102000000000
          S999-1      103000000000

```

## Appendix G – Functional Servers

---

Several J.D. Edwards programs access functional servers. The purpose of functional servers is to provide a central location for standard business rules about entering documents, such as vouchers, invoices, and journal entries. These business rules establish the following:

- Data dictionary default values
- Field edits and valid values
- Error processing
- Relationships between fields or applications

The advantages of a functional server are:

- It reduces maintenance of entry programs because edit rules reside in one central location.
- You can standardize documents across all applications because you create them using the same business rules.
- Generally, the user interface (appearance and interaction) of a screen is now separate from how a program works.

The steps for setting up business rules for an entry program are:

1. Create a DREAM Writer version for a specific functional server program (for example, XT0411Z1 for voucher entry).
2. Set the processing options within the version according to your company requirements.
3. Specify the version you want the entry program to use in the processing options for that entry program.

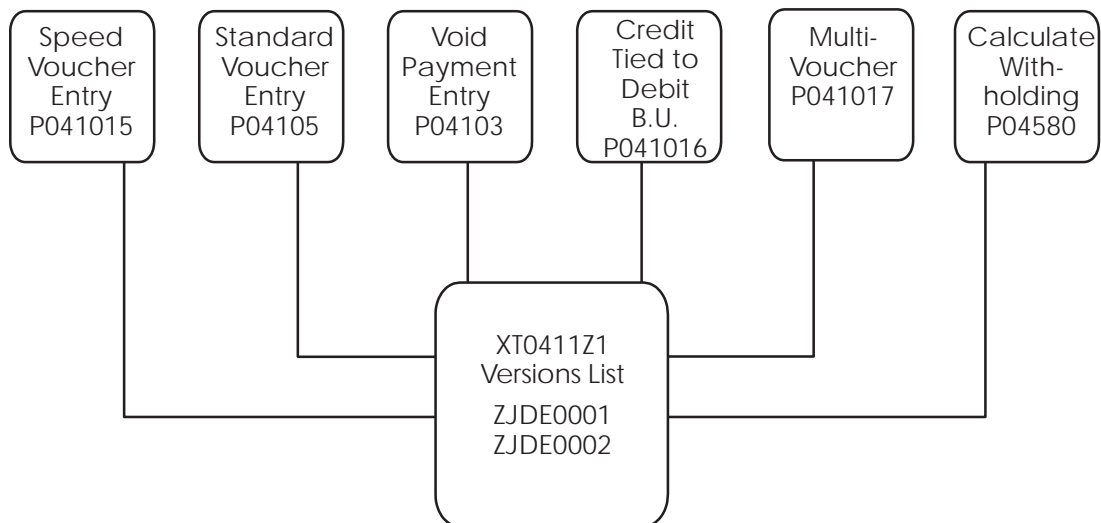
You can have all your entry programs use the same DREAM Writer version (and thus, use the same rules) or you can set up different DREAM Writer versions. J.D. Edwards provides DREAM Writer version ZJDE0001 as the default functional server version for your entry programs.



Only the person responsible for system-wide setup should make changes to the functional server version. For more information about how to set up DREAM Writer versions, see the *Technical Foundation Guide*.

### Example: Voucher Processing Functional Server

The following graphic shows the programs that use the voucher processing functional server. J.D. Edwards provides two demo versions of the functional server, ZJDE0001 and ZJDE0002.



# Glossary





# Glossary

---

This glossary defines terms in the context of your use of J.D. Edwards systems and the accompanying user guide.

**AAI.** See Automatic Accounting Instructions.

**access.** To get to the information or functions provided by the system through menus, screens, and reports.

**activity levels.** The activity level of a storage pool is the number of jobs that can run at the same time in a storage pool. The machine manages the control of this level. Often during processing in a job, a program waits for a system resource or a response from a work station user. During such waits, a job gives up its use of the storage pools in order that another job that is ready to be processed can take its place.

**A/D Cycle.** Application Development Cycle.

**advanced operating system.** A single integrated operating system which contains: relational database, display manager, storage manager, communication manager, work manager, security manager and other managers for the BIG BOSS.

**AEC.** Architectural, Engineering and Construction group.

**allocating pools.** If the system cannot allocate all the requested storage, it allocates as much storage as is available and allocates all the other as storage becomes available.

**alphabetic character.** Represents data by using letters and other symbols from the keyboard (such as \*&#). Contrast with *numeric character*.

**alphanumeric character.** Represents data in a combination of letters, numbers, and other symbols (such as \*&#).

**ANSI.** American National Standards Institute.

**answers.** Remember the online education system on the AS/400. All you need to remember is the command, *GO SUPPORT*.

**AP.** Accounts Payable.

**APD.** Application Program Driver.

**API.** An application programming interface describes the means by which a programmer can access the features provided by the interfaced object.

**APPC.** Advanced Program to Program Communications.

**application.** A collection of computer programs that allows you to perform specific business tasks. Some examples of applications are accounts payable, inventory, and order processing. Synonymous with *system*.

**APPN.** Advanced Peer-to-Peer Networking.

**AS/400.** Application System/400.

**AS/400 Office.** An IBM word processing program.

**ASCII.** American Standard Code for Information Interchange.

**ASPs.** Auxiliary Storage Pools.

**attributes.** To regard as belonging.

**attribute byte.** First character on a display field. This character controls how the field is displayed.

**audit trail.** The detailed, verifiable history of a processed transaction. The history consists of the original documents, transaction entries, and posting of records, and usually concludes with a report.

**authority.** The right to do some thing on the system or to use an object in the system, such as a file or a program.

**automatic accounting instruction**

**(AAI).** A code that points to an account in the chart of accounts. AAIs define rules for programs that automatically generate journal entries. This includes interfaces between Accounts Payable, Accounts Receivable, and Financial Reporting and the General Accounting system. Each system that interfaces with the General Accounting system has AAIs. For example, AAIs can direct the Post to General Ledger program to post a debit to a certain expense account and an automatic credit to a certain accounts payable account.

**autostart job entry.** A job is automatically started each time the subsystem is started.

**ATC.** Area Training Coordinator.

**AR.** Accounts Receivable.

**backup copy.** A copy of original data preserved on a magnetic tape or diskette as protection against destruction or loss.

**BAPR.** Approved Budget Field Description.

**BASIC.** Beginners Application Software Introduction Class.

**batch.** A group of like records or transactions that the computer treats as a single unit during processing. For identification purposes, the system usually assigns each batch a unique identifier, known as a “batch number.”

**batch header.** Information the computer uses as identification and control for a group of transactions or records in a batch.

**batch job.** A task or group of tasks you submit for processing that the system treats as a single unit during processing, for example, printing reports and purging files. The computer performs these tasks with little or no user interaction.

**batch processing.** A method by which the computer selects jobs from the job queue, processes them, and writes output to the outqueue. Contrast with *interactive processing*.

**batch type.** A code that designates which J.D. Edwards system the associated transactions pertain to, thus controlling what records are selected for processing. For example, in the Post General Journal process, only unposted transaction batches with a batch type of G for General Accounting are selected for posting.

