

JD Edwards World

Computer Aided Software Engineering Guide

Release A9.3

E21955-02

April 2013

Copyright © 2013, Oracle and/or its affiliates. All rights reserved.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Contents

Preface	xv
Audience	xv
Documentation Accessibility	xv
Related Documents	xv
Conventions	xv

1 Overview to Computer Aided Software Engineering (CASE)

1.1	System Integration	1-1
1.1.1	Specifications	1-2
1.1.2	Fundamentals	1-2
1.1.3	History of the Program Generator	1-2
1.1.4	Evolution of the Program Generator	1-2
1.2	Features	1-3
1.2.1	What are the Benefits of CASE?	1-3
1.3	Terms and Concepts	1-4
1.3.1	CASE, as an industry term	1-4
1.3.2	CASE, as a JD Edwards World term	1-4
1.4	Detailed Information	1-4
1.4.1	CASE Profile	1-5
1.4.2	Computer Assisted Design (CAD)	1-5
1.4.3	Computer Assisted Programming (CAP)	1-5
1.4.4	About The Program Generator	1-5
1.4.5	About Program Types	1-6
1.4.6	About Master Source Code	1-6
1.4.7	About Program Specifications	1-6
1.4.8	Program Types	1-6
1.5	Menu Overview	1-7

Part I Foundation

2 Foundation

2.1	About Foundation Information	2-1
-----	------------------------------------	-----

3 Work with Prerequisites JD Edwards World Provides

3.1	Working with Prerequisites JD Edwards World Provides	3-1
3.1.1	Program Generator Files.....	3-1
3.1.2	Common UDCs	3-2
3.1.3	Source Code for Copy Modules.....	3-5
3.1.4	Source Code for JD Edwards World Files	3-5

4 Work with User-Provided Prerequisites

4.1	Development Libraries.....	4-1
4.2	Multi-member Source File (JDESRC)	4-1
4.3	Job Queues	4-3
4.4	Project Management	4-3
4.5	CASE Profiles.....	4-4
4.5.1	Function Exits.....	4-8
4.6	Object Authorities	4-8
4.6.1	Job Control Authority	4-8
4.6.2	Source Library	4-8
4.6.3	Source File.....	4-8
4.6.4	Job Queues	4-8

Part II Program Generator

5 Overview to Program Generator

5.1	About Program Generator Steps	5-1
-----	-------------------------------------	-----

6 Access Program Generator Specifications

6.1	Accessing Program Generator Specifications	6-1
6.2	Function Exits	6-3

7 Define Program Purpose and Type

7.1	Defining Program Purpose and Type	7-1
7.2	Function Exits	7-8

8 Work with File Specifications

8.1	What Are File Specifications?.....	8-1
8.2	Function Exits	8-8

9 Define General Instructions

9.1	About Special Characters.....	9-1
9.2	Special Characters within Help Instructions	9-2

10 Define Option and Function Exits

10.1	Defining Option and Function Exits	10-1
10.2	Function Exit.....	10-5

10.2.1	What You Should Know About.....	10-5
11	Work with the Detailed Programming Facility	
11.1	About the Detailed Programming Facility	11-1
11.1.1	Available Options.....	11-3
11.1.2	Function Exits.....	11-3
11.2	About Full Data Field Parameters	11-4
11.2.1	Primary Uses of Full Data Field Parameters.....	11-4
11.2.2	Function Exits.....	11-8
11.3	Loading VC0 Description Fields.....	11-9
11.3.1	Example.....	11-9
11.3.2	Example: User Defined Code.....	11-9
11.4	Enabling the Database Update Function for Subfiles	11-11
11.5	Creating *ENTRY PLIST Entries.....	11-12
11.5.1	Example.....	11-12
11.6	Protecting Fields from Being Cleared	11-14
11.6.1	What You Should Know About.....	11-15
11.7	Disabling Data Dictionary Edits	11-16
11.8	Creating a Partial KLIST for a File.....	11-17
11.8.1	Example.....	11-17
12	Define Processing Options	
12.1	Overview	12-1
12.1.1	What You Should Know About.....	12-2
12.2	Example - Interactive Programs Using Processing Options.....	12-2
12.3	Example - Report Program Using Processing Options	12-4
12.4	Defining Processing Options.....	12-4
12.4.1	Function Exits.....	12-8
Part III Program Design Language		
13	Overview to Program Design Language	
13.1	Objectives	13-1
13.2	About PDL	13-1
14	About PDL Statements and Syntax	
14.1	About PDL Statements	14-1
14.1.1	Keywords.....	14-2
14.1.2	Variables.....	14-2
14.1.3	Database Files.....	14-2
14.1.4	Operators	14-2
14.1.5	Constants.....	14-3
14.1.6	Punctuation.....	14-3
14.2	About Blocks of Statements.....	14-3
14.2.1	Keywords and Syntax	14-3

14.2.2	Rules	14-3
14.3	About Comments	14-4
14.3.1	Keywords and Syntax	14-4
14.3.2	Rules	14-4
14.4	About Assignments	14-5
14.4.1	Operator and Syntax	14-5
14.4.2	Rules	14-5
14.5	About Database Operations	14-5
14.5.1	Keywords and Syntax	14-5
14.5.2	Rules	14-6
14.6	About Program Calls	14-7
14.6.1	Keywords and Syntax	14-7
14.6.2	Rules	14-7
14.7	About Loops	14-7
14.7.1	Keywords and Syntax	14-7
14.7.2	Rules	14-8
14.8	About Conditions.....	14-8
14.8.1	Keywords and Syntax	14-8
14.8.2	Symbols	14-9
14.8.3	Rules	14-9
14.9	About Miscellaneous Keywords and Syntax	14-10
14.9.1	Keywords and Syntax	14-10
14.9.2	Rules for Include	14-10
14.9.3	Rules for Return	14-12

15 Understand Additional PDL Operations

15.1	Editing	15-1
15.2	Parsing	15-1
15.3	Source Code Generation	15-1
15.3.1	Data Item Formula Examples	15-2
15.3.1.1	Example User Defined PDL	15-2
15.3.1.2	Example Subfile Field	15-3
15.4	Add PDL to a Field	15-4
15.5	Function Exits	15-5

Part IV Source Modifications

16 Overview to Source Modifications

16.1	Objectives	16-1
16.2	About Source Modifications.....	16-1

17 Change Generated Source Code

17.1	Pre-SEU and Post-SEU Process	17-1
------	------------------------------------	------

18 Regenerate Source Code

18.1	When to Regenerate Source Code	18-1
------	--------------------------------------	------

18.2	Changing CAP Status	18-2
18.3	Resolving CAP Status Invalid Error.....	18-3
19	Work with Model Control Language Programs	
19.1	Working with CL Models	19-1
19.2	JD Edwards World Model CL Programs.....	19-2
Part V	CASE Programs	
20	Overview to CASE Programs	
20.1	Objectives	20-1
20.2	About CASE Programs.....	20-1
21	Overview to Subfile Inquiry Programs	
21.1	Program Type Description	21-1
21.2	Display File Definition	21-1
21.3	CL Program Definition.....	21-1
21.4	File Specifications.....	21-2
21.5	Detailed Programming Facility.....	21-2
21.6	Special Considerations	21-2
21.7	Quick Start Generation.....	21-2
22	Overview to Subfile Maintenance Programs	
22.1	Program Type Description	22-1
22.2	Display File Definition	22-1
22.3	CL Program Definition.....	22-2
22.4	File Specifications.....	22-2
22.5	Detailed Programming Facility.....	22-2
22.6	Special Considerations	22-2
22.7	Quick Start Generation.....	22-2
23	Create Report Programs	
23.1	Understanding RDA Special Use Fields.....	23-1
23.2	Creating a Total Format	23-3
23.3	Defining a Subheading.....	23-6
23.4	Understanding DREAM Writer Considerations	23-7
Part VI	Additional Tools	
24	Overview to Additional Tools	
24.1	Objectives	24-1
24.2	About Additional Tools	24-1

25	Work with Quick Start CL Generator	
25.1	Working with Quick Start CL Generator	25-1
26	Work with the Quick Start Application Tool	
26.1	Quick Start Process	26-2
26.2	Defining the Application	26-2
26.3	Selecting Data Fields.....	26-4
26.4	Browsing or Updating the Screens or Reports (Optional)	26-5
26.5	Compiling the Screens or Report (Optional)	26-6
26.6	Modifying Specifications (Optional)	26-7
26.7	Submitting the Program to Compile (Optional).....	26-8
26.8	Updating the Data Dictionary and Glossary	26-9
27	Work with Action Diagramming	
27.1	Building an Action Diagram	27-1
27.2	Viewing an Action Diagram.....	27-2
27.2.1	Function Exits.....	27-3
27.2.2	Cursor Sensitive Function Exits.....	27-4
27.2.3	Option Field Values.....	27-5
27.3	Accessing the Logic Translation Feature.....	27-5
Part VII Source Code Inventory and Database		
28	Overview to Source Code Inventory and Database	
28.1	Objectives	28-1
28.2	About the Source Code Inventory and Database.....	28-1
29	Understand Source Sequence	
29.1	Source Serial Numbers	29-1
29.2	Source Sequence Line Structure.....	29-1
29.3	Structure of the Serial Number	29-2
29.3.1	Source Inventory Master File (F93001) - XXXX	29-2
29.3.2	Generation Execution - YYYY	29-2
29.3.3	User Change in SEU - ZZZZ	29-2
30	Working with Program Types	
30.1	Reviewing Abbreviations for Program Types	30-1
30.2	Reviewing Program Types Index	30-2
30.2.1	Available Options.....	30-2
30.3	Reviewing Program Types Cross Reference.....	30-3
30.4	Creating or Modifying Program Types	30-4
30.4.1	Program Type.....	30-4
30.4.2	Primary Module.....	30-4
30.4.3	Glossary K.....	30-4

31 Work with Logic Modules

31.1	Primary Logic Modules.....	31-1
31.2	Detail Logic Modules	31-2
31.3	Generation Options	31-2
31.3.1	Help Instructions Edit/Build	31-2
31.3.2	All Help Instructions.....	31-2
31.3.3	Global Program Regeneration	31-2
31.4	Viewing the Logic Module Index	31-3
31.5	Viewing Logic Module Cross Reference	31-3
31.6	Viewing Logic Module Op Codes	31-4
31.7	Maintaining the Logic Module File	31-5
31.7.1	Resequence Logic Module.....	31-5
31.7.2	Remove Logic Module	31-5
31.8	Creating or Modifying Logic Modules	31-6
31.9	Creating or Modifying Formula Library Entry	31-7
31.10	Copying or Moving Program Specifications.....	31-8
31.11	Printing Program Generator Specifications	31-9
31.12	Reviewing Source Modifications	31-10
31.13	Using Program Generator Updates.....	31-11
31.14	Using CASE Specifications Inquiry	31-11

32 Understand Directives

32.1	Functional Directives.....	32-1
32.2	Substitution Directives	32-7
32.3	Exception Directives	32-10
32.4	Conditional Directives	32-11

33 Work with the Question and Answer System

33.1	About Simple Question & Answer.....	33-1
33.2	Reviewing Questions in a Master Dialogue.....	33-1
33.3	Adding New Q & A Dialogue.....	33-3
33.4	Working with an Existing Dialogue.....	33-6

34 Create User Defined PDL

34.1	Creating User Defined PDL.....	34-1
------	--------------------------------	------

A Program Generator Checklist

A.1	Data File Design Aid	A-1
A.2	Screen Design Aid.....	A-1
A.3	Report Design Aid	A-2
A.4	Program Generator	A-2

B Programming Standards

B.1	Error Handling	B-1
-----	----------------------	-----

B.2	Indicator Usage	B-2
B.3	Naming Conventions	B-3
B.4	Key List (KLIST)	B-3
B.5	Work Fields	B-4
B.6	Current Date and Time	B-4

C CASE Program Types

C.1	Guidelines	C-2
C.2	A0010 - Interactive Subfile Inquiry	C-2
C.2.1	Description	C-2
C.2.2	Display File Definition	C-2
C.2.3	CL Program Definition	C-2
C.2.4	File Specifications	C-3
C.2.5	Detailed Programming Facility	C-3
C.2.6	Special Considerations	C-3
C.2.7	Quick Start Generation	C-3
C.3	A0020 - Interactive Single Record Inquiry	C-3
C.3.1	Description	C-3
C.3.2	Display File Definition	C-3
C.3.3	CL Program Definition	C-3
C.3.4	File Specifications	C-3
C.3.5	Special Considerations	C-4
C.3.6	Quick Start Generation	C-4
C.4	B0010 - Interactive Single Record Maintenance	C-4
C.4.1	Description	C-4
C.4.2	Display File Definition	C-4
C.4.3	CL Program Definition	C-4
C.4.4	File Specifications	C-4
C.4.5	Detailed Programming Facility	C-4
C.4.6	Quick Start Generation	C-4
C.5	C0010 - Batch Report with Totals	C-4
C.5.1	Description	C-5
C.5.2	Printer File Definition	C-5
C.5.3	CL Program Definition	C-5
C.5.4	File Specifications	C-5
C.5.5	Special Considerations	C-5
C.5.6	Quick Start Generation	C-5
C.6	C0020 - Batch Report with Totals and Subheadings	C-5
C.6.1	Description	C-5
C.6.2	Printer File Definition	C-6
C.6.3	CL Program Definition	C-6
C.6.4	File Specifications	C-6
C.6.5	Special Considerations	C-6
C.6.6	Quick Start Generation	C-6
C.7	C0025 - Batch Report with Totals and Subheadings	C-6
C.7.1	Description	C-6
C.7.2	Printer File Definition	C-6