**bit.** Binary digit. Either a zero or a one at the MI level.

**Bomb.** Fail.

**Boolean logic operand.** In J.D. Edwards DREAM Writer, the parameter of the Relationship field. The Boolean logic operand tells the system to perform a mathematical calculation on certain records or parameters. Available operands are:

EQ = Equal To

LT = Less Than

LE = Less Than or Equal To

GT = Greater Than

GE = Greater Than or Equal To

NE = Not Equal To

NL = Not Less Than

NG = Not Greater Than

**BORG.** Original/Beginning Budget Field BPC *v.* Budget Pattern Code.

**BREQ.** Requested Budget Field Description.

**B/S.** Balance Sheet.

**buffer.** A reserved memory area used for performing input/output operations.

**business unit.** Formerly cost center.

**Caching.** Refers to the use of a technique to locally store the results of input and output operations to minimize the use of slower accesses to disk drives and other storage devices.

**CAD/CAP.** Computer Assisted Design/Computer Assisted Programming. A set of automated programming tools for designing and developing applications. These tools automate system design, generate source code and documentation, enforce design standards, and help to ensure consistency throughout all J.D. Edwards systems.

**category code.** In user defined codes, a temporary title for an undefined category. For example, if you are adding a code that designates different sales regions, you could change *category code 4* to *Sales Region*, and define E (East), W (West), N (North), and S (South) as the valid codes. Category codes were formerly known as *reporting codes*.

**CC.** Cost center. *Now known as Business Unit.*

**CC.OBJ.SUB.** Cost Center.Object.Subsidiary (J.D. Edwards Account Code Structure).

**character.** Any letter, number, or other symbol that a computer can read, write, and store.

**character, special.** Representation of data in symbols that are neither letters nor numbers. Some examples are: \*&#/.

**CLONE.** Crazy Logic Only Nerds Enjoy. (Old term for the Program Generator.)

**COBOL.** Common Business Oriented Language.

**Column.** *See field.*

**command.** A character, word, phrase, or combination of keys you use to tell the computer to perform a defined activity.

**compile.** To change source code into computer readable code.

**constants.** Parameters or codes that rarely change. The computer uses constants to standardize information processing by an associated system. Some examples of constants are allowing or disallowing out-of-balance postings and having the system perform currency conversions on all

amounts. Once you set constants such as these, the system follows these rules until you change the constants.

**Core.** The central and foundational systems of J.D. Edwards software, including General Accounting, Accounts Payable, Accounts Receivable, Address Book, Financial Reporting, Financial Modeling and Allocations, and Back Office. Now called Financials.

**CPG.** Complementary Products Group.

**CRP.** Capacity Requirements Planning.

**CRP.** Conference Room Pilot. A simulation of the client's business in a conference room environment.

**CUA.** Common User Access. IBM's specification of a user interface definition across applications.

**CUM.** A representation of changes to J.D. Edwards software, which your organization receives on magnetic tapes or diskettes.

**current library.** Specifies a single library that is searched before any other user libraries in the library list. A current library is optional and can be different for each user or job. On displays, the current library is represented by the value \*CURLIB.

**cursor.** The blinking underscore or rectangle on your screen that indicates where the next keystroke appears.

**cursor sensitive help.** *See field help.*

**data.** Numbers, letters, or symbols that represent facts, definitions, conditions, and situations, that a computer can read, write, and store.

**data item.** A code which represents a field, file, program, menu message, error message or help text stored in the data dictionary. Each piece of information within the database is defined by a data item. Data item name definition is limited to four characters in the J.D. Edwards systems to allow for program manipulation of the item.

**database.** A continuously updated collection of all information a system uses and stores. Databases make it possible to create, store, index, and cross-reference information online.

**data character.** A pattern of 8 bits.

**data dictionary.** A database file consisting of the definitions, structures, and guidelines for the usage of fields, messages, and help text. The data dictionary file does not contain the actual data itself.

**data field.** A collection of data characters.

**data Integrity.** Refers to checking the relationships between data items (fields) and being sure that values correlate correctly.

**data validation.** Determining if data is correct when compared to a set of conditions.

**DDE.** Dynamic Data Exchange.

**DDM.** Distributed Data Management.

**DDP.** Distributed Data Processing.

**DDS.** Data Description Specifications.

**default.** A code, number, or parameter the system supplies when you do not enter one. For example, if an input field's default is N and the you do not enter something in that field, the system supplies an N.

**descriptive title.** See *user defined code*.

**detail.** The individual pieces of information and data that make up a record or transaction. Contrast with *summary*.

**DFU.** Data File Utility. An IBM product.

**DIF.** Data Interchange Format.

**display.** (1) To cause the computer to show information on a terminal's screen. (2) A specific set of fields and information that a J.D. Edwards system might show on a screen. Some screens can show more than one display when you press a specified function key.

**display field.** A field of information on a screen that contains a system-provided code or parameter that you cannot change. Contrast with *input field*.

**DMA.** Direct Memory Access.

**DNS.** Do Not Spread.

**DOS.** Disk Operating System.

**DREAM Writer.** Data Record Extraction And Management Writer. A flexible data manipulator and cataloging tool. You use this tool to select and sequence the data that is to appear on a programmed report.

**DRP.** Distribution Requirements Planning.

**Dynamic.** Is constantly changing.

**DASD.** Data Auxiliary Storage Device.

**ECS.** Electronic Customer Support.

**edit.** (1) To make changes to a file by adding, changing, or removing information. (2) The program function of highlighting fields into which you have entered inadequate or incorrect data.

**EDI.** Electronic Data Interchange. The transmission of business documents among computers of independent organizations.

**EFT.** Electronic Fund Transfer.

**EIS.** Executive Information System.

**Engagement letter.** A letter identifying the mutual understandings and initial expectation of the client and J.D. Edwards.

**environment.** The list of files required by a user to perform certain tasks. For example, a programmer has access to a test environment and an environment which includes live data. Each environment utilizes a different set of files.

**execute.** See *run*.

**exit.** (1) To interrupt or leave a computer program by pressing a specific key or a sequence of keys. (2) An option or function key displayed on a screen that allows you to access another screen.

**facility.** A collection of computer language statements or programs that provides a specialized function throughout a system or throughout all integrated systems. Some examples DREAM Writer and FASTR.

**Fast Path Mnemonics.** A method of using a UDC to define execution to a J.D. Edwards program.

**FASTR.** Financial Analysis Spreadsheet Tool and Report Writer. A report writer that lets you design your own report specifications using the general ledger database.

**FDA.** File Design Aid. A J.D. Edwards design tool.

**field.** (1) An area on a screen where you type in data, values, or characters. (2) A defined area, usually within a record, which can contain a specific piece of information such as name, document type or amount. For example, a vendor record consists of the fields Vendor Name, Vendor Address and Telephone Number. The field Vendor Name contains only the name of the vendor. See *input field* and *display field*. Also known as *column*.

**field help.** J.D. Edwards online Help function, which lets you view a description of a field, its purpose and, when applicable, a list of the valid codes that you can enter. You access this information by pressing F1 with the cursor positioned in the field.

**file.** A collection of related data records organized for a specific use and electronically stored by the computer. Also known as *table*.