C.7.3	CL Program Definition	C-7
C.7.4	File Specifications	C-7
C.7.5	Special Considerations	C-7
C.7.6	Quick Start Generation	C-7
C.8	D0010 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number	C-7
C.8.1	Description.....	C-7
C.8.2	Display File Definition	C-7
C.8.3	CL Program Definition	C-8
C.8.4	File Specifications	C-8
C.8.5	Detailed Programming Facility	C-8
C.8.6	Special Considerations	C-8
C.8.7	Quick Start Generation	C-8
C.9	D0020 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number	C-9
C.9.1	Description.....	C-9
C.9.2	Display File Definition	C-9
C.9.3	CL Program Definition	C-9
C.9.4	File Specifications	C-9
C.9.5	Detailed Programming Facility	C-9
C.9.6	Special Considerations	C-10
C.9.7	Quick Start Generation	C-10
C.10	D0030 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record	C-10
C.10.1	Description.....	C-10
C.10.2	Display File Definition	C-10
C.10.3	CL Program Definition	C-10
C.10.4	File Specifications	C-10
C.10.5	Detailed Programming Facility	C-11
C.10.6	Special Considerations	C-11
C.10.7	Quick Start Generation	C-11
C.11	D0040 - Interactive Subfile Maintenance with Action Code, with Options, by Key	C-11
C.11.1	Description.....	C-11
C.11.2	Display File Definition	C-11
C.11.3	CL Program Definition	C-12
C.11.4	File Specifications	C-12
C.11.5	Detailed Programming Facility	C-12
C.11.6	Special Considerations	C-12
C.11.7	Quick Start Generation	C-12
C.12	D0050 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number	C-12
C.12.1	Description.....	C-12
C.12.2	Display File Definition	C-12
C.12.3	CL Program Definition	C-13
C.12.4	File Specifications	C-13
C.12.5	Detailed Programming Facility	C-13
C.12.6	Special Considerations	C-13
C.12.7	Quick Start Generation	C-14

C.13	D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key	C-14
C.13.1	Description.....	C-14
C.13.2	Display File Definition	C-14
C.13.3	CL Program Definition	C-14
C.13.4	File Specifications	C-14
C.13.5	Detailed Programming Facility	C-14
C.13.6	Special Considerations	C-14
C.13.7	Quick Start Generation	C-15
C.14	D0070 - Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number	C-15
C.14.1	Description.....	C-15
C.14.2	Display File Definition	C-15
C.14.3	CL Program Definition	C-15
C.14.4	File Specifications	C-15
C.14.5	Detailed Programming Facility	C-15
C.14.6	Special Considerations	C-16
C.14.7	Quick Start Generation	C-16
C.15	D0080 - Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number	C-16
C.15.1	Description.....	C-16
C.15.2	Display File Definition	C-16
C.15.3	CL Program Definition	C-16
C.15.4	File Specifications	C-17
C.15.5	Detailed Programming Facility	C-17
C.15.6	Special Considerations	C-17
C.15.7	Quick Start Generation	C-17
C.16	D0090 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance	C-17
C.16.1	Description.....	C-17
C.16.2	Display File Definition	C-18
C.16.3	CL Program Definition	C-18
C.16.4	File Specifications	C-18
C.16.5	Detailed Programming Facility	C-18
C.16.6	Special Considerations	C-18
C.16.7	Quick Start Generation	C-18
C.17	D0100 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key	C-19
C.17.1	Description.....	C-19
C.17.2	Display File Definition	C-19
C.17.3	CL Program Definition	C-19
C.17.4	File Specifications	C-19
C.17.5	Detailed Programming Facility	C-19
C.17.6	Special Considerations	C-19
C.17.7	Quick Start Generation	C-19
C.18	E0010 - Interactive Window	C-20
C.18.1	Description.....	C-20
C.18.2	Display File Definition	C-20
C.18.3	CL Program Definition	C-20

C.18.4	File Specifications	C-20
C.18.5	Define Option and Function Key Exits.....	C-20
C.18.6	Detailed Programming Facility	C-20
C.18.7	Special Considerations	C-21
C.18.8	Quick Start Generation	C-21
C.19	X0010 - Batch Update with Report	C-21
C.19.1	Description.....	C-21
C.19.2	Printer File Definition.....	C-21
C.19.3	CL Program Definition	C-21
C.19.4	File Specifications	C-21
C.19.5	Special Considerations	C-21
C.19.6	Quick Start Generation	C-22
C.20	X0020 - Batch Update	C-22
C.20.1	Description.....	C-22
C.20.2	Printer File Definition.....	C-22
C.20.3	CL Program Definition	C-22
C.20.4	File Specifications	C-22
C.20.5	Special Considerations	C-22
C.20.6	Quick Start Generation	C-22
C.21	X0030 - Batch Update with Subroutine S001.....	C-23
C.21.1	Description.....	C-23
C.21.2	Printer File Definition.....	C-23
C.21.3	CL Program Definition	C-23
C.21.4	File Specifications	C-23
C.21.5	Special Considerations	C-23
C.21.6	Quick Start Generation	C-23
C.22	X0040 - Batch Update with Report	C-23
C.22.1	Description.....	C-23
C.22.2	Printer File Definition.....	C-24
C.22.3	CL Program Definition	C-24
C.22.4	File Specifications	C-24
C.22.5	Special Considerations	C-24
C.22.6	Quick Start Generation	C-24
C.23	Y0010 - Conversion, Two Files with Error Report	C-24
C.23.1	Description.....	C-24
C.23.2	Printer File Definition.....	C-25
C.23.3	CL Program Definition	C-25
C.23.4	File Specifications	C-25
C.23.5	Special Considerations	C-25
C.23.6	Quick Start Generation	C-25
C.24	Y0020 - Conversion, One File Update with Error Report	C-25
C.24.1	Description.....	C-25
C.24.2	Printer File Definition.....	C-26
C.24.3	CL Program Definition	C-26
C.24.4	File Specifications	C-26
C.24.5	Special Considerations	C-26
C.24.6	Quick Start Generation	C-26

C.25	Y0030 - Conversion, One File Write with Error Report.....	C-26
C.25.1	Description.....	C-26
C.25.2	Printer File Definition.....	C-26
C.25.3	CL Program Definition	C-27
C.25.4	File Specifications	C-27
C.25.5	Special Considerations	C-27
C.25.6	Quick Start Generation	C-27

D Source Listings

D.1	Program Status Data Structure - I00DSPROG	D-1
D.2	Copy Module - Retrieve Soft Coding Data Structure - I00SC	D-2
D.3	Item Master Information - P928011	D-9

E JD Edwards World Subroutines and Flows

E.1	Subroutines	E-1
E.2	Flows.....	E-2
E.2.1	Interactive Non-Subfile Program	E-3
E.2.2	Subfile Program with Options	E-4
E.2.3	Report Program without Subheadings.....	E-5
E.2.4	Report Program with Subheadings.....	E-6

F Sample Code

G Functional Servers

G.1	Example: Voucher Processing Functional Server.....	G-2
-----	--	-----

Index

Preface

Welcome to the JD Edwards World Computer Aided Software Engineering Guide.

Audience

This guide is intended for implementers and end users of JD Edwards World Computer Aided Software Engineering.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

You can access related documents from the JD Edwards World Release Documentation Overview pages on My Oracle Support. Access the main documentation overview page by searching for the document ID, which is 1362397.1, or by using this link:

- <https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=1362397.1>

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Overview to Computer Aided Software Engineering (CASE)

This chapter contains these topics:

- [Section 1.1, "System Integration,"](#)
- [Section 1.2, "Features,"](#)
- [Section 1.3, "Terms and Concepts,"](#)
- [Section 1.4, "Detailed Information,"](#)
- [Section 1.5, "Menu Overview."](#)

1.1 System Integration

Computer Aided Software Engineering (CASE) covers the entire application development life cycle, including:

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

You use these tools to develop, operate, and maintain flexible, business application software.

Application Development Cycle

There are three technical levels in the JD Edwards World Application Development Cycle (A/D Cycle):

Level	A/D Cycle
Level 1	The Application Platform, which represents the Technical Foundation Guide.
Level 2	The Design Platform, which represents the Advanced Programming Concepts and Skills (APCS) Guide.

Level	A/D Cycle
Level 3	The Development Platform, which represents the Program Generator (CASE) Guide.

1.1.1 Specifications

You define a program using various Program Generator specifications. You perform the following:

- Define program purpose and type
- Specify the files
- Create help text
- Define function exits and options
- 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, adding the correct validation files and servers to complete the program.

1.1.2 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. You can add AAIs to your programs and create Control Language (CL) programs to launch programs from menu options.

1.1.3 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

1.1.4 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 (PDL) allow you to make modifications instead of making changes through Source Entry Utility

1.2 Features

JD Edwards World provides several tools to help create and customize your programs.

- PDL is available to add field-specific logic to your programs.
- Quick Start asks a few basic questions, and then creates a basic Report Program Generator (RPG) or CL program.
- JD Edwards World provides many different utilities to assist you in creating and maintaining your code.
- PDL enables you to add calculations or comparisons to specific fields within the program.
- Precompiler commands are available to specialize your compile environment

CASE includes the following features:

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

Note: You cannot use the Program Generator to modify existing JD Edwards World programs.

1.2.1 What are the Benefits of CASE?

Every program you create using the Program Generator automatically includes and uses JD Edwards World functionality, such as:

- Data Dictionary
- User defined codes
- Vocabulary overrides
- Action code security
- Business unit security
- Standard function exits
- Function exit and option exit security

- Cursor sensitive help
- Program help
- DREAM Writer
- Processing options

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

The Program Generator is the same tool that JD Edwards World uses to generate the JD Edwards World application programs.

You can create simple programs in a short period of time using the Program Generator. Due to the standard structure and subroutines of the programs you generate, it is easier to incorporate complexities in either the Program Specifications or the source code.

You generate the source RPG code from Program Specifications, Program Types, and Master Source Code; therefore, you can regenerate the source as JD Edwards World enhances the functionality of its software. Because the enhanced functionality is in the Master Source Code File, you need to regenerate only the source code using the original Program Specifications.

Note: The JD Edwards Program Generator output is in RPGIII code. When you complete the program development cycle, you can use RPGIII to RPGIV converters from IBM and third party vendors. As of A7.3.14, A8.1.5 and A9.1, you can use the Software Versions Repository to manage RPGIV programs with the Function Code set to RPGL.

1.3 Terms and Concepts

1.3.1 CASE, as an industry term

As an industry term, CASE is an acronym for Computer-Aided Software Engineering. Many suppliers offer tools that implement various aspects of software engineering. These tools are either 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.

1.3.2 CASE, as a JD Edwards World term

As a JD Edwards World term, CASE refers to a set of tools that you use in the software development process. Following are the components of the CASE tools.

1.4 Detailed Information

1.4.1 CASE Profile

See CASE Profiles *JD Edwards World Advanced Programming Concepts and Skills Guide* for information on setting up the CASE profile. There are several methods to access the CASE profile:

- Choose CASE profile from the Computer Aided Design menu (G92)
- Choose Software Versions Repository from the Computer Aided Design menu (G92), and then choose Repository Services (F6) to access the Repository Services window.

You enter the CASE program source generation file and compiler options on this screen.

1.4.2 Computer Assisted Design (CAD)

CAD includes the following:

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

1.4.3 Computer Assisted Programming (CAP)

CAP includes the following:

- 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 through J98MODEL8)
 - Quick Start CL Generator
- Quick Start Application Tool

1.4.4 About The Program Generator

The Program Generator is the JD Edwards World tool that generates source code for both RPG programs and CL programs. In many respects the Program Generator is a very simple tool that combines three functions and produces the source code as a result of combining the functions. The three functions are:

- Program Types
- Master Source Code
- Program Specifications

1.4.5 About Program Types

The Program Generator builds software depending on the program type you choose. The program types combine the features of:

- Interactive, for example, screen
- Batch, for example report or conversion
- Single record or multi-record

These program types contain a list of individual definitions which you combine to form a functional program. JD Edwards World refers to individual definitions as primary logic modules and the system uses them to build the source code for the program type. The system stores each primary logic module in the Master Source Code File. These logic modules are the components of all JD Edwards World program types.

1.4.6 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 JD Edwards World directives, which the Program Generator interprets and replaces with RPG code. The interpretation of the directives is based on the Program Specifications that you establish for generating a specific program type.

1.4.7 About Program Specifications

To generate a program, you must first complete the Program Generator Specifications. These specifications are the details of your program that the Program Generator uses to complete the RPG code it builds from the master source directives. There are six specifications. You must determine:

- A program type
- The files the program uses

After you specify this information, you can generate complete source code that you can then compile and execute.

1.4.8 Program Types

There are five categories of program types:

Interactive

- Can be either update or inquiry
- Can contain Action Codes
- Can contain a subfile
- Can interface with DREAM Writer to provide run-time options

Window

- Normally includes cursor sensitive helps (F1)
- Fits inside current interactive program

Report

- Provides for accumulated values (totals)

- Interfaces with DREAM Writer
- Can contain sub-headings

Server

- Name includes a prefix of X
- Updates master files
- Can contain a report

Conversion

- Use this program type to convert data from one file to another
- Can contain a report

JD Edwards World currently provides 25 pre-defined program types with the Program Generator. The CASE Guide includes information to modify existing program types and to create of your own program types.

1.5 Menu Overview

JD Edwards World systems are menu driven. System functions are organized according to their function and frequency of use. The options on these screens illustrate the flow to the functions in this guide.

Figure 1-1 Master Directory screen

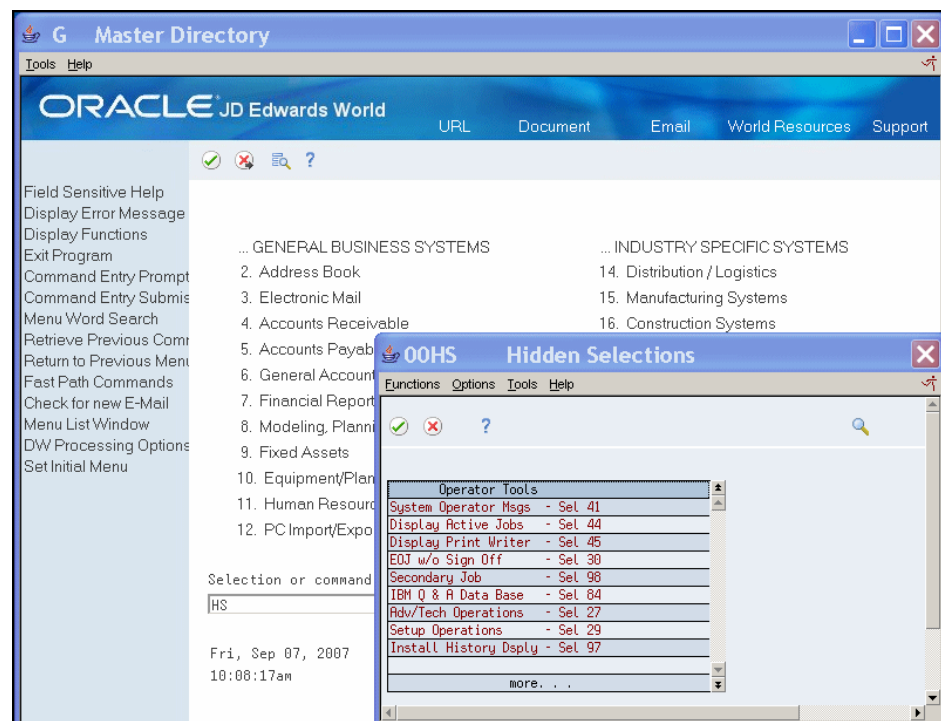


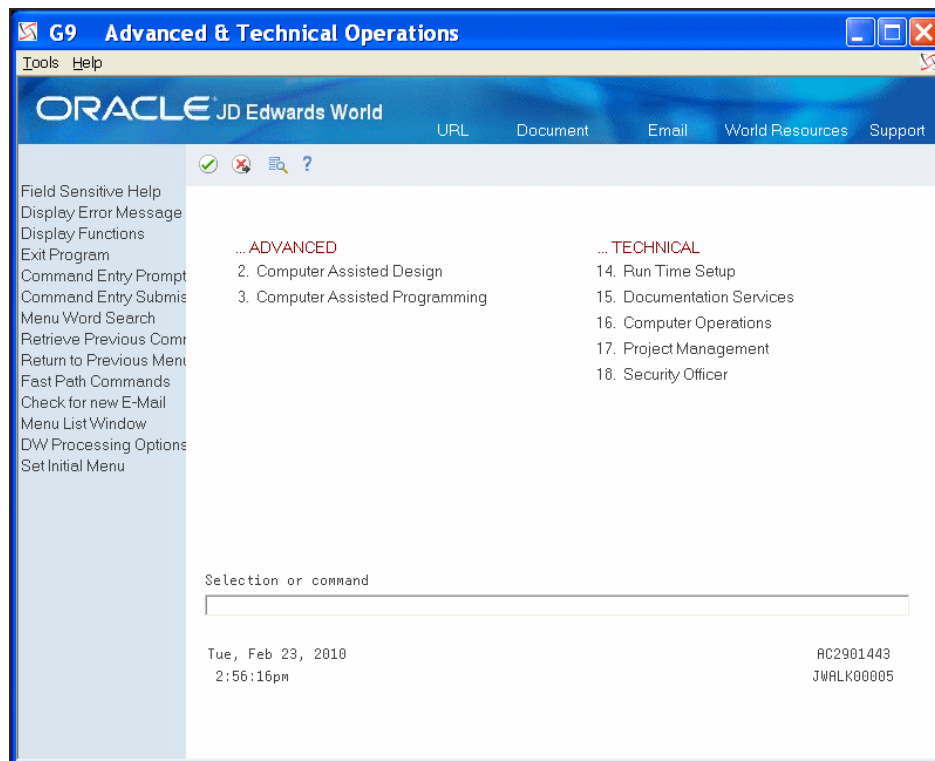
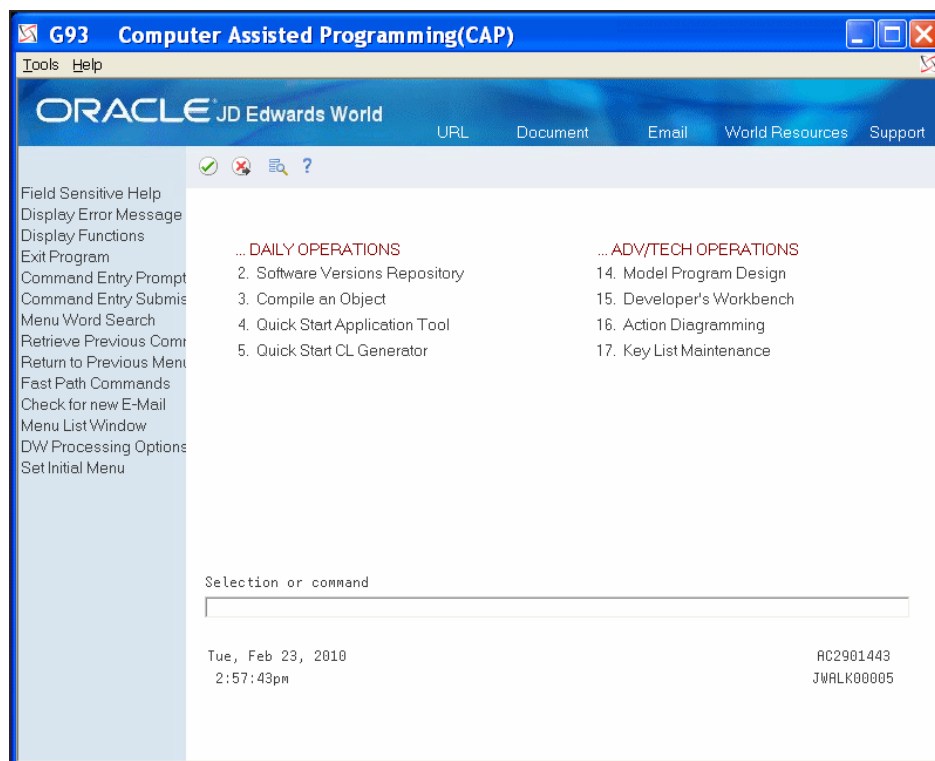
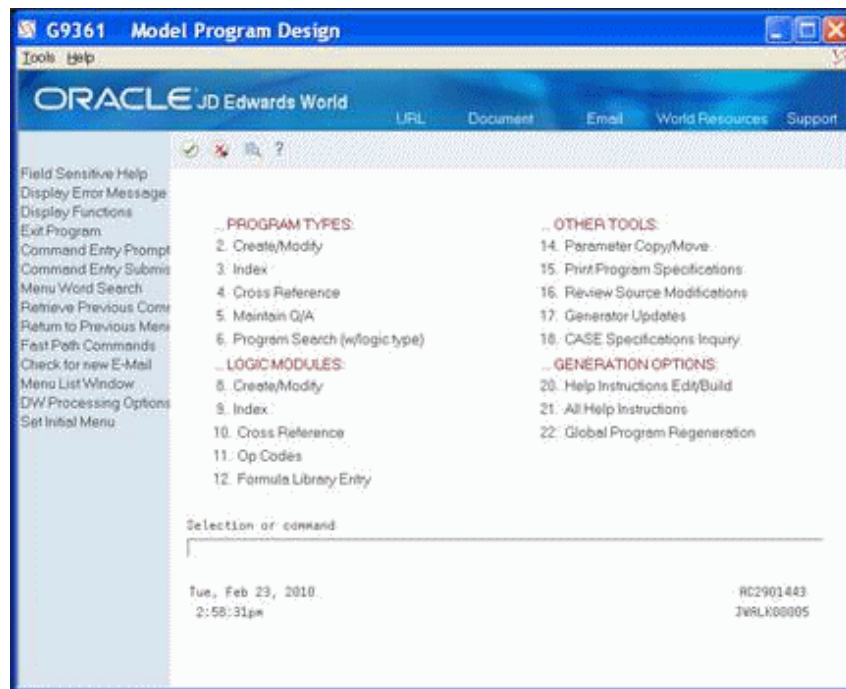
Figure 1–2 Advanced & Technical Operations screen**Figure 1–3 Computer Assisted Programming (CAP) screen**

Figure 1-4 Model Program Design screen



Part I

Foundation

This part contains these chapters:

- [Chapter 2, "Foundation,"](#)
- [Chapter 3, "Work with Prerequisites JD Edwards World Provides,"](#)
- [Chapter 4, "Work with User-Provided Prerequisites."](#)

This chapter contains the topic:

- [Section 2.1, "About Foundation Information."](#)

2.1 About Foundation Information

Before the Program Generator can successfully generate source code, a number of foundation items need to be in place. JD Edwards World provides some of these and you must verify they exist. You must perform additional prerequisites.

JD Edwards World provides the following prerequisites:

- Program Generator Files
- Common User Defined Codes
- Source Code for Copy Modules
- Source Code for JD Edwards World Files

You provide the following prerequisites:

- Development Libraries
- Multi-member Source File copied from F93002 (8 fields, 142 char record)
- Job Queues
- Project Management
- CASE Profiles
- Object Authorities

Work with Prerequisites JD Edwards World Provides

This chapter contains the topic:

- [Section 3.1, "Working with Prerequisites JD Edwards World Provides."](#)

3.1 Working with Prerequisites JD Edwards World Provides

The following are prerequisites JD Edwards World provides. You must verify their existence.

- Program Generator Files
- Common User Defined Codes (UDCs)
- Source Code for Copy Modules
- Source Code for JD Edwards World Files

3.1.1 Program Generator Files

The Program Generator files follow. Each has a specific function when the system generates a program. Some of these files include data; while others have no data. You need to verify that the files exist in your CASE environment and that they appropriately contain data or not.

Program Generator

The following two files are database files and include data.

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

Source Modifications/Helps

The Help/Modification Master (F93002) file is a multi-member source file, and includes an empty F93002 member.

Program Generator Specifications

The following files are database files and do not include data.

- Program Purpose and Type (F93101)
- File Specifications (F93102)
- File Formats (F93103)

- Selection/Function Exits (F93104)
- Detail Field Definitions (F93105)
- Automatic Accounting Instructions (F93106)

The DREAM Writer Master Parameter (F98301) file is a database file and includes processing options.

Program Design Language (PDL)

The following are database files. The Generation Operation Codes (F93108) file includes data; the other two files do not include data.

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

Q&A Dialogue

The following files are database files and include data.

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

3.1.2 Common UDCs

The Program Generator uses the following four UDCs:

- Logic Modules, 93/LM. Identifies the pieces of code within the Master Source Inventory file (F93001) that the system uses to create your RPG program.

Figure 3–1 General User Defined Codes screen

00051 General User Defined Codes

ORACLE JD Edwards World

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
More Details
User Defined Code Typ
Repository Services
Redisplay Previously C
Memo (Cursor Sensitive)
Where Used
Alternate Lang Desc (C)
Print User Defined Code
Clear Screen

00051 General User Defined Codes

System Code
User Defined Codes
Skip To Code
Primary Logic Modules

Action Code
See Memo
10 Character

Code	Description	Description-2
FILEDEFN01	File Specification	
FILEEXTN0	Tables & Arrays - STD Video	
FILEEXTN1	Tables & Arrays - SPL Video	
FILEEXTN2	Tables & Arrays - STD Rpt	
FILEEXTN3	Tables & Arrays - 2F - Conv	
FILEEXTN4	Tables & Arrays - Batch	
FILEEXTN5	Tables & Arrays - Windows	
FILEEXTN6	Tables & Arrays - 2F/Batch	
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	

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

- Common Subroutine Copy Members, 93/ /C. Lists all of the copy modules on the system. Description-2 field contains any additional copy modules that are necessary to make the common subroutine function properly. For example, C0012 requires copy module E0012.