**financial systems.** The central and foundational systems of J.D. Edwards software, including General Accounting, Accounts Payable, Accounts Receivable, Address Book, Financial Reporting, Financial Modeling and Allocations, and Back Office. *Previously known as core*.

**fold area.** An area of a screen, accessed by pressing F4, that displays additional information associated with the records or data items displayed on the screen.

**function.** A separate feature within a facility that allows you to perform a specific task, for example, the field help function.

**function key.** A key you press to perform a system operation or action. For example, you press F4 to have the system display the fold area of a screen.

**Form.** One World term for video.

**glossary.** The collection of text related to specific data items. The glossary contains help text and message text.

**GL.** General Ledger.

**GA.** General Accounting.

**GST.** Goods & Service Tax.

**GUI.** Graphical User Interface.

**hard code.** Program instructions which can only be altered by a programmer. The altered instructions must then recompiled so the computer can understand them.

**hard copy.** A presentation of computer information printed on paper. Synonymous with *printout*.

**header.** Information at the beginning of a file. This information is used to identify or provide control information for the group of records that follows.

**help instructions.** Online documentation or explanations of fields that you access by pressing the Help key or by pressing F1 with your cursor in a particular field.

**helps.** See *help instructions*.

**hidden selections.** Menu selections you cannot see until you enter HS in a menu's Selection field. Although you cannot see these selections, they are available from any menu. They include such items as Display Submitted Jobs (33), Display User Job Queue (42), and Display User Print Queue



(43). The Hidden Selections window displays three categories of selections: user tools, operator tools, and programmer tools.

**HMC.** Horizontal Microcode.

**HS.** J.D. Edwards Hidden Selections.

**ICCC.** InterCompany Cost Center. *Now known as business unit.*

**ICF.** Intersystem Communication Function.

**ICH.** InterCompany Hub.

**IDDU.** Interactive Data Definition Utility – IBM Product.

**IMP.** Internal Microprogram Load.

**IMPI.** Internal Microprogramming Interface.

**Implementation Methodology.** Nine steps to provide J.D. Edwards consulting staff with a guide for implementing the software in a thorough and consistent manner.

**input.** Information you enter in the input fields on a screen or that the computer enters from other programs, then edits and stores in files.

**input field.** An area on a screen, distinguished by underscores ( \_ \_ ), where you type data, values, or characters. A field represents a specific type of information such as name, document type, or amount. Contrast with *display field*.

**install system code.** The four-character identifier of a J.D. Edwards system. For example, 01 for the Address Book system, 04 for the Accounts Payable system, and 09 for the General Accounting system. *Now known as system code.*

**integrity.** Soundness, completeness.

**interactive job.** An interactive job starts when a user signs on a display station and ends when the user signs off. During the job, the user interacts with the system.

**interactive processing.** A job the computer performs in response to commands you enter from a terminal.

During interactive processing, you are in direct communication with the computer, and it might prompt you for additional information during the processing of your request. See *online*. Contrast with *batch processing*.

**interface.** A link between two or more J.D. Edwards systems that allows these systems to send information to and receive information from one another.

**I/O.** Input/Output.

**IPL.** Initial Program Load.

**ITF.** Interactive Terminal Facility.

**JDE.** Jack, Dan and Ed. Founders of JD Edwards & Co.

**jargon.** A J.D. Edwards term for system-specific text. You base your jargon help text on a specific reporting code you designate in the Data Dictionary Glossary. You can display this text as part of online help. You create your jargon text descriptions and titles for data items through the Data Dictionary, menu and vocabulary overrides record using a reporting system code. Jargon text descriptions and titles for data items display on screens as field names.

**job.** A single identifiable set of processing actions you tell the computer to perform. You start jobs by choosing menu selections, entering commands, or pressing designated function keys. An example of a computer job is check printing in the Accounts Payable system.

**job description.** An object consisting of a set of specifications about a computer job and its executing environment.

**job log.** A job log is a record of requests (such as commands) submitted by the system by a job, the messages related to the requirements and the actions performed by the system on the job.

**job queue.** A group of jobs waiting to enter a subsystem.

**Join logical file.** Presents composite records consisting of fields extracted from two or more physical records from two or more physical files.

**justify.** To shift information you enter in an input field to the right or left side of the field. Many of the facilities within J.D. Edwards systems justify information. The system does this only after you press Enter.

**KBG.** Knowledge-Based Generator. See *program generator*.

**key field.** A series of identifying or controlling characters a computer uses to retrieve related information tied to the key. An employee number, for example, is a key field consisting of references to other files in the system that contain information about the given employee.

**Key General Ledger Account (Key G/L).** See *automatic accounting instructions*.

**LAN.** Local Area Network.

**leading zeros.** A series of zeros that certain facilities in J.D. Edwards systems place in front of a value you enter. This normally occurs when you enter a value that is smaller than the specified length of the field. For example, if you enter 4567 in a field that accommodates eight numbers, the facility places four zeros in front of the four numbers you enter. The result would look like this: 00004567.

**level check.** A mechanism of the OS/400 that assures that a file version and program using that file are in sync with one another.

**level of detail.** (1) The degree of difficulty of a menu in J.D. Edwards software. The levels of detail for menus are as follows:

- A=Major Product Directories
- B=Product Groups
- 1=Daily Operations
- 2=Periodic Operations
- 3=Adv/Tech Operations
- 4=Computer Operations

5=Programmers

6=Advanced Programmers

Also known as *menu levels*. (2) The degree to which account information in the General Accounting system is summarized. The highest level of detail is 1 (least detailed) and the lowest level of detail is 9 (most detailed).

**library.** A library groups objects. A library is an object itself. Similar to directory on a PC.

**library list.** An ordered list of libraries used for locating objects. Similar to path on a PC.

**LIOM.** Line Input/Output Manager.

**LOD.** Level of Detail.

**logical file.** Contains no data, but provides a view of one or more physical files upon which it is based.

**master file.** A computer file that a system uses to store data and information which is permanent and necessary to the system's operation. Master files might contain data or information such as paid tax amounts and vendor names and addresses.

**MDA.** Menu Design Aid. A J.D. Edwards design tool.

**menu.** A screen that displays numbered selections. Each of these selections represents a program. To access a selection from a menu, type the selection number and then press Enter.

**menu levels.** See *level of detail*.

**menu masking.** A security feature of J.D. Edwards systems that allows you to prevent individual users from accessing specified menus or menu selections. When this security is in effect for a user, the selections that have been secured do not appear on the screen.

**menu message.** Text that appears on a screen after you make a menu selection. It displays a warning, caution, or information about the requested selection.

**menu traveling.** A method of moving between menus by typing the menu identifier in the selection field of the screen.

**MI.** Machine Interface.

**MRP.** Manufacturing Resource Planning.

**MRPx.** J.D. Edwards Manufacturing Software.

**MVS.** Multiple Virtual Storage.

**next number facility.** A J.D. Edwards software facility you use to control the automatic numbering of such items as new G/L accounts, vouchers, and addresses. It lets you specify your desired numbering system and provides a method to increment numbers to reduce transposition and typing errors.