Figure 3–2 General User Defined Codes (93//C) screen

00051 General User Defined Codes

ORACLE JD Edwards World

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
More Details
User Defined Code Typ
Repository Services
Redisplay Previously C
Memo (Cursor Sensitive)
Where Used
Alternate Lang Desc (C)
Print User Defined Code
Clear Screen

00051 General User Defined Codes

System Code
User Defined Codes
Skip To Code
Common Subroutine Copy Members

Action Code
10 Character

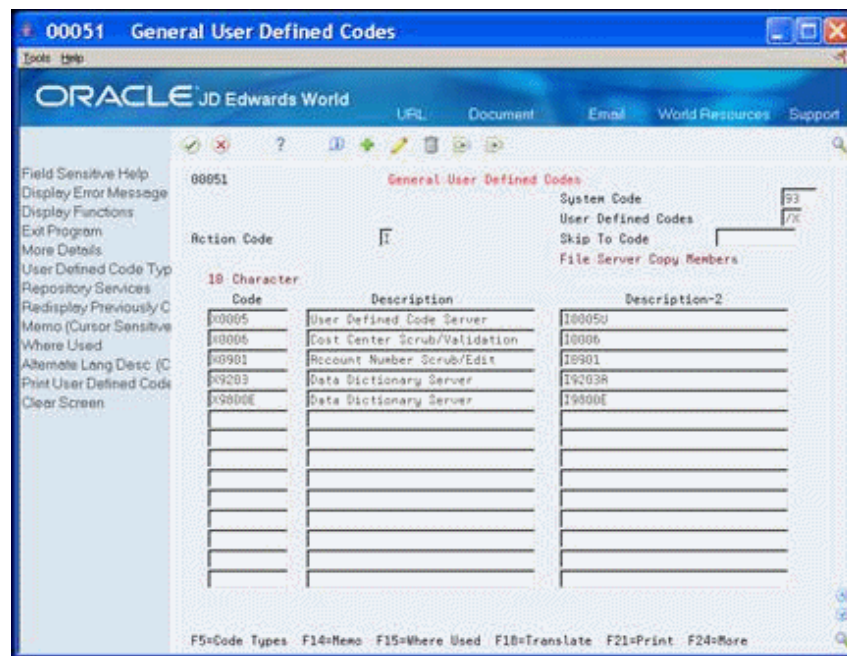
Code	Description	Description-2
C00RSC	Soft Coding Server - Reports	C,I
C00SC	Soft Coding Server - Videos	C,I
C0000	Cost Center Security Check	C
C0001	Edit Action Code	D,E,C
C0001A	Edit Action Code - Reg Inquiry	D,E,C
C0010	Next Numbering - Automatic	E,I,C
C0011	Center Descriptive Titles	E,C
C0012	Right Justify Numeric Fields	E,C
C0012N	Right Justify Numeric Fields -	C
C0015	Currency - Translate Video File	C
C00151	Currency - Translate Video File	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=Translate F21=Print F24=More

Alphanumeric Code	Type of Copy Module
D	Copy the member into the F specifications
E	Copy the member into the E specifications
I	Copy the member into the I specifications
C	Copy the member into the C specifications

- Servers, 93//X. This is a partial list of server programs and the associated copy member for each.

Figure 3–3 General User Defined Codes (93//X) screen



- Program Types, 93/PT. These are all the program types within the CASE tool.

Figure 3–4 General User Defined Codes (93/PT) screen

00051 General User Defined Codes

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
More Details
User Defined Code Type
Repository Services
Redisplay Previously C
Memo (Cursor Sensitive
Where Used
Alternate Lang Desc (C
Print User Defined Code
Clear Screen

00051 General User Defined Codes

System Code 93
User Defined Codes PT
Skip To Code
Program Type

Action Code I

10 Character

Code	Description
A00010	I
A00020	I
A00030	I
A00040	I
A00050	I
A00060	I
A00070	I
A00080	I
A00090	I
C00010	P
C00020	P
C00030	P
C00040	P
C00050	P
C00060	P
C00070	P
C00080	P
C00090	P
D00010	I
D00020	I
D00030	I
D00040	I
D00050	I
D00060	I
D00070	I
D00080	I
D00090	I

F5=Code Types F14=Memo F15=Where Used F16=Translate F21=Print F24=More

3.1.3 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 in file JDECPY in library JDFSRC.

3.1.4 Source Code for JD Edwards World Files

Source code for JD Edwards World database files must also be in the CASE environment. File source is in file JDESRC in library JDFSRC.

Work with User-Provided Prerequisites

This chapter contains these topics:

- [Section 4.1, "Development Libraries,"](#)
- [Section 4.2, "Multi-member Source File \(JDESRC\),"](#)
- [Section 4.3, "Job Queues,"](#)
- [Section 4.4, "Project Management,"](#)
- [Section 4.5, "CASE Profiles,"](#)
- [Section 4.6, "Object Authorities."](#)

There are several prerequisites that the user must provide. These prerequisites include the program developer being signed on to a JD Edwards World environment including QGPL in the library list and additional prerequisites.

4.1 Development Libraries

You must provide three types of libraries for CASE:

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

While it might be customary to create three different libraries for these purposes, this task is optional. Either of the following scenarios is acceptable.

Unique Libraries	Common Libraries
Source = DEVSRC	Source = DEVLIB
Object = DEVOBJ	Object = DEVLIB
Data = DEVDTA	Data = DEVLIB

4.2 Multi-member Source File (JDESRC)

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

- Be 142 bytes long to allow for the Program Generator serial number.
- Contain eight specific fields. For example, DSPFFD F93002.

To create the multi-member source file JDESRC

1. Enter the following on the Command Line:

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

You use the F93002 file because it is in the correct format to generate the program.

The To file can be any name and you can use JDESRC. However, JD Edwards World pristine source resides in JDESRC, so you can only use the JDESRC name if it is in a different library than the pristine JD Edwards World source library (JDESRC).

Do not use CRTSRCPF as it has only three fields in it, Date, Time, and Data, and the Program Generator requires extra fields.

Figure 4–1 Copy File (CPYF) screen

Copy File (CPYF)

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions

Type choices, press Enter.

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

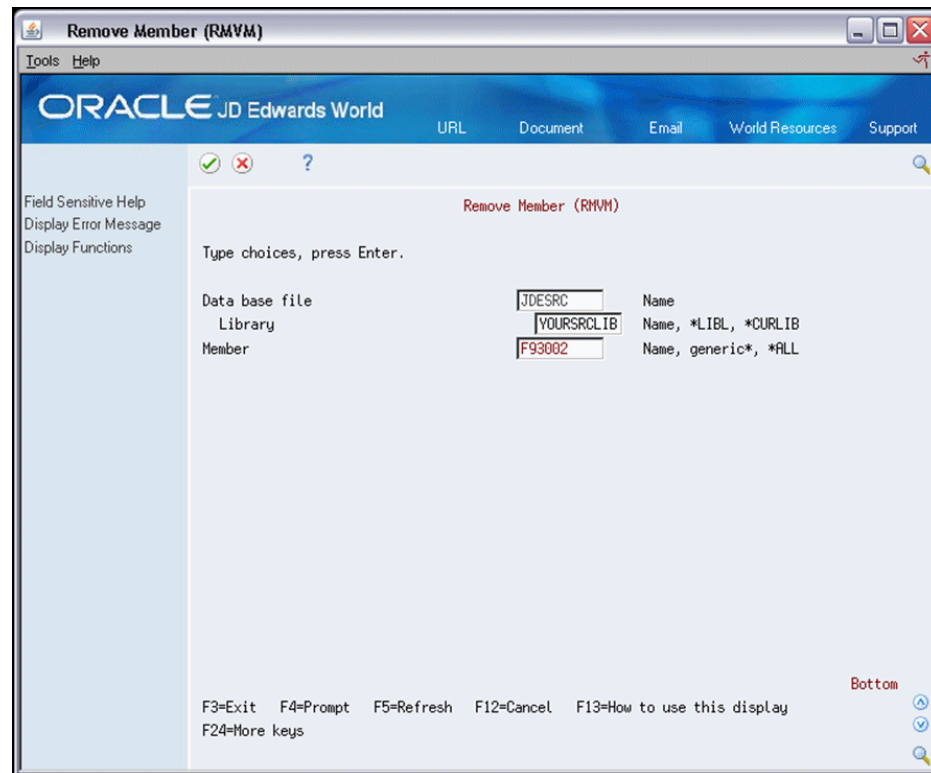
Bottom

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

After you create the JDESRC file, you can remove the empty member that the system adds.

2. Enter the following on the Command Line:

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

Figure 4–2 Remove Member (RMVM) screen

After you create the JDESRC file, you can remove the empty member that the system adds.

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

- The wrong length
- Improper formatting

4.3 Job Queues

By default, the system submits jobs to generate the program to the CLONE job queue, and the jobs to compile the program to the COMPILE job queue. If you want to use these default job queues, then you must create them and attach them to an existing subsystem.

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

4.4 Project Management

You must determine the following regarding Project Management:

1. Whether to manage CASE generated programs (or any development work) using the JD Edwards World Software Action Request System (SAR). JD Edwards World includes the SAR system as part of System 00, General Back Office, under the name of Work Order Processing.

- See About SAR System Setup in the *JD Edwards World Advanced Programming Concepts and Skills Guide* for more information about the JD Edwards World SAR System.
 - If you are going to use the JD Edwards World SAR System, you create a SAR before starting development or use the number of an existing SAR for development.
 - If you are not going to use the JD Edwards World SAR system, you can disable the function that allows the system to validate the SAR number by entering *NONE in the SAR Number field on the CASE Profiles screen.
2. If you use the JD Edwards World SAR System to manage software development, then you must determine whether to use 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. These include the Data Dictionary, UDCs, Files, Programs, Vocabulary Overrides, DREAM Writer, and Menus. SAR Logging allows you to identify what components you must move from your development environment to a testing environment and a production environment.

If you use SAR Logging, you must determine what method you use to link a SAR number with each piece of the development work. There are two ways to associate a SAR number with development:

- Use a default SAR number, which you use with all development work until you change the default number.
- Enter the SAR number as you perform the development work.

The results of your decisions reside in your *PUBLIC CASE Profiles.

4.5 CASE Profiles

CASE profiles are user-defined values that can pertain to individual users or to one default *PUBLIC user profile. The system:

- Stores information in the CASE Profiles file (F98009).
- Uses the profiles to define the overall CASE operating environment.

You define various processing control parameters, including:

- Default development libraries
- Compile job queue
- Program Generator source generation job queue
- Compile print options
- SAR logging options

The system overrides the *PUBLIC default values with the individual CASE profile values.

- You must complete all fields when entering information for *PUBLIC.
- You maintain default CASE Profile values in a record with the User ID *PUBLIC. Enter CASE Profile values for individual users only if you want to override the *PUBLIC values.
- You can leave all fields blank except for the specific values you want to override when entering values for individual users.

- The system uses the values in the record for User ID *PUBLIC as the defaults for all users unless individual user profiles have been set up.
- You cannot delete the *PUBLIC record.

The system uses the values in the SAR Number and SAR Delivery fields to determine what type of SAR logging should occur.

Value in the SAR Delivery field	SAR Logging
*NONE	No SAR logging at all.
*LOG	The system does not include a SAR number as part of the SAR logging.
*DFT	The system uses the SAR number you specify for the SAR logging.
*PROMPT	The system prompts you for a SAR number and revision notes when it creates an entry to the SAR log.

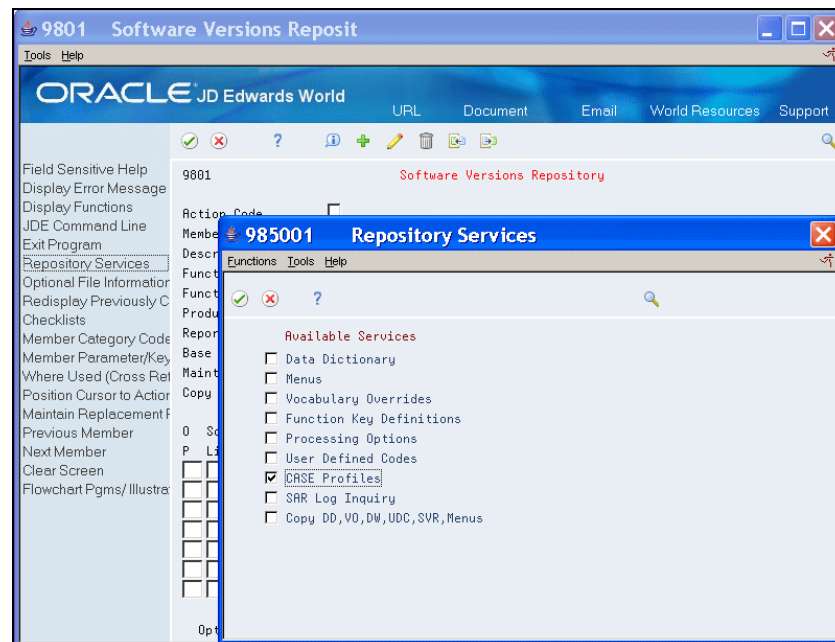
To access CASE profiles

Navigation

From Computer Assisted Design (G92), choose Case Profiles

Alternatively, on the Software Versions Repository screen, choose Repository Services. On the Repository Services window, choose Case Profiles and then click Enter.

Figure 4–3 Software Versions Repository screen



The program attempts to locate the CASE profile for your User ID. An error message displays on the CASE Profiles screen if your user ID is not set up.

1. On Case Profiles, locate the *PUBLIC user ID.
2. Enter your Used ID, complete any of the fields and click Change to create your record.

Figure 4–4 CASE Profiles screen

Field	Explanation
Source File	The Source File Name field contains the name of the file where the source for an object exists. In the program generator File Specifications this name is defaulted to "JDESRC". In combination with the source library name it identifies where the program generator places the generated source code. The system uses this file to find the source for each data file, display file or report file which it must analyze to create the data field parameters. As used in the automated installation processing file this is the source file of an object at the time the object was created.
Source Library	The default library where source will be stored. The source file specified must reside within this library.
Object Library	The default library where compiled objects will be stored.
CL Source File	The default library where source for CL programs will be stored. This file must reside within the specified source library.
Data File Library	The default data file library specifies the test (or development) library for physical and logical files. This library is used as the default object library for the Software Versions Repository when copying source code for physical or logical files.
SAR Number	An abbreviation for software action request (SAR). <ul style="list-style-type: none"> *NONE = the SAR number will not be validated in any of the CAD/CAP programs and can be left blank. If a SAR number is entered, it is used in conjunction with the SAR Delivery Type of *DFT (default). If the SAR Number is left blank, you must enter a valid SAR number when using the CAD/CAP tools.
Version ID	The software version number to be defaulted in the Software Versions Repository file.