**non-join logical file.** Presents records that are composed of fields extracted from just one physical record, but can effectively merge two or more physical files.

**numeric character.** Represents data using the numbers 0 through 9. Contrast with *alphabetic character* and *alphanumeric character*.

**object.** A discrete entity.

**object existence.** The right to delete an object from the system.

**object management.** The right to change the name or library of an object, for physical files, the right to create a logical file over it.

**object operational.** The right to display the description of an object and the right to the general use of that object.

**object orientation.** Everything on the AS/400 system that can be stored or retrieved is contained in an object.

**offline.** Computer functions that are not under the continuous control of the system. For example, if you were to run a certain job on a personal computer and then

transfer the results to a host computer, that job would be considered an offline function. Contrast with *online*.

**One Step Install.** A method developed to make our software easier to install.

**online.** Computer functions over which the system has continuous control. Each time you work with a J.D. Edwards system-provided screen, you are online with the system. Contrast with *offline*. See *interactive processing*.

**online information.** Information the system retrieves, usually at your request, and immediately displays on the screen. This information includes items such as database information, documentation, and messages.

**Open Application Architecture.** An architecture that uses a functional server to allow the various blocks of user interface logic to **access** the same block of data integrity logic.

**operand.** See *Boolean logic operand*.

**option.** A numbered selection from a J.D. Edwards screen that performs a particular function or task. To select an option, you enter its number in the Option field next to the item you want the function performed on. When available, for example, option 4 lets you return to a prior screen with a value from the current screen.

**OS/400.** Operating system for the AS/400.

**OS/2.** Operating system for the IBM personal computer.

**OSI.** Open Systems Interconnection.

**output.** Information the computer transfers from internal storage to an external device, such as a printer or a computer screen.

**output queue.** A group of spool files waiting to be attached to a writer.



**override.** The process of entering a code or parameter other than the one provided by the system. Many J.D. Edwards systems offer screens that provide default field values when they appear. By typing a new value over the default code, you can *override* the default. See *default*.

**PACO.** Posted After Cutoff.

**parameter.** A number, code, or character string you specify in association with a command or program. The computer uses parameters as additional input or to control the actions of the command or program.

**password.** A unique group of characters that you enter when you sign on to the system that the computer uses to identify you as a valid user.

**PBCO.** Posted Before Cutoff.

**PC.** Personal computer.

**PDM.** Program Development Manager. IBM design tool.

**PDM.** Product Data Management – a module of J.D. Edwards software.

**physical file.** A file that contains actual data records. Has a maximum record length of 32K, maximum fields per record is 8000.

**Plug-&Go.** A 2/18/92 announcement where J.D. Edwards selects PROGRESS to develop client applications for the AS/400. The plug-&-go format offers clients the J.D. Edwards Core financial solutions on the IBM AS/400 E series model.

**PPAT.** People, Places and Things.

**printout.** A presentation of computer information printed on paper. Synonymous with *hard copy*.

**print queue.** A group of items waiting to be printed. See *output queue*.

**processing options.** A feature of the J.D. Edwards DREAM Writer that lets you supply parameters to direct the functions of a program. For example, processing options allow you to specify defaults for certain screen displays, control the format in which

information gets printed on reports, change the way a screen displays information, and enter “as of” dates.

**product library.** A library containing programs and related data needed for IBM licensed programs that are installed on your system.

**production library.** A production library is a library you create to contain your live J.D. Edwards data files.

**production environment.** A list of libraries that contains “live” programs and data.

**program.** A collection of computer statements that tells the computer to perform a specific task or group of tasks.

**Progress.** A software corporation that is a partner with J.D. Edwards. They are a leading supplier of 4th generation application development systems.

**program generator.** The World CASE system of programs which create a new program based upon user specifications.

**program help.** J.D. Edwards online facility which displays information about a program’s use and functionality.

**program-specific help text.** Glossary text written to describe the function of a field within the context of the program.

**prompt.** (1) A reminder or request for information displayed by the system. When a prompt appears, you must respond in order to proceed. (2) A list of codes or parameters or a request for information provided by the system as a reminder of the type of information you should enter or action you should take.

**PTF.** See *CUM*.

**purge.** The process of removing records or data from a file.

**PYEB.** Post Year End Balance.

**P&L.** Profit and Loss Statements.

**PG.** Program Generator.

**QA.** Quality Assurance.

**QJDF data area.** A space within the system to hold the system values information for the J.D. Edwards software. This area is referenced at sign-on and during installs and reinstalls for critical system information, such as security codes and initial libraries.

**QSECOFR.** The security officer of the AS/400.

**query.** A fast means to select and display (or print) information from a database. An IBM utility for databases.

**queue.** A list of things to be used in an order. See *job queue*, *output queue*, and *print queue*.

**RAID.** Redundant Array of inexpensive disks.

**RAM.** Random Access Memory.

**RDA.** Report Design Aid. A J.D. Edwards design tool.

**read only.** A type of access to data that allows it to be read but not copied, printed or modified.

**rebuild.** The process of sequencing files, integrating recently added data.

**record.** A collection of related, consecutive fields of data the system treats as a single unit of information. For example, a vendor record consists of information such as the vendor's name, address, and telephone number. *Also known as row.*

**record format.** The definition of how data is structured in the records contained in a file.

**record level locking.** Prevents two people from simultaneously updating the same data base information.

**REP.** Rapidly, Economically and Predictably.

**reply list.** A system wide automatic message handler for the system.

**recursive.** In DREAM Writer, the ability to create a unique version from the original, process the new version and delete it, leaving the original intact.

**re-engineering modules.** Programs written for the purpose of changing many existing programs in mass.

**reporting system code.** The four-character identifier of a J.D. Edwards system that uses an object for reporting.

**REQIO.** Request Input/Output.

**reverse image.** Screen text that displays in the opposite color combination of characters and background from what the screen typically displays (for example, black on green instead of green on black).

**RIBA.** Ricevuta Bancaria Elettronica — common way for vendors to receive payments from their customers in Italy.

**ROM.** Read Only Memory.

**ROW.** *See record.*

**RPG.** Report Program Generator. A programming language developed by IBM.

**Rumba.** A PC Emulator for the AS/400.

**run.** To cause the computer to perform a routine, process a batch of transactions, or carry out computer program instructions.

**SAA.** Systems Application Architecture.

**SAR.** *See Software Action Request.*

**server.** A program that speeds the flow of data between screens, reports and the data files. These programs can also be used to edit data fields.

**scroll.** To use the roll keys to move screen information up or down a screen at a time. When you press the Rollup key, for instance, the system replaces the currently displayed text with the next screen of text if more text is available.

**SDA.** Screen Design Aid Utility. An IBM product.

**selection.** Found on J.D. Edwards menus, selections represent functions that you can access from a given menu. To make a selection, you type its associated number in the Selection field and press Enter.

**SEU.** Source Entry Utility.

**SIC.** Standard Industry Code.

**SIOM.** Station Input/Output Manager.

**Ski Slope.** Reflects the analogy between the diverse nature of a ski slope and the diverse nature of our software. S levels: Basic, Intermediate, Advanced, Computer Operations and Program Modifications.

**SNA.** Systems Network Architecture.

**SNADS.** Systems Network Architecture Distribution Services.

**Sleeper.** A subsystem which activates jobs set to run during off-peak hours.

**softcoding.** A J.D. Edwards term that describes an entire family of features that lets you customize and adapt J.D. Edwards software to your business environment. These features lessen the need for you to use computer programmers when your data processing needs change.

**software.** The operating system and application programs that tell the computer how and what tasks to perform.

**Software Action Request.** A record which identifies an activity, such as the development of a new program or maintenance of an existing program.

**Software Security Code.** A code that restricts user access to software.

**special character.** Representation of data in symbols that are neither letters nor numbers. Some examples are \* & # /.

**spool.** Simultaneous Peripheral Operations On Line. The function by which the system puts generated output into a storage area to await printing or processing.

**spooled file.** A holding file for output data waiting to be printed or input data waiting to be processed.

**SQL.** Structure Query Language.

**STAR.** Spreadsheet Tool for Asset Reporting.

**subfile.** An area on the screen where the system displays detailed information related to the header information at the top of the screen. Subfiles might contain more information than the screen can display in the subfile area. If so, use the roll keys to display the next screen of information. See *scroll*.

**submit.** See *run*.

**subsystem.** An operating environment where jobs are run.

**summary.** The presentation of data or information in a cumulative or totaled manner in which most of the details have been removed. Many of the J.D. Edwards systems offer screens and reports that are summaries of the information stored in certain files.

**SVR.** Software Versions Repository.

**system.** A collection of computer programs that lets you perform a specific business function, such as Accounts Payable, Inventory, or Order Processing. Synonymous with *application*.

**system library.** Lists libraries containing objects, such as user profiles, that are used by the system. This part of a library list is defined by the system value QSYSLIBL and is usually the same for all jobs.

**Simplified Install.** J.D. Edwards new way to install J.D. Edwards software. Also called one step Install.

**SME.** Subject Matter Expert.

**T/B.** Trial Balance.

**Table.** One World term for a file.

**UNIX.** A multi-user, multi-tasking operating system.

**Unscheduled PTF.** A form of PTF that includes fixed for a particular system.

**UPS.** Uninterruptible power source.

**user class/group.** Place to enter group profiles associated with J.D. Edwards Users.

**user defined code.** The individual codes you create and define within a user defined code type. Code types are used by programs to edit data and allow only defined codes. These codes might consist of a single character or a set of characters that represents a word, phrase, or definition. These characters can be alphabetic, alphanumeric, or numeric. For example, in the user defined code type table ST (Search Type), a few codes are C for Customers, E for Employees, and V for Vendors.

**user defined code (type).** The identifier for a table of codes with a meaning you define for the system (for example, ST for the Search Type codes table in Address Book). J.D. Edwards systems provide a number of these tables and allow you to create and define tables of your own. User defined codes were formerly known as *descriptive titles*.

**user index.** An object that stores data, allows search functions, and automatically sorts data based upon a key value.

**user identification (user ID).** The unique name you enter when you sign on to a J.D. Edwards system to identify yourself to the system. This ID can be up to 10 characters long and can consist of alphabetic, alphanumeric, and numeric characters.

**user library.** A libraries that contains objects, such as files and programs used by the user.

**user profile.** A file of information which identifies the user to the J.D. Edwards system. This file is used to validate the users authority within the system.

**user space.** An object made up of a collection of bytes used for storing user-defined information.

**user type.** A code which identifies a list of files which remain open while the user is signed on to the system.

**valid codes.** The allowed codes, amounts, or types of data that you can enter in a specific input field. The system checks, or edits, user defined code fields for accuracy against the list of valid codes.

**version.** A specific release of software. Usually numbered in ascending order.

**VCS.** Version Control System.

**Vertex.** Callable routines and tables that calculate US PIR taxes.

**video.** The display of information on your monitor screen. Normally referred to as the *screen*.

**VM.** Virtual Machine.

**VMC.** Vertical Microcode.

**vocabulary overrides.** A J.D. Edwards facility that lets you override field, row, or column title text on a screen-by-screen or report- by-report basis.

**WACO.** Way After Cutoff.

**WAN.** Wide Area Network.

**window.** A software feature that allows a part of your screen to function as if it were a screen in itself. Windows serve a dedicated purpose within a facility, such as searching for a specific valid code for a field.

**writer.** A J.D. Edwards printer attached to an outqueue.

**World Vision.** A complementary product that converts graphical user interfaces to J.D. Edwards business applications for the AS400.

**World VISTA.** A windows-based direct access to J.D. Edwards data on the AS/400.

**WW.** World Writer. A J.D. Edwards software product.

**XREF.** Cross reference tool for J.D. Edwards software.

**YTD.** Year to Date.



# Index





# Index

---

## Symbols

\*ENTRY PLIST entries, creating, 3-54

## A

Abbreviations for the program types index, 8-9

About action diagramming, 7-21

About additional tools, 7-1

About changing generated source, 5-3

About creating or modifying program types, 8-7

- abbreviations for the program types index, 8-9

- clone status all/only active toggle, 8-11

- glossary K, 8-8

- program types corss reference, 8-11

- program types index, 8-9

- selections, 8-10

About edit screen, 3-33

About foundation information, 2-1

- J.D. Edwards provided prerequisites, 2-1

- user provided prerequisites, 2-1

About Logic modules, 8-13

About master source code, 1-11

About option and function key exits, 3-39