Field	Explanation
Status Code	This code determines the status of the software as well as where it resides in production. It will specify that the software is in production, in development, or in release.
Compile Job Queue	Specifies which job queue will be used when submitting programs to compile. This job queue is used for programs with function code of RPG, CBL, PLI, C and SYSC.
Prog Gen Job Queue	Specifies which job queue will be used when submitting jobs to the Program Generator. These jobs include the source code generation and the source code monitor from SEU.
Compile Target Release	Used by various System CRT commands (RPG, CLP, COBOL, C) to compile an object compatible with a specified target release. <ul style="list-style-type: none"> ■ A value of *CURRENT compiles an object compatible to the release of the machine at compile time. ■ A value of *PRV compiles an object compatible with both one release back and the current release.
Print Option	Used to designate whether or not a report will be generated when an object is compiled. <ul style="list-style-type: none"> ■ 0 = no print ■ 1 = print ■ 2 = print and hold spool file ■ 3 = print only, does not generate execution object (applies to COBOL and RPG only) ■ 4 = print when compile or creation fails
Cross-Reference Listing	Enter Y or N. Specifies whether a cross-reference listing will be generated for variables and fields in a program's compile listing.
SAR File Library	Specifies which library the Software Action Request (SAR) file (F4801) being used for software development exists in. If left blank, the user's library list will be used. You may specify *NONE in the SAR number field (MSAR) if you do not want any SAR number editing.
SAR Delivery Type	Associated with SAR logging, which tracks all modifications to JD Edwards World' software. For example, it will track when User Defined Codes are modified. <ul style="list-style-type: none"> ■ *NONE = no logging. ■ *LOG = log to SAR number 00000000 (no SAR number is used for logging). ■ *DFT = log to a default SAR number (specified in the SAR Number field). ■ *PROMPT = log and prompt the user for the SAR number to be used and allow the user to enter the revision notes.
Source Gen Opt (Future)	Specifies whether to generate source interactively or in batch for programs with this option (for example, Fast Path Application Tool). Allowed values are as follows: <ol style="list-style-type: none"> 1. generate source on-line (interactively) 2. generate source in batch
Helps Maint Opt(Future)	Enter a user defined code, 92/HL.

4.5.1 Function Exits

Choose Redisplay Previously Changed Member (F9) to locate the last record to which you made changes.

4.6 Object Authorities

The system checks the user's authorities to some objects at different steps in generating programs using CASE. Therefore, it is necessary that you review these authorities initially.

4.6.1 Job Control Authority

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

4.6.2 Source Library

Ensure that the user has Object Management authority to the Source Library for software development.

4.6.3 Source File

Ensure that the user has Object Management authority to the Source File for software development.

4.6.4 Job Queues

Ensure that the user has authorization to use the job queues for generating source code and compiling programs.

Part II

Program Generator

This part contains these chapters:

- [Chapter 5, "Overview to Program Generator,"](#)
- [Chapter 6, "Access Program Generator Specifications,"](#)
- [Chapter 7, "Define Program Purpose and Type,"](#)
- [Chapter 8, "Work with File Specifications,"](#)
- [Chapter 9, "Define General Instructions,"](#)
- [Chapter 10, "Define Option and Function Exits,"](#)
- [Chapter 11, "Work with the Detailed Programming Facility,"](#)
- [Chapter 12, "Define Processing Options."](#)

Overview to Program Generator

This chapter contains the topic:

- [Section 5.1, "About Program Generator Steps."](#)

5.1 About Program Generator Steps

You perform the following tasks to create a program using the Program Generator:

- 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

Access Program Generator Specifications

This chapter contains these topics:

- [Section 6.1, "Accessing Program Generator Specifications,"](#)
- [Section 6.2, "Function Exits."](#)

6.1 Accessing Program Generator Specifications

You use the specification screens in the Program Generator to create a program.

- You must enter two specifications:
 - Program Purpose and Type
 - File Specifications

The system allows a third specification, Detailed Programming Facility, which it creates after you enter the File Specifications.

- Optional specifications include:
 - General Instructions
 - Option and Function Exits
 - Processing Options
 - Automatic Accounting Instructions

The system only requires source for files and common copy modules during the specifications and generation steps. The system does not require objects you define externally until you compile the program.

This section contains the following tasks:

- To access the Program Generator
- To access Program Generator Options

To access the Program Generator

The Software Versions Repository screen serves as the portal screen to all JD Edwards World Design tools including the Program Generator.

Navigation

From Computer Assisted Design (G92), choose Software Versions Repository

1. Locate a member from the Software Versions Repository. For example, locate P92801.

Figure 6–1 Software Versions Repository (Program Generator) screen

9801 Software Versions Repository

Field Sensitive Help
Display Error Message
Display Functions
JDE Command Line
Exit Program
Repository Services
Optional File Information
Redisplay Previously C
Checklists
Member Category Code
Member Parameter/Key
Where Used (Cross Ref)
Position Cursor to Action
Maintain Replacement
Previous Member
Next Member
Clear Screen
Flowchart Pgms/ Illustrat

Action Code: 1
Member ID: P92801
Description: Item Maintenance
Function Code: RPG RPG Programs
Function Use: 198 Model Source Member
Product Code: 92 Computer Assisted Design
Reporting System: 92 Computer Assisted Design
Base Member Name: P92801
Maint/RSTDSP: 1
Copy Data (Y/N): N
File Prefix:
Generation Seq:
Common File: N

DREAM Writer Form Exists

P	Source	Object	Source	SAR	Version	S	D	User	Date
	Library	Library	File	Number	ID	C	P	ID	Modified
10	JDWSRC81	JDW08J81	JDESRC	5	A81	1		JDE	07/14/05

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

2. Enter 10 (Design) next to the environment in the following field:
- Option
- The Program Generator Specification screen displays.

Figure 6–2 Define Generator Specification screen

93100M Define Generator Specification

Field Sensitive Help
Display Error Message
Display Functions
Parameter Validation M
Exit Program
Repository Functions
Software Search
Automatic Accounting In
Select All Functions
Delete All Specifications

Member ID: P92801
File ID: JDESRC
SAR Number: 99
Src Library: JDFSRC91

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

Opt Generator Definition Option

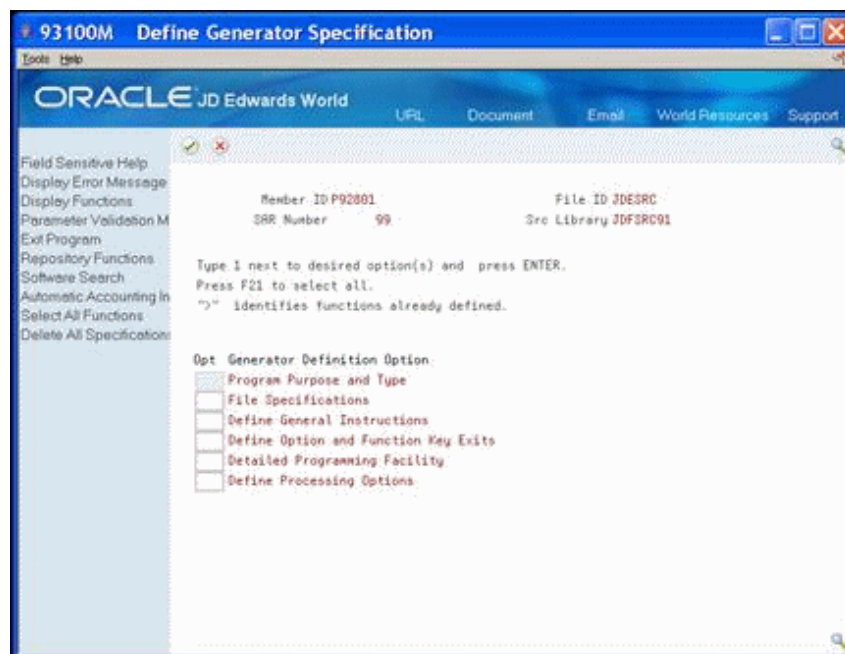
- ☐ Program Purpose and Type
- ☐ File Specifications
- ☐ Define General Instructions
- ☐ Define Option and Function Key Exits
- ☐ Detailed Programming Facility
- ☐ Define Processing Options

To access Program Generator Options

Enter 1 in the following field for the appropriate option on the Define Generator Specifications screen.

- Option

Figure 6–3 Define Generator Specification (Options) screen



Specification	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 files.
Define Processing Options	Allows the user to define processing options the program can use.

6.2 Function Exits

Parameter Validation Monitor (F2)

Choose Parameter Validation Monitor after you enter all of the Program Generator specifications to determine if the monitor program can detect any pre-defined errors.

- This program verifies important features that are pertinent to generating source code by the Program Generator
- This program does not verify whether you regenerate the file specifications after you change your video file

The monitor program verifies that you:

- Specify \$\$ fields in the TOTAL formats of the report file for the a generated report program if the report includes a total column
- Define the SH#RRN field for interactive subfile programs processing by relative record number
- Define the Fie Information Data Structure for interactive programs processing by relative record number
- Choose a keyed master file for programs processing by relative record number
- Define a field as mandatory entry N for transaction processor programs (subfiles)
- Define a hidden field for interactive transaction processor programs
- Define the master file key fields as output
- Attach a validation file to fields that are set up to use next numbers

Repository Functions (F6)

Choose Repository Functions (F6) to access a window of JD Edwards World repositories. This is the same window you access from the Software Versions Repository and CASE Profiles screens.

Software Search (F9)

Choose Software Search (F9) to access the Software Search facility. On the Software Search facility, enter a program name (generic*) to view all program names that meet or are greater than the search criteria.

Automatic Accounting Instructions (F13)

Choose Automatic Accounting Instructions (F13) to access the Automatic Accounting Instructions screen. Use this screen for reference only as you cannot generate code from this screen. Data you enter on this screen appears in the AAP portion of the Help instructions you generate.

Select All Functions (F21)

Choose Select All Functions (F21) to access all of the Program Generator definition screens.

Delete All Specifications (F23)

Choose Delete All Specifications (F23) to delete all of the Program Generator specifications for the program. This removes the Pxxxxx and Hxxxxx members from the F93002 file.

Define Program Purpose and Type

This chapter contains these topics:

- [Section 7.1, "Defining Program Purpose and Type,"](#)
- [Section 7.2, "Function Exits."](#)

7.1 Defining Program Purpose and Type

Defining the program purpose and the program type is the first step in creating a program using the CASE 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 system:

- Stores information in the General Purpose/Type Parameters file (F93101)
- Creates the Pxxxxx member in the Additional Help/Modifications Master file (F93002)
- Creates a data item in the Data Item Master file (F9200)

The Program Purpose and Type screen includes:

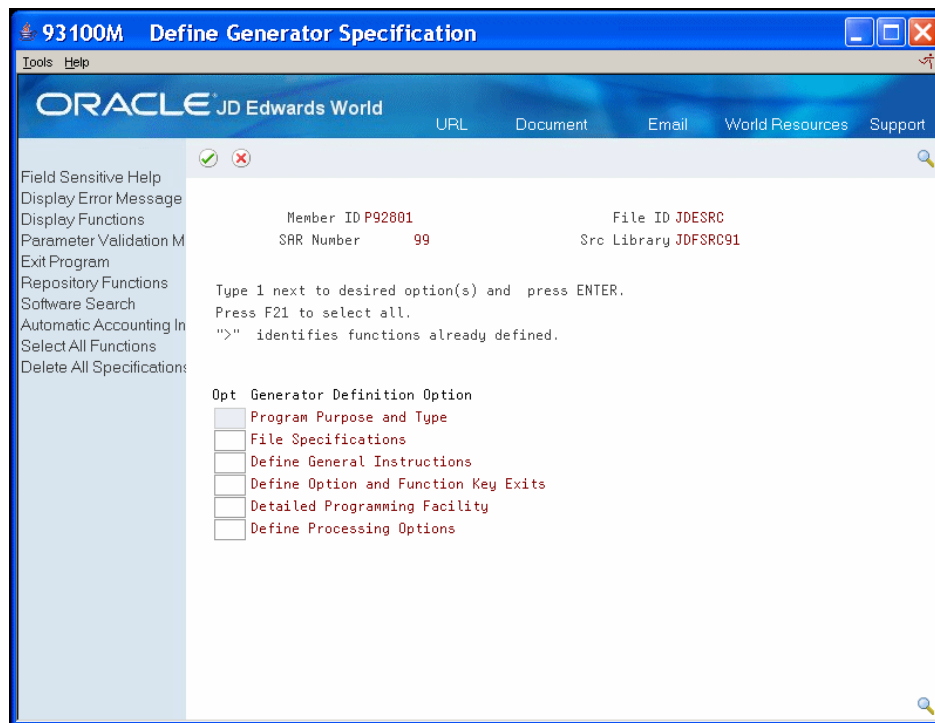
- Software Action Request (SAR) number for the program
- Install system value
- Additional information from the Software Version Repository

The Dialogue Selection screen is a series of questions you use to determine the Program Type.

To define program purpose and type

1. On Define Generator Specification, enter 1 in the following field next to Program Purpose and Type.
 - Option

Figure 7-1 Define Generator Specification (Purpose and Type) screen



2. On Program Purpose and Type, complete the following fields.

- Program ID
- Title
- Purpose
- Product Code
- SAR Number
- CAP Status
- Program Type
- Lockout Action Codes

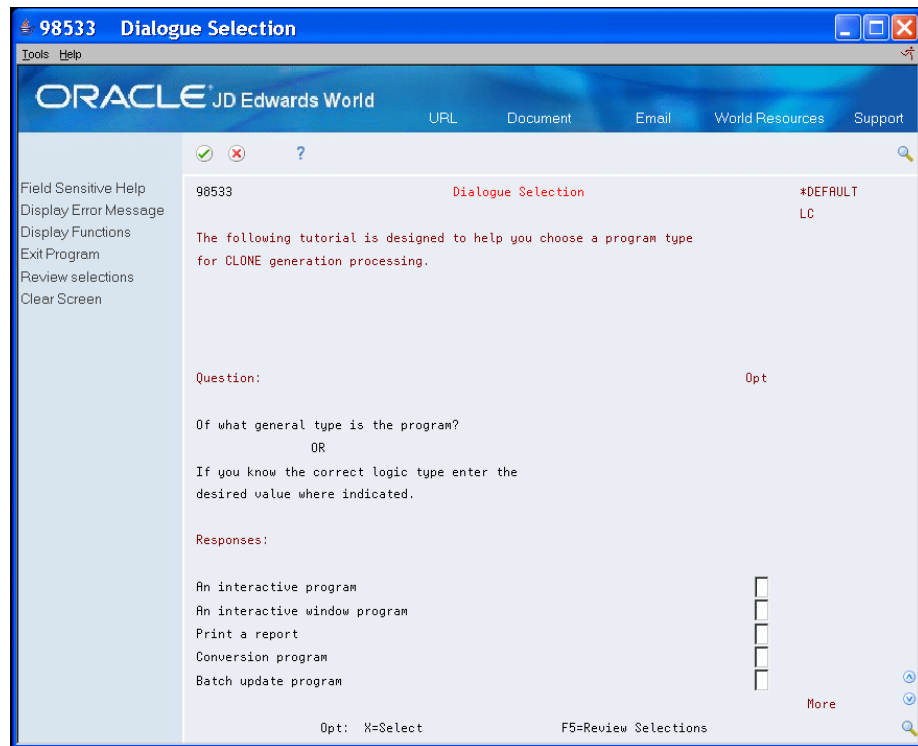
Figure 7-2 Program Purpose and Type screen

Field	Explanation
Program ID	RPG program name specified in the Software Versions Repository. The system adds a data item by this name, with a glossary group of P to the data dictionary as part of this program definition.
Title	<p>The title defaults to the description in the Software Versions Repository and should not be changed.</p> <p>When help instructions are generated, this title appears as the Help program title.</p> <p>Serves as the alpha description for the data item previously mentioned.</p>
Purpose	The full glossary of terms (Data Dictionary) definition. When entering the definition, be aware that the text is wrapped around to fit within 65 characters when being printed for Help Instructions. Therefore, if you are indenting certain parts of the definition, keep the entire line to within 65 characters to retain your desired indentation.
Product Code	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>This field indicates whether the source code for a program can be generated using the program generator. The default for this field is "Y" (Yes, can be generated). The generation program will only generate source code for programs having a code of "Y". This should be set to "N" (No, cannot be generated) if the Program Generator should not be used to generate the program source or if the source generation process is complete and the program has moved into production.</p> <p><i>Screen-specific information</i></p> <p>This field also indicates whether the source code for a program can be modified using the program generator.</p> <p>The five additional serial number fields are still included in the source file (142 characters).</p> <p>When the source generation process is complete and the program has moved into a production source file (92 characters)</p>
Program Type	<p>The Program Type is a name used to identify the basic functions of a program. Each program type is made up of several logic modules. Each logic module contains small sections of RPG code. The program type determines which particular logic modules go together to create the desired program.</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 Action Code 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

1. On Program Purpose and Type, choose Program Type Determination (F11).
2. On Dialogue Selection, answer the questions.

Figure 7-3 Dialogue Selection screen

1. The following graphics illustrate the flow you use in selecting the proper program type.

Figure 7-4 *Selecting the Proper Program Type*

**What is the general type
of program?**

<i>Interactive</i>	A
<i>Interactive form</i>	E0010
<i>Print a report</i>	B
<i>Conversion program</i>	C
<i>Batch update program</i>	D

Figure 7-5 A: Interactive Program flow

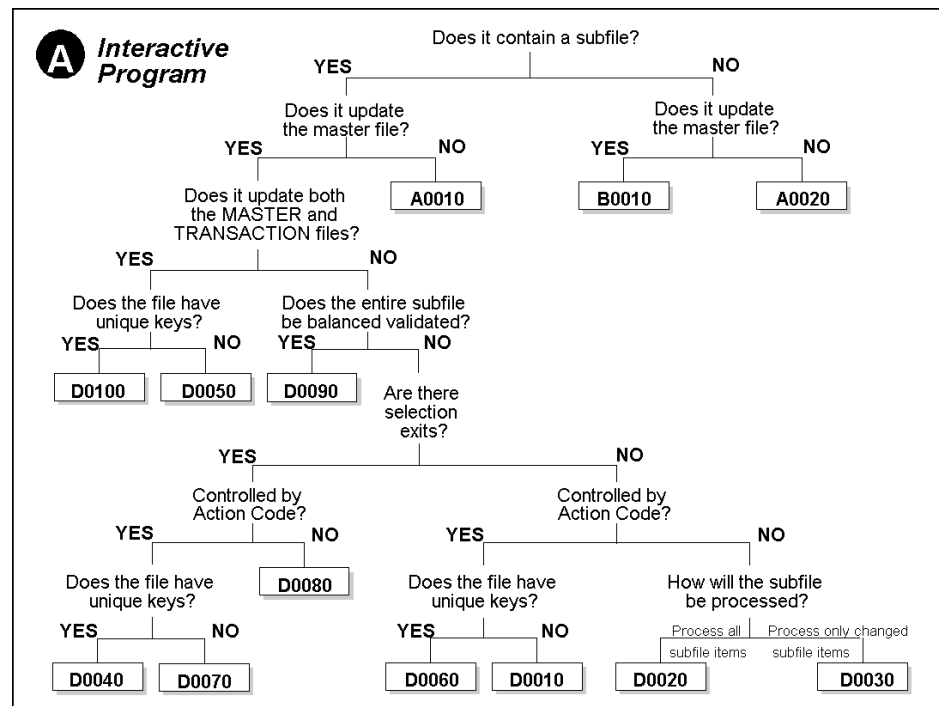
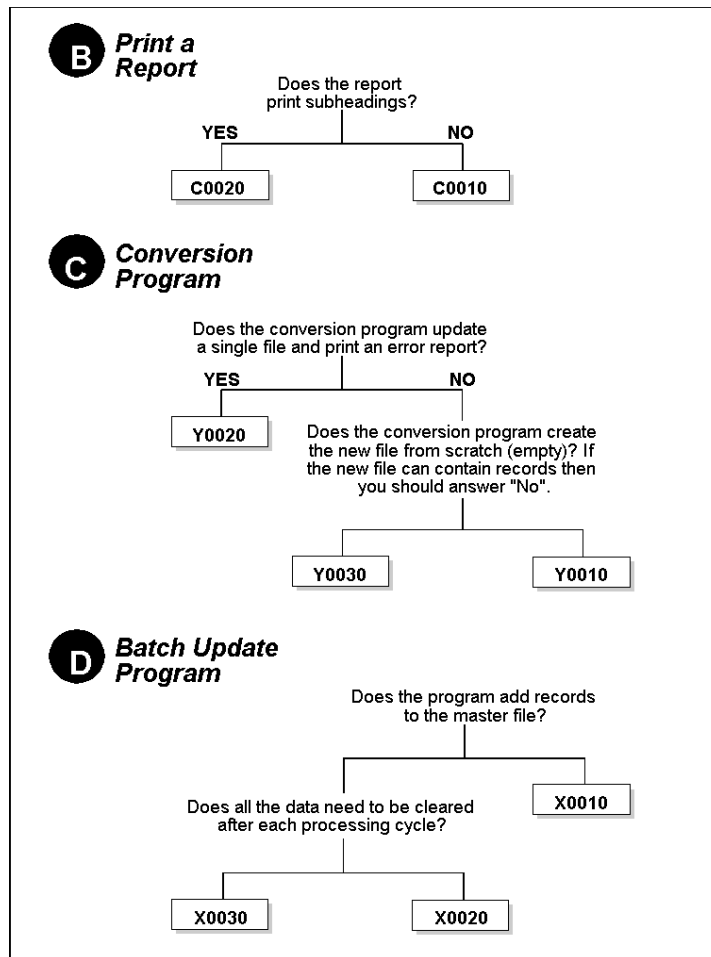


Figure 7-6 Three Program flow



7.2 Function Exits

Program Type Cross-Reference (F2)

Choose Program Type Cross-Reference (F2) to access Program Type Cross-Reference which allows you to view all the programs with the same program type.

Program Type Determination (F11)

Choose Program Type Determination (F11) to access the first dialogue screen if there is no program type.

When you copy a program with specifications, it is not necessary to complete the question and answer process, which the system uses to determine the program or logic type.

Work with File Specifications

This chapter contains these topics:

- [Section 8.1, "What Are File Specifications?"](#)
- [Section 8.2, "Function Exits."](#)

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. The system stores information in F93102 and F93103 and creates the F93105 records.

A significant feature of the Program Generator is its interpretive ability to include secondary editing and referencing files.

This chapter includes the following tasks:

- To enter file specifications
- To generate source code from file specifications

8.1 What Are File Specifications?

A key step in generating source code is that you correctly specify the master files for a program. The database Input/Output operations depend on the files you specify.

PROGRAM TYPE	DESCRIPTION	SPECIFICATION
A0010	SFL (IBM Subfile) Inquiry	Specify the master file with an M or 1 in the Input field.
A0020	Single Record Inquiry	
C0010	Standard Report	
C0020	Standard Report -	
C0025	Subheading	
E0010	Standard Report - Subheading above Columns	
	Window	

PROGRAM TYPE	DESCRIPTION	SPECIFICATION
B0010	Single Record	Specify the master file with an M or 1 in the Update field
D0040	Maintenance	
D0045	SFL Maintenance - KEY	
D0060	SFL Maintenance - KEY, No Action Code	
X0010	SFL Maintenance - KEY	
Y0020	Batch Update - 1 File	
Y0030	File Conversion - 1 File File Conversion - 1 File	
D0100	SFL Maintenance - KEY, 2 Update Files	Specify the master file, which the system maintains in the SFL Control format, with a 1 in the Update field. Specify the transaction file, which the system maintains in the SFL format, with a 2 in the Update field.
D0010	SFL Maintenance - RRN	Specify the logical file which the system uses to fill the subfile, with a 1 in the Input field. Also, enter a File Information Data Structure name for the logical file in the fold area.
D0020	SFL Maintenance - RRN	
D0030	No Action code	
D0070	SFL Maintenance - RRN	Specify the physical file that the system updates with a 2 in the Update field. Also, enter N in the Key field for the physical file in the fold area.
D0070	No Action code	
D0080	SFL Maintenance - RRN	
D0090	SFL Maintenance - RRN	
	SFL Maintenance - RRN No Action code SFL Maintenance - RRN	
D0050	SFL Maintenance - RRN, 2 Update Files	Specify the master file, which the system maintains in the SFL Control format with a 1 in the Update field. Specify the logical file that the system uses to fill the subfile with a 3 in the Input field. Also, enter a File Information Data Structure name for the logical file in the fold area. Specify the physical file that the system updates with a 2 in the Update field. Also, enter N in the Key field for the physical file in the fold area.
X0020	Batch Update, 2 Files	Specify the input file with a 1 in the Input field. Specify the output file with a 2 in the Update field.
X0030	Batch Update, 2 Files	
Y0010	File Conversion, 2 Files	

The Program Generator requires that you:

- Specify one master file with an M or a 1. Do not specify one file with an M and another file with a 1.
- Enter the correct function code on the Software Versions Repository for the screen or report. Otherwise, the Program Generator does not generate moves to the screen or report.

You can use non-JD Edwards World files with the Program Generator, but you must enter the file in the Software Versions Repository.

The Program Generator does not require that you enter values in the Input, Output, or Update fields for a screen or report.

After you complete the appropriate fields on the File Specifications screen, a job runs interactively that analyzes the file specifications and creates records for three Program Generator files. As the system processes this job, messages display at the bottom of the screen.

File	Description
File Specifications F93102	<p>The system updates this file with one record for each file in the File Specification.</p> <ul style="list-style-type: none"> ■ If the master file includes a Business Unit field, then the system adds the Business Unit Security file (F0001) to the File Specifications. ■ If the master file is for an interactive program and contains a field that uses a validation file, then the system adds that file to the File Specifications. If you later decide that the file validation is not necessary, you can delete it on the File Specifications screen.
Data Base Format Parameters F93103	<p>The system updates this file with one record for each format in each file.</p> <p>If the file is a database file, then the F93103 record contains the name of the Key List that the Program Generator uses, and the names of the key fields.</p>
Detail Program Logic Parameters F93105	<p>The system updates this file with one record for each field in each file. The system uses the records in the Detailed Programming Facility.</p> <ul style="list-style-type: none"> ■ If the file is a master file or device file, then the system includes all fields. ■ If the file is a database file that you use only for input purposes, then the system includes only the key fields.

Note: If you make changes to the fields in any of the files you list in the File Specifications, you must run the File Specifications again. If you delete a field from a file, you must manually delete that field from the Detailed Programming Facility. Running the File Specifications again does not remove records from the F93105 file.

To enter file specifications

1. On Define Generator Specification, enter 1 in the following field next to File Specifications
 - Option

Figure 8–1 Define Generator Specification (Enter File) screen

93100M Define Generator Specification

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
Parameter Validation Mon
Exit Program
Repository Functions
Software Search
Automatic Accounting Instr
Select All Functions
Delete All Specifications

Member ID P92801 File ID JDESRC
SAR Number 99 Src Library JDFSRC91

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

Opt Generator Definition Option

<input type="checkbox"/>	Program Purpose and Type
<input checked="" type="checkbox"/>	File Specifications
<input type="checkbox"/>	Define General Instructions
<input type="checkbox"/>	Define Option and Function Key Exits
<input type="checkbox"/>	Detailed Programming Facility
<input type="checkbox"/>	Define Processing Options

2. On File Specifications, complete the following field:
 - File
3. Complete the appropriate field:
 - Input
 - Output
 - Update
 - Add

Figure 8–2 File Specifications screen

File	Input	Output	Update	Add	CC Sec	
F0001	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Business Unit Security
F92001	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	SDM Item Master File
F92001LR	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	LF - Cost Center, Item ID
V92001	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Item Maintenance
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

If you specify Update for a file, the Program Generator examines all fields in that file and includes any other files necessary to edit those fields during an update.