About program design language, 4-1

About program specifications, 1-11

About program types, 1-11

About quick start application tool, 7-7

About source modifications, 5-1

About special characters, 3-34

About the detailed programming facility form, 3-45

About the Program Generator, 1-10

About the quick start CL generator, 7-3

About user defined PDL, 8-59

About using the source code inventory and database, 8-1

Accessing CASE profiles, 2-15

- default development environment, 2-17

- miscellaneous options, 2-19

- program creation options, 2-18

- SAR options, 2-19

Accessing data item formula revisions, 4-3

Accessing logic translation feature, 7-28

Accessing the data dictionary glossary, 7-16

Accessing the logic module index, 8-16

Accessing the Program Generator, 3-4

Accessing the quick start CL generator, 7-4

Accessing the screen or report you are creating, 7-12

Action diagram

- building, 7-22

- viewing, 7-23

Action diagramming, 7-21

Action Diagramming form (P9363), 7-21

Adding new Q & A dialogue, 8-46

Additional tools, 7-1

Advanced & Technical Operations form (G9), 1-14

All help instructions, 8-27

Answer Entry form (P98552), 8-48

Application development cycle, 1-5

Application generation, completing, 7-18

Arrays, EMK, @MK and @ER, A-1

Assignments, understanding, 4-10

Authorities, object, 2-21

Automatic Accounting Instructions form (P93106), 3-7

## B

Benefits of CASE, 1-8

Blocks of statements, understanding, 4-8

Browse form, 3-56, 3-67, 3-68, 3-69, 3-70, 8-16

Build Action Diagrams form (P98300), 7-22

Building an action diagram, 7-22

## C

CAD. *See* Computer Assisted Design

Calls, understanding, 4-13

CAP. *See* Computer Assisted Programming  
CAP status, changing, 5–8  
CAP status invalid error, to correct, 5–10  
CASE profiles  
    accessing, 2–15  
    understanding, 2–14  
CASE Profiles form (P98009), 2–16  
CASE specifications inquiry, using, 8–26  
CASE Specifications Inquiry form (P93130), 8–26  
Changing a dialogue, 8–51  
Changing CAP status, 5–8  
Changing the program specifications, 7–14  
CL program, compiling, 7–6  
Clearing fields, protecting from, 3–57  
Comments, understanding, 4–9  
Common User Defined Codes, 2–5  
    common subroutine copy members, 2–6  
    logic modules, 2–6  
Compiling CL program, 7–6  
Completing application generation, 7–18  
Computer Assisted Design. *See* CAD  
Computer Assisted Design form (G92), 2–15  
Computer Assisted Programming. *See* CAP  
Computer Assisted Programming (CAP) form (G93), 1–14, 3–4  
Conditional directives, 8–40  
Conditions, understanding, 4–15  
Copy File form (CPYF), 2–11  
Copying a dialogue, 8–52  
Create, JDESRC file for use with Program Generator, 2–10  
Create/Modify Logic Modules form (P93001SEU), 8–15  
Create/Modify Program Types form (P93001), 8–8, 8–60  
Creating \*ENTRY PLIST entries, 3–55  
Creating a partial KLIST for a file, 3–60  
Creating a user defined PDL, 8–60  
    limitations, 8–62  
Creating or modifying formula library entry, 8–21  
Creating or modifying logic modules, 8–15  
Creating or modifying parameter copy/move, 8–22

## D

Data dictionary edits, disabling, 3–59  
Data dictionary glossary, accessing, 7–16  
Data fields, selecting, 7–11  
Data item formula revisions, accessing, 4–3  
Data Item Formula Revisions form (P93109), 4–4, 4–18, 4–23, 4–24, 8–21, 8–63  
Database operations, understanding, 4–11  
Database update function for subfiles, enabling, 3–54  
Default development environment, 2–17  
Define Generator Specification form (P93100M), 3–10, 3–20, 3–32, 3–39, 3–44, 3–62, 5–9  
Defining general instructions, 3–31  
Defining option and function key exits, 3–40  
Defining processing options, 3–61  
Defining Program Generator specifications, 3–3, 3–5  
Defining Program Generator Specifications form (P93100M), 3–5  
Defining program purpose and type, 3–9  
Deleting a dialogue, 8–56  
Detailed information, 1–9  
    CASE profiles, 1–9  
    Computer Assisted Design (CAD), 1–9  
    Computer Assisted Programming (CAP), 1–10  
Detailed programming facility, 3–45  
    working with, 3–43  
Detailed Programming Facility form (P92801), 3–45  
Detailed Programming Facility form (P93105), 3–60, 4–3, 8–62  
Development libraries, 2–10  
Dialogue  
    changing, 8–51  
    copying, 8–52  
    deleting, 8–56  
    guidelines, 8–57  
    inquiring on, 8–49  
    rename, 8–53  
    running, 8–54  
Dialogue Copy form (P98536), 8–52, 8–53  
Dialogue Descriptions form (P98541), 8–47  
Dialogue Flow Revisions form (P98531), 8–50, 8–51

Dialogue Lists form (P98530), 8-49  
 Dialogue Selection form (P98533), 3-13, 8-55  
 Dialogue Test form (P98535), 8-54, 8-56  
 Dialogue Test form (P98537), 8-57  
 Directives  
     conditional, 8-40  
     exception, 8-39  
     functional, 8-29  
     substitution, 8-37  
     understanding, 8-29  
 Disabling data dictionary edits, 3-59  
 Display Action Diagram form (P92705), 7-23, 7-24, 7-26

## E

Edit form, 3-33, 3-35, 8-24, 8-61  
 Edit screen, 3-33  
 Editing, parsing, and source generation, PDL, 4-21  
 Enabling database update function for subfiles, 3-54  
 Error handling, Arrays, A-1  
 Exception directives, 8-39

## F

Features, 1-7  
 File Specification form, understanding, 3-21  
 File specifications, 3-23  
     processing, 3-25  
     working with, 3-19  
 File Specifications form (P93102), 3-21, 3-26  
 Forms  
     Action Diagramming, 7-21  
     Advanced & Technical Operations, 1-14  
     Answer Entry, 8-48  
     Automatic Accounting Instructions, 3-7  
     Browse, 3-56, 3-67, 3-68, 3-69, 3-70, 8-16  
     Build Action Diagrams, 7-22  
     CASE Profiles, 2-16  
     CASE Specifications Inquiry, 8-26  
     Computer Assisted Design (CAD), 2-15

Computer Assisted Programming (CAP), 1-14, 3-4  
 Copy File (CPYF), 2-11  
 Create/Modify Logic Modules, 8-15  
 Create/Modify Program Types, 8-8, 8-60  
 Data Item Formula Revisions, 4-4, 4-18, 4-23, 4-24, 8-21, 8-63  
 Define Generator Specification, 3-10, 3-20, 3-32, 3-39, 3-44, 3-62, 5-9  
 Define Generator Specifications, 3-5  
 Detailed Programming Facility, 3-45, 3-60, 4-3, 8-62  
 Dialogue Copy, 8-52, 8-53  
 Dialogue Descriptions, 8-47  
 Dialogue Flow Revisions, 8-50, 8-51  
 Dialogue Lists, 8-49  
 Dialogue Selection, 3-13, 8-55  
 Dialogue Test, 8-54, 8-56, 8-57  
 Display Action Diagram, 7-23, 7-24, 7-26  
 Edit, 3-33, 3-35, 8-24, 8-61  
 File Specifications, 3-21, 3-26  
 Formula Library Entry, 4-18  
 Full Data Field Parameters, 3-48, 3-51, 3-53, 3-54, 3-55, 3-57, 3-58, 3-59  
 Generator Updates, 8-25  
 Glossary, 7-17  
 Logic Module Op Codes, 8-19  
 Logic Module X-Reference, 8-18  
 Master Directory, 1-14  
 Model Program Design, 1-14, 8-2  