- 4. Choose Extended Parameters (F4) to complete the fields in the fold area.

Figure 8–3 File Specifications (Fold) screen

Field	Explanation
File	The member ID of the file used by the program.
Input	<p>A code used to specify that a data file will be used as input only by the program being generated. The value entered in this field designates secondary meanings for the use of the file: "M" or "1" thru "9" = Master input file No. 1 thru No. 9,</p> <p>P – Primary input file. The "P" will generate the F specification as input primary</p> <p>S – Secondary input file. The "S" designates input secondary.</p> <p>X – Input file. Any master file designation or an "X" will generate the RPG file (F) specification as input full procedural.</p> <p>When working within a Report program type: Files are not treated as input/primary. The program forces a read of the file for control level processing. You can only define one master file read within a report.</p>
Output	A code of X indicates this data file is used as output only by the program to be generated. The RPG file (F) specification will be generated with a file type of O in position 15.

Field	Explanation
Update	<p>A code in this field designates that a file is to be updated within the program being generated. The value entered in this field designates secondary meanings for the use of the file:</p> <p>M or 1 thru 9 - Update master file</p> <p>P - Update primary file</p> <p>S or X - Update secondary file</p> <p>T - Update transaction file</p> <p>When defining a subfile transaction processor program type that updates the master file by relative record number, you must designate the keyed file as master file No. 1 and the file updated by relative record number as master file No. 2.</p>
Add	<p>A code of X specifies that a file will have records written to it in the program being generated.</p> <p>The data file designated as the master file in all file maintenance programs must be designated as allowing file additions.</p> <p>A code of X will generate an A in column 66 of the file (F) specification in RPG.</p>
Src Lib/File	<p>The Library Name field contains the name of a valid AS/400 library name. Defaults from SVR.</p> <p>In the Program Generator Data File parameters this library name is the library where the data file's source file resides. For logical files it is necessary that the based on physical file's source exist in the same source file.</p>
Keyed(Y/N)	<p>A code of Y indicates the data file being specified is keyed. A value of N indicates the file access will be by relative record number. The default value is Y.</p> <ul style="list-style-type: none"> ■ If processing by RRN, the physical file that is being updated must be specified as keyed = N.
File Info DS	<p>Name assigned to an RPG III file information data structure if needed for an associated data file.</p> <ul style="list-style-type: none"> ■ 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. ■ Suggested naming conventions are INFDS1, INFDS2, and so forth. ■ Used with a keyed data file that does not have UNIQUE keys. <p>If you use the POST operation code from IBM for a file information data structure, you must do it for every file information data structure in the program.</p>
PF Src Lib/File	<p>Library where the source resides for the physical file linked to the logical file.</p>

Field	Explanation
External(Y/N)	<p>The External File field designates to the program generator whether or not you wish the data file being specified to be an externally defined file or an internally defined file. The default is "Yes" if left blank.</p> <p>Absolutely no data files specified in normal application software created by JD Edwards World and Company may be internally defined. This parameter is strictly reserved for files designated in multi-file access utility functions that deal with source files or other types of system software.</p>

8.2 Function Exits

Extended Parameters (F4)

Choose Extended Parameters (F4) to display the fold area with the library names of the source files. Default library names are in the Software Versions Repository and your library list.

Data Model (F5)

Choose Data Model (F5) to access the Work with File Relationships screen to build the data model. You must build or rebuild the Cross Reference Index before you can view the data model by choosing Cross-Ref Index from the Rebuilds and Global Updates menu (G9642).

Search (F9)

Choose Search (F9) to access the Software Search facility. On the Software Search facility, you enter a program name to view all programs that meet or are greater than the search criteria.

Notice that the system updates the fields on this screen from your File Specifications.

To generate source code from file specifications

When you generate the source, the system submits a batch job to process your program specifications. The system submits the job to the generation job queue in your CASE Profile. This naming convention for this job is your member ID with a prefix of G.

1. On Software Versions Repository, locate a member.

Figure 8–4 Software Versions Repository (Source Code) screen

9801 Software Versions Repository

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
JDE Command Line
Exit Program
Repository Services
Optional File Information
Redisplay Previously Cha
Checklists
Member Category Codes
Member Parameter/Key Li
Where Used (Cross Refer
Entry Parameter Definition
Position Cursor to Action C
Maintain Replacement Prc
Previous Member
Next Member
Clear Screen
Flowchart Pgms/ Illustrate

9801 Software Versions Repository

Action Code: I
Member ID: P92801
Description: Item Maintenance
Function Code: RPG
Function Use: 198
Product Code: 92
Reporting System: 92
Base Member Name: P92801
File Prefix:
Maint/RSTDSP: 1
Omit Option:
Copy Data (Y/N): N
Optional File: N
Common File: N

DREAM Writer Form Exists

Source	Object	Source	SAR	Version	S	D	User	Date
P	Library	Library	File	Number	ID	C	ID	Modified
JDFSRC91	JDFOB791	JDESRC	99	A91	1		AC2901443	26.10.07
T920835HGB	T920835HGB	JDESRC	99	A91	2		HB920835	03.08.06

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

2. Enter 15 in the following field to generate the source and help.
 - Option
3. Enter 14 in the following field to compile the program.
 - Option
4. Review the program compile and correct any errors.
5. Repeat the steps to generate and compile if necessary.

Define General Instructions

This chapter contains these topics:

- [Section 9.1, "About Special Characters,"](#)
- [Section 9.2, "Special Characters within Help Instructions."](#)

You use General Instructions to create or change program-specific help text for the program that you are creating. To work with Define General Instructions you should be familiar with:

- Entering and changing text on the Edit screen
- Using special characters
- Updating the help file

The system stores information in the Hxxxxx member of the Additional Help/Modifications Master file (F93002).

9.1 About Special Characters

Following are special characters for general instructions:

Character	Explanation
**	Must be in positions 1 & 2. This causes a page skip when you print the text.
++	Must be in positions 1 & 2 which you follow with a data item. This causes the system to enter the most current data dictionary information.
>>	Enters all help instruction records for the program after the >> character. This character displays only when you print the text.
//BYPASS	Marks the beginning of help information that the system ignores. Enter at the beginning of comment lines.
//END	Marks the end of help information that the system ignores. Enter at the end of comment lines.
	Underlines text.
¢	Underlines and highlights the text.

Character	Explanation
~	Highlights the text Press Shift + Tilde, and then press the Space Bar. Alternatively, press ALT + HEX + A1 if you do not have a Tilde (~) on your keyboard.

9.2 Special Characters within Help Instructions

You can use the special characters to display fields with specific attributes. You begin and end the text selection as follows.

- To underline General Use, enter | General Use |
- To highlight Additional Features, enter ~Additional Features~
- To underlines and highlight Special Considerations, enter ¢Special Considerations¢

After you define the General Instructions, you must rebuild the help instructions in order to include them in the interactive Help Instructions Master file (F98HELP).

Figure 9–1 Help Instructions Modifications

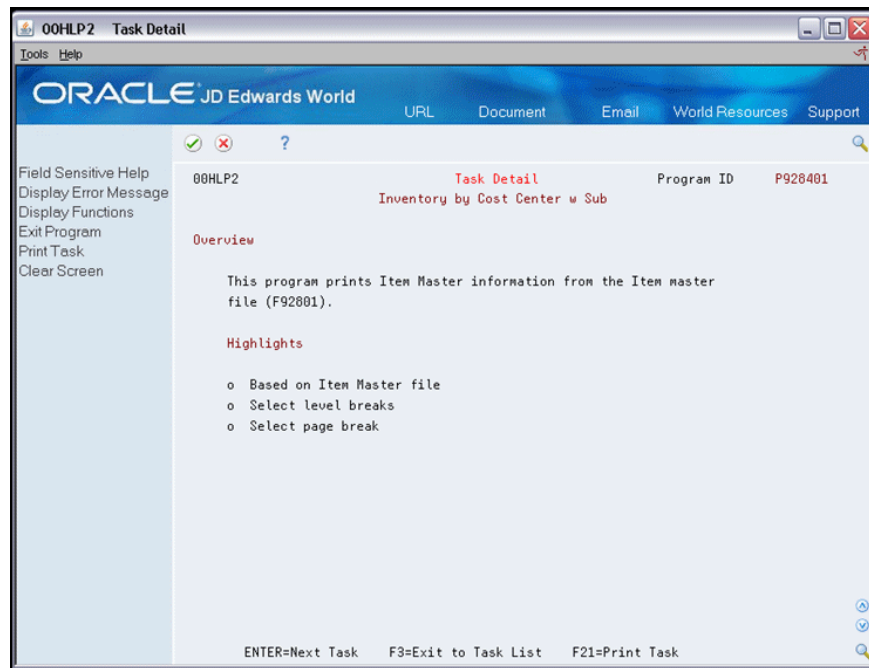


When you choose Help Instruction from the Computer Assisted Design (CAD) menu (G92) to review the F00HELP1 file, notice that the system adds some directional statements to the general instructions you create. The format is as follows:

++ 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 automatically creates field explanations and a list of functions and selections for the program. HELP100, HELP200, and so on, are entries in the Data Dictionary.

The results might display as follows:

Figure 9–2 Task Detail screen

Select General Instructions to view the program-specific help text for the program.

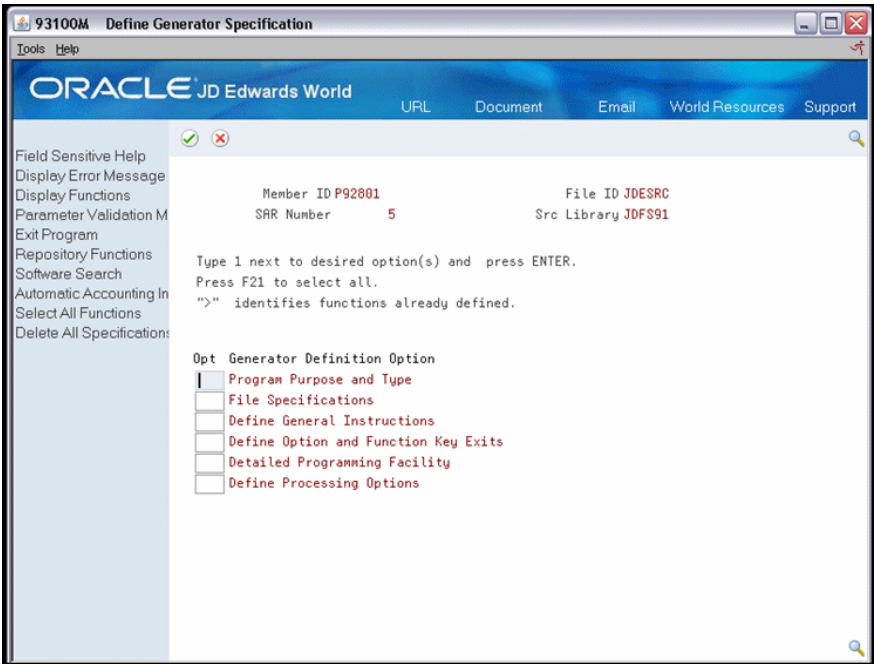
This chapter includes the following tasks:

- To define general instructions
- To update the help instructions

To define general instructions

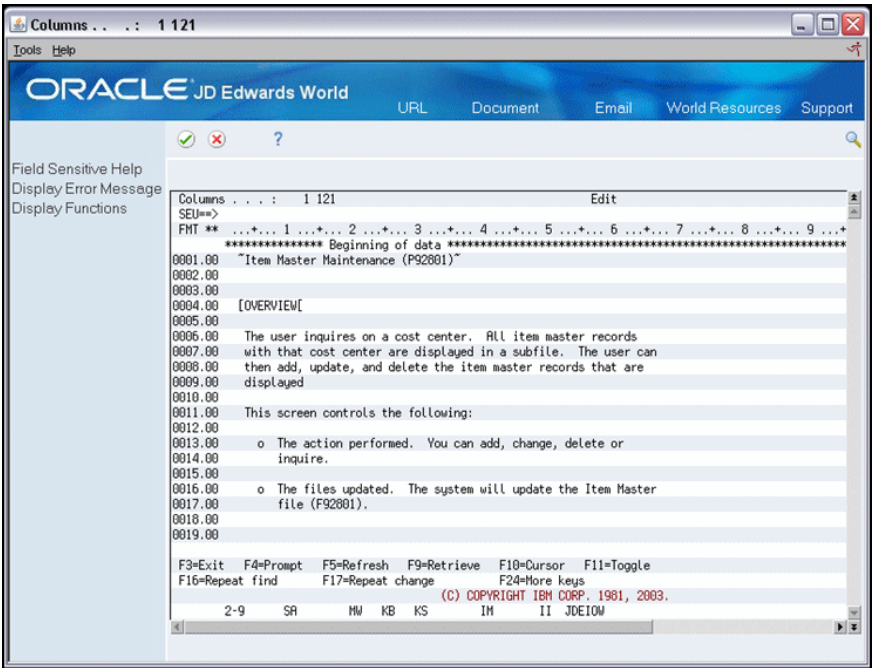
1. On Define Generator Specification, enter 1 in the following field next to Define General Instructions:
 - Option

Figure 9-3 Define Generator Specification (General Instructions) screen



- 2. On the Edit screen, enter the program-specific help text.
You should keep the text between columns 5 and 70 or the text will be truncated.