 Option & Function Key Exits, 3-40  
 Parameter Copy/Move, 8-22  
 Print KBG Specification, 8-23  
 Processing Options Setup, 3-63  
 Program Purpose and Type, 3-11, 5-8  
 Program Types Cross Reference, 8-11  
 Program Types Index, 8-9  
 Question Entry, 8-46, 8-47  
 Quick Start Application Tool, 7-9, 7-11, 7-12, 7-13, 7-14, 7-15, 7-16, 7-18  
 Quick Start CL Generator, 7-4, 7-6  
 Quiz Answer Review, 8-55  
 Remove Member, 2-12  
 Sample Program – Item Mtc, 3-37  
 Simple Question & Answer, 8-45  
 Software Versions Repository, 2-16, 3-28  
 Translation Table, 7-28  
 User Defined Code Revisions, 2-6, 2-7, 2-8  
 World CASE Q & A, 8-44

- Formula library entry, creating or modifying, 8–21
- Formula Library Entry form (P93109), 4–18
- Foundation information, 2–1
- Full data field parameters
  - accessing, 3–48
  - understanding, 3–47
- Full Data Field Parameters form (P93125), 3–48, 3–51, 3–53, 3–54, 3–55, 3–57, 3–58, 3–59
- Function key exits, 3–17, 3–26, 3–42, 3–46
  - automatic accounting instructions, 3–7
  - cursor sensitive
    - data cross reference, 7–27
    - data dictionary, 7–27
    - file field description, 7–26
    - software versions repository, 7–26
  - data model, 3–27
  - delete all specifications, 3–8
  - display file usage, 7–24
  - exit, 3–27
  - extended parameters, 3–26
  - file field description window, 3–50
  - flowchart, 7–25
  - from PDL
    - display variable definitions, 4–25
    - repository services, 4–25
  - language preference text, 3–64
  - monitor, 3–6
  - previous profile, 2–19
  - print, 7–25
  - repository services, 3–6, 3–46, 3–64
  - return to previous logic level, 7–24
  - scan backward, 7–25
  - scan forward, 7–24
  - search, 3–6, 3–27, 3–42
  - select \*PROC fields on/off, 3–46
  - select all, 3–8
  - skip to end group, 7–25
  - skip to start group, 7–25
- Function keys
  - menu, 1–2
  - screen, 1–3
- Functional directives, 8–29

## G

- General instructions, defining, 3–31

- Generated source, changing, 5–3
- Generated source code, to change, 5–4
- Generating source from specifications, 3–28
- Generation options
  - all help instructions, 8–27
  - global program regeneration, 8–27
  - help instructions edit/build, 8–27
- Generation problems, solving, 5–10
- Generator Updates form (P9366), 8–25
- Global program regeneration, 8–27
- Glossary, updating, 7–17
- Glossary form (P98002), 7–17
- Glossary K, 8–8
- Graphic
  - help instructions, 3–36
  - interactive program, 3–15
  - pre-SEU and post-SEU process, 5–5
  - program type, 3–14
  - program types, 3–16
  - structure of the serial number, 8–5
- Guidelines, dialogue, 8–57

## H

- Help file, updating, 3–36
- Help instructions edit/build, 8–27
- Hidden selections, 1–4

## I

- Indicators, Table, A–3
- Inquiring on a dialogue, 8–49
  - selection exits, 8–49

## J

- J.D. Edward provided prerequisites
  - common user defined codes, 2–5
  - Program Generator files, 2–4
  - source code for copy modules, 2–8
  - source code for J.D. Edwards files, 2–8
  - working with, 2–3
- J.D. Edwards training environment, 1–16
- Job queues, 2–12

**L**

- Library naming conventions, 1-13
- Loading VCO description fields, 3-51
- Logic module cross reference, using, 8-17
- Logic module index, accessing, 8-16
- Logic module op codes, using, 8-19
- Logic Module Op Codes form (P93108), 8-19
- Logic Module X-Reference form (P93952), 8-18
- Logic modules, 8-13
  - creating or modifying, 8-15
  - detail, 8-14
  - manually performed, 8-20
  - primary, 8-13
- Logic translation feature, accessing, 7-28
- Loops, understanding, 4-14

**M**

- Manually performed logic modules
  - remove logic module, 8-20
  - resequence logic module, 8-20
- Master Directory form (G), 1-14
- Menu, overview, 1-14
- Menu function keys, 1-2
- Menu G9364, option 3 – simple question and answer, 8-45
- Miscellaneous keywords and syntax, understanding, 4-18
- Miscellaneous options, 2-19
- Model Program Design form (G9361), 1-14
- Model Program Design form (P9361), 8-2
- Multi-member source file, 2-10

**O**

- Object authorities, 2-21
  - job control, 2-21
  - job queues, 2-21
  - source file, 2-21
  - source library, 2-21
- Option & Function Key Exits form (P93104), 3-40
- Option and function key exits, 3-39, 3-40

**P**

- Parameter copy/move, creating or modifying, 8-22
- Parameter Copy/Move form (P93890), 8-22
- Partial KLIST for a file, creating, 3-60
- Passing parameters, 3-42
- PDL
  - See also* Program Design Language
  - user defined, 8-59
    - creating, 8-60
- PDL editing, parsing, and source generation
  - editing, 4-21
  - parsing, 4-21
  - source code generation, 4-21
- PDL statements, 4-5
  - constants, 4-7
  - database files, 4-7
  - keywords, 4-6
  - operators, 4-7
  - punctuation, 4-7
  - variables, 4-6
- Primary module, 8-8
- Print KBG Specification form (P98300), 8-23
- Printing program generator specifications, 8-23
- Processing file specifications, 3-25
- Processing options, defining, 3-61
- processing options, defining, 3-66
- Processing Options Setup form (P98304), 3-63
- Processing options setup screen, 3-63
- Program, submitting to compile, 7-15
- Program creation options, 2-18
- Program design language, 4-1
- Program Design Language (PDL). *See* PDL
- Program Generator, accessing, 3-4
- Program Generator files, 2-4
  - program design language, 2-5
  - Q & A dialogue, 2-5
  - source modifications/helps, 2-4
  - specifications, 2-4
- Program Generator specifications, defining, 3-3, 3-5
- Program generator specifications, printing, 8-23
- Program generator updates, using, 8-25
- Program purpose and type, defining, 3-9

- Program Purpose and Type form (P93100), 3–11, 5–8
- Program specifications, changing, 7–14
- Program type, 8–8
- Program types, 1–12
  - batch, 1–12
  - conversion, 1–12
  - creating or modifying, 8–7
  - interactive, 1–12
  - report, 1–12
  - window, 1–12
- program types cross reference, 8–11
- Program types index, 8–9
- Program Types Index form (P93900), 8–9
- Program Types X-Reference form (P93953), 8–11
- Programs and IDs
  - Browse, 3–56
  - browse, 3–67, 3–68, 3–69, 3–70, 8–16
  - CPYF (copy file), 2–11
  - Edit, 3–33, 3–35
  - edit, 8–24, 8–61
  - G (master directory), 1–14
  - G9 (advanced & technical operations), 1–14
  - G92 (computer assisted design), 2–15
  - G93 (computer assisted programming), 1–14, 3–4
  - G9361 (model program design), 1–14
  - P00051 (user defined code revisions), 2–6, 2–7, 2–8
  - P2710 (translation table), 7–28
  - P92705 (display action diagrams), 7–23, 7–24, 7–26
  - P92801 (detailed programming facility), 3–45
  - P92801 (sample program – item mtc), 3–37
  - P93001 (create/modify program types), 8–8, 8–60
  - P93001SEU (create/modify logic modules), 8–15
  - P93100 (program purpose and type), 3–11, 5–8
  - P93100M (define generator specification), 3–10, 3–20, 3–32, 3–39, 3–44, 3–62, 5–9
  - P93100M (define generator specifications), 3–5
  - P93102 (file specifications), 3–21, 3–26
  - P93104 (option & function key exits), 3–40
  - P93105 (detailed programming facility), 3–60, 4–3, 8–62
  - P93106 (automatic accounting instructions), 3–7
  - P93108 (logic module op codes), 8–19
  - P93109 (data item formula revisions), 4–4, 4–18, 4–23, 4–24, 8–21, 8–63
  - P93109 (formula library entry), 4–18
  - P93125 (full data field parameters), 3–48, 3–51, 3–53, 3–54, 3–55, 3–57, 3–58, 3–59
  - P93130 (CASE specifications inquiry), 8–26
  - P93513 (quick start application tool), 7–9
  - P93513J (quick start CL generator), 7–4
  - P93515V (quick start application tool), 7–11, 7–12, 7–13, 7–14, 7–15, 7–16, 7–18
  - P93515V (quick start CL generator), 7–6
  - P9361 (model program design), 8–2
  - P9363 (action diagramming), 7–21
  - P9364 (world CASE Q & A), 8–44
  - P9366 (generator updates), 8–25
  - P93890 (parameter copy/move), 8–22
  - P93900 (program types index), 8–9
  - P93952 (logic module x-reference), 8–18
  - P93953 (program types x-reference), 8–11
  - P98002 (glossary), 7–17
  - P98009 (CASE profiles), 2–16
  - P9801 (software versions repository), 2–16, 3–28
  - P98300 (build action diagrams), 7–22
  - P98300 (print KBG specification), 8–23
  - P98304 (processing options setup), 3–63
  - P98529 (simple question & answer), 8–45
  - P98530 (dialogue lists), 8–49
  - P98531 (dialogue flow revisions), 8–50, 8–51
  - P98533 (dialogue selection), 3–13, 8–55
  - P98534 (quiz answer review), 8–55
  - P98535 (dialogue test), 8–54, 8–56
  - P98536 (dialogue copy), 8–52, 8–53
  - P98537 (dialogue test), 8–57
  - P98541 (dialogue descriptions), 8–47
  - P98551 (question entry), 8–46, 8–47
  - P98552 (answer entry), 8–48
  - RM/M (remove member), 2–12
- Project management, 2–13
- Protecting fields from being cleared, 3–57



## Q

Q & A dialogue, adding new, 8-46  
 Question and answer system, working with, 8-43  
 Question Entry form (P98551), 8-46, 8-47  
 Questions, reviewing, 8-45  
 Quick start, steps, 7-8  
 Quick start application definition, selecting, 7-9  
 Quick start application tool, 7-7  
 Quick Start Application Tool form (P93513), 7-9  
 Quick Start Application Tool form (P93515V), 7-11, 7-12, 7-13, 7-14, 7-15, 7-16, 7-18  
 Quick start CL generator, 7-3  
   accessing, 7-4  
 Quick Start CL Generator form (P93513J), 7-4  
 Quick Start CL Generator form (P93515V), 7-6  
 Quiz, running, 8-56  
 Quiz Answer Review form (P98534), 8-55

## R

Regenerating source code, 5-7  
 Remove logic module, 8-20  
 Remove Member (RM/M), 2-12  
 Rename a dialogue, 8-53  
 Resequence logic module, 8-20  
 Reviewing questions, 8-45  
 Reviewing source modifications, 8-24  
 Running a dialogue, 8-54  
 Running a quiz, 8-56

## S

Sample Program – Item Mtc form (P92801), 3-37  
 SAR. *See* Software Action Request  
 SAR options, 2-19  
 Screen function keys, 1-3  
 Screen or report  
   accessing, 7-12

  compiling, 7-13  
 Selecting data fields, 7-11  
 Selecting the quick start application definition, 7-9  
 Selection exits, 3-46  
   view, 7-27  
 Selections, 8-10  
 Setup screen, processing options, 3-63  
 Signing on and off, 1-1  
 Simple Question & Answer form (P98529), 8-45  
 Simple question and answer, menu G9364, option 3, 8-45  
 Software Action Request. *See* SAR  
 Software Versions Repository form (P9801), 2-16, 3-28  
 Solving generation problems, 5-10  
 Source code  
   regenerating, 5-7  
   when to regenerate, 5-7  
 Source code for copy modules, 2-8  
 Source code for J.D. Edwards files, 2-8  
 Source code inventory and database, 8-1  
 Source from specifications, generating, 3-28  
 Source modifications, 5-1  
   reviewing, 8-24  
 Source sequence line numbers, understanding, 8-3  
 Source sequence line structure, 8-4  
 Source serial numbers, 8-3  
 Special characters, 3-34  
   within help instructions, 3-35  
 Statements, PDL, 4-5  
 Steps of quick start, 7-8  
 Structure of the serial number, 8-5  
 Submitting the program to compile, 7-15  
 Substitution directives, 8-37  
 System integration, 1-5  
   application development cycle, 1-5  
   fundamentals, 1-6  
   history of program generator, 1-6  
   specifications, 1-6

## T

Terms and concepts, 1-9  
 Training environment, 1-16  
   signon naming conventions, 1-16

student library setup, 1–16  
Translation Table form (P92710), 7–28

## U

UDC. *See* User Defined Codes  
Understanding assignments, 4–10  
    operator and syntax, 4–10  
    rules, 4–10  
Understanding blocks of statements, 4–8  
    keywords and syntax, 4–8  
    rules, 4–8  
Understanding calls  
    keywords and syntax, 4–13  
    rules, 4–13  
Understanding CASE profiles, 2–14  
Understanding Comments, 4–9  
    keywords and syntax, 4–9  
    rules, 4–9  
Understanding conditions  
    keywords and syntax, 4–15  
    rules, 4–16  
    symbols, 4–15  
Understanding database operations  
    keywords and syntax, 4–11  
    rules, 4–12  
Understanding directives, 8–29  
Understanding full data field parameters,  
3–47  
Understanding loops  
    keywords and syntax, 4–14  
    rules, 4–14  
Understanding miscellaneous keywords and  
syntax  
    keywords and syntax, 4–18  
    rules for include, 4–18  
    rules for return, 4–20  
Understanding source sequence line  
numbers, 8–3  
    source sequence line structure, 8–4  
    source serial numbers, 8–3  
    structure of the serial number, 8–5  
Understanding the File Specification form,  
3–21  
Updating the glossary, 7–17  
Updating the help file, 3–36  
User Defined Code Revisions form  
(P00051), 2–6, 2–7, 2–8

User Defined Codes (UDC). *See* UDC  
User provided prerequisites, working with,  
2–9  
Using CASE specifications inquiry, 8–26  
Using logic module cross reference, 8–17  
Using logic module op codes, 8–19  
Using program generator updates, 8–25

## V

VCO description fields, loading, 3–51  
Viewing an action diagram, 7–23

## W

What are file specifications?, 3–23  
What are the cursor sensitive function key  
exits?, 7–26  
What are the selection exits, 7–27  
When to regenerate source code, 5–7  
Work fields, A–4  
Working with file specifications, 3–19  
Working with J.D. Edwards provided  
prerequisites, 2–3  
Working with the detailed programming  
facility, 3–43  
Working with the question and answer  
system, 8–43  
Working with user provided prerequisites,  
2–9  
    development libraries, 2–10  
    job queues, 2–12  
    multi-member source file, 2–10  
    project management, 2–13  
World CASE Q & A form (P9364), 8–44



## Exercises