Figure 9-4 Item Master Maintenance screen



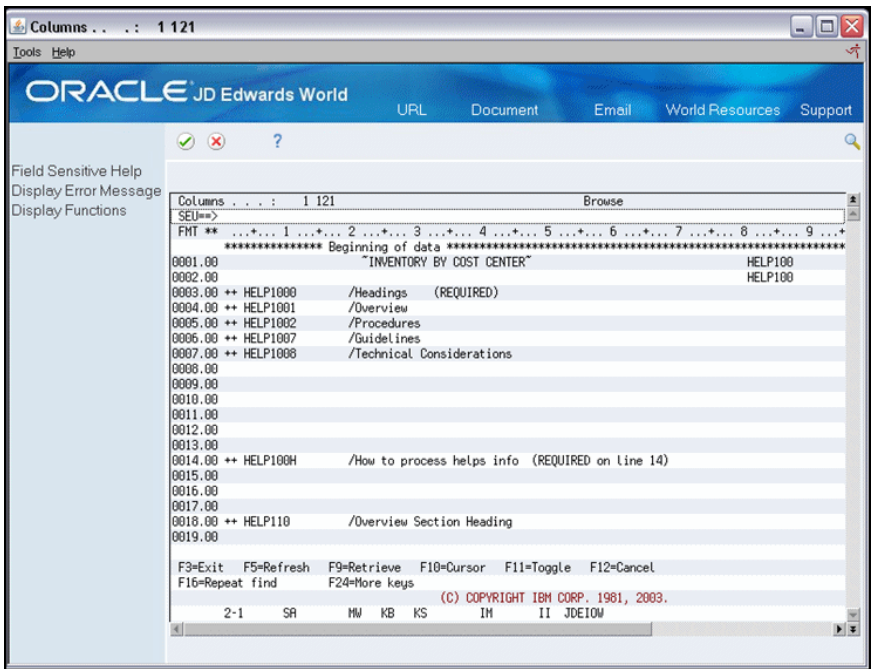
To update the help instructions

There are two methods to update the help instructions.

On Software Versions Repository, enter 15 in the following field to regenerate the program. Alternatively, enter 18 in the following field to rebuild the help instructions:

- Option

Figure 9–5 Columns (Inventory by Cost Center) screen



Define Option and Function Exits

This chapter contains these topics:

- [Section 10.1, "Defining Option and Function Exits,"](#)
- [Section 10.2, "Function Exit."](#)

10.1 Defining Option and Function Exits

To add function exits (function key exits) and subfile selection options to your interactive program, use Define Option and Function Key Exits. The program generator automatically adds the standard function exits to your program, such as Prompt for valid field values (F1), Display Error Message (F7), and Exit (F3). You can highlight the function exits you want to display on line 24 of the program screen using Screen Design Aid (SDA) or Vocabulary Overrides. The program generator creates a list of function exits and selection options for the program. This allows the user to choose Display Options (F24) on any screen to display the list of all function exits and use Prompt for valid field values (F1) in the Option field to display the list of options.

The system stores information for CASE in the Program Exit Parameters file (F93104).

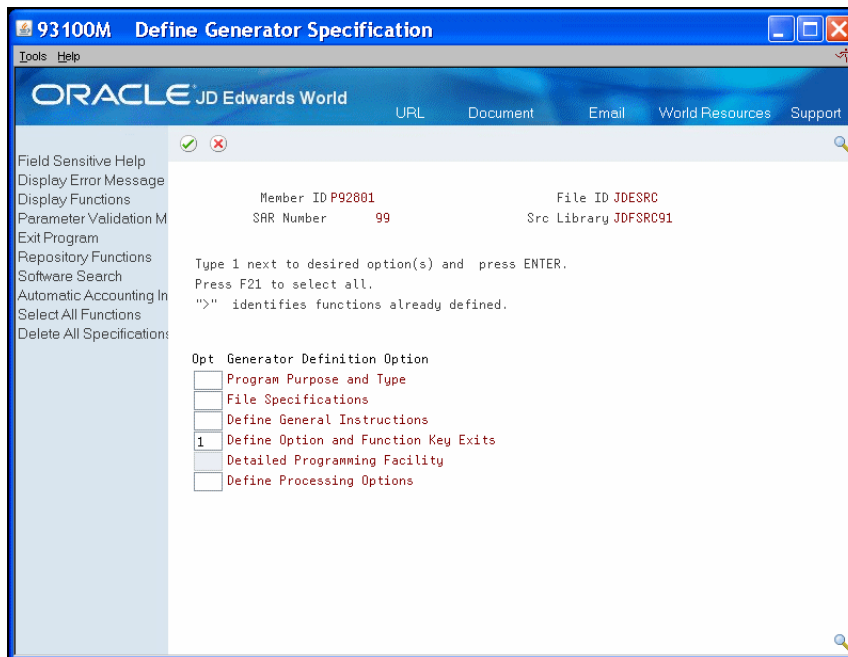
The system creates or updates the Function Key Translation Detail (F9611) and Generic Function Key Master (F96012) tables for the specific screen.

Each set of fields on the Option & Function Key Exits screen pertains to the Function Key or Selection Option you are defining. There is a correlation between the values you enter in the Field field and the Key field. The value in the Field field is the internal data name for the program and the system compares this value to the value in the Key field. The system uses the value in the Program ID field for the name of the program or routine the system executes. The Parm fields are the parameters the system needs for the program or routine.

To define Option and Function Exits

1. On Define Generator Specification, enter 1 in the following field to define option and function exits:
 - Option

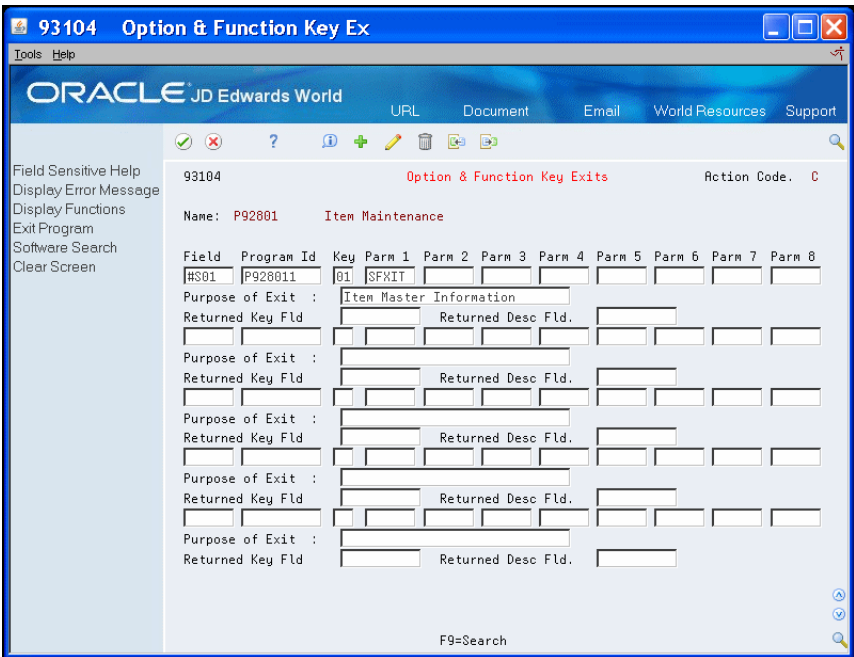
Figure 10–1 Define Generator Specification (Option) screen



2. On Option & Function Key Exits, complete the following fields:

- Field
- Program ID
- Key
- Parm (Parameter) 1 through 8
- Purpose of Exit
- Returned Key Fld
- Returned Desc Fld

Figure 10–2 Option & Function Key Exits screen



Field	Explanation
Field	<p>The RPG field name (6 bytes) to be passed as a parameter on function key exits or subfile options.</p> <p><i>Screen-specific information</i></p> <p>The internal field name the system assigns to each option and function exit in the program you are generating.</p> <p>Correlation exists between this field and the Function exit Definitions repository.</p> <p>Maintained in the soft coding server data structure (I00SC).</p> <ul style="list-style-type: none">■ This is a required field■ Use #S01 - #S16 for options■ Use #F01 - #F15 for function exits
Program Id	<p>The identification, such as program number, table number, and report number, this is assigned to an element of software.</p> <p><i>Screen-specific information</i></p> <p>The name of the program that the system executes when you choose the function exit or enter a selection option value.</p> <p>By prefixing the name with an asterisk (*) you may designate the name of a logic module. A logic module's name that you use for this purpose must begin with an X followed by any eight characters. The name cannot be longer than nine characters in order to allow for entry of the asterisk prefix. This function allows the programmer to create logic other than the standard execution of an external program when a user chooses a function exit or enters a selection option.</p>

Field	Explanation
Key	<p>This field is used to specify the function key or subfile option number for a particular exit in the program being generated. Along with function key or option number, you must enter the program ID and any parameters that should be passed to the external program.</p> <p><i>Screen-specific information</i></p> <p>You can only define function exits for #F01 through #F15 and subfile options for #S01 through #S16.</p>
Parm 1	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 2	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 3	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 4	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 5	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 6	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 7	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Parm 8	The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.
Purpose of Exit	<p>A name or remark that describes an element in the JD Edwards World systems.</p> <p><i>Screen-specific information</i></p> <p>Special Use: For the CASE system, this field is used for program exit remarks and AAI remarks. It also allows you to enter a data dictionary key when prefixed with an * (asterisk). This lets you use standard explanations and provide more extensive explanations for each exit or AAI when viewed with the help instructions.</p>
Returned Key Fld	<p>This field is used to specify the name of the data field to be updated by a returned key when exiting to an inquiry program that allows passing the selected record's key back to the initiating program.</p> <p><i>Screen-specific information</i></p> <p>Causes logic generation to let a returned key pass through the local data area and loads the value in the specified key field. Only valid with the CL program J98LDAKY</p>

Field	Explanation
Returned Desc Fld	<p>This field is used to specify the name of the data field to be updated by a returned description when exiting to an inquiry program that allows passing the selected record's description back to the initiating program.</p> <p><i>Screen-specific information</i></p> <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"> ■ Only valid with the CL program J98LDAKY ■ For more information on using the Returned Key and Returned Desc Fld, see the program level Helps for P93104

10.2 Function Exit

Search (F9)

Choose Search (F9) to access the Software Search facility. On Software Search, enter a program name to view all programs that meet or are greater than the search criteria. You can also enter Generic* to view the names of all program in the SVR.

10.2.1 What You Should Know About

Topic	Description
Values in the Parameter fields	<p>Use caution when using an internal program data name in the Parm fields. Using screen (VD prefix) or subfile (SF prefix) fields might cause issues because the program the system retrieves can change the data in the field.</p> <p>To avoid transferring screen or subfile fields values, alternative options for VDxxxx or SFxxxx include:</p> <ul style="list-style-type: none"> ■ Transfer PSxxxx This requires a manual source change to the program in order to properly load the PSxxxx field with the screen or subfile field, or load the field using Program Design Language. ■ Transfer SHxxx You can define the SHxxxx fields as hidden fields on their screen and then load them with the proper information using the Detailed Programming facility.

Work with the Detailed Programming Facility

This chapter includes these topics:

- [Section 11.1, "About the Detailed Programming Facility,"](#)
- [Section 11.2, "About Full Data Field Parameters,"](#)
- [Section 11.3, "Loading VC0 Description Fields,"](#)
- [Section 11.4, "Enabling the Database Update Function for Subfiles,"](#)
- [Section 11.5, "Creating *ENTRY PLIST Entries,"](#)
- [Section 11.6, "Protecting Fields from Being Cleared,"](#)
- [Section 11.7, "Disabling Data Dictionary Edits,"](#)
- [Section 11.8, "Creating a Partial KLIST for a File."](#)

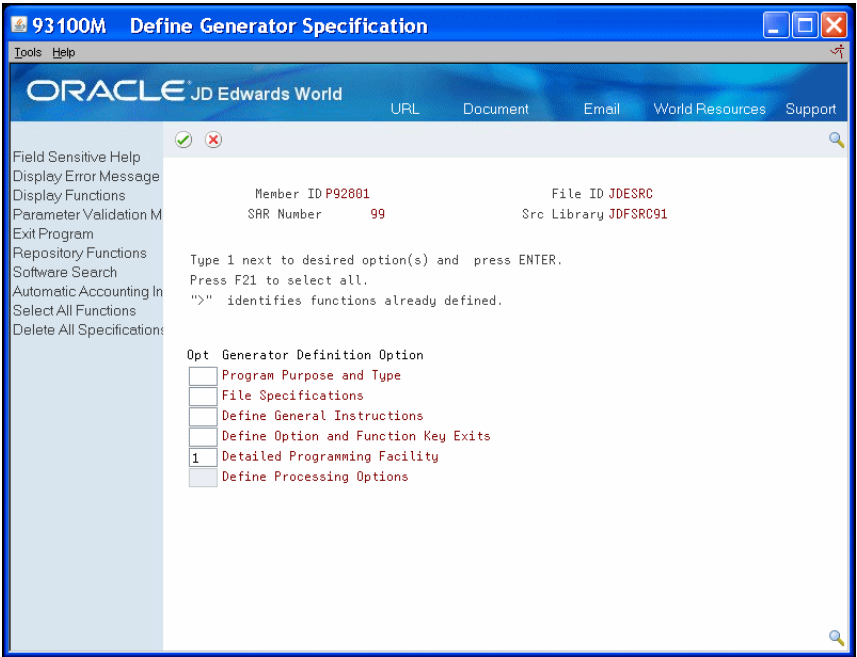
The Detailed Programming Facility allows you to specify data field definition parameters. The Detailed Programming Facility screen lists the files in order and then each field in order within the files for the shell program the Program Generator creates. It also provides access to Field Detail and Program Design Language. The system creates this specification after you enter the File Specifications. Additionally, the system stores the information in the Detail Program Logic Parameters file (F93105).

11.1 About the Detailed Programming Facility

After you enter the file specifications, you access the Detailed Programming Facility from the Define Generator Specification screen.

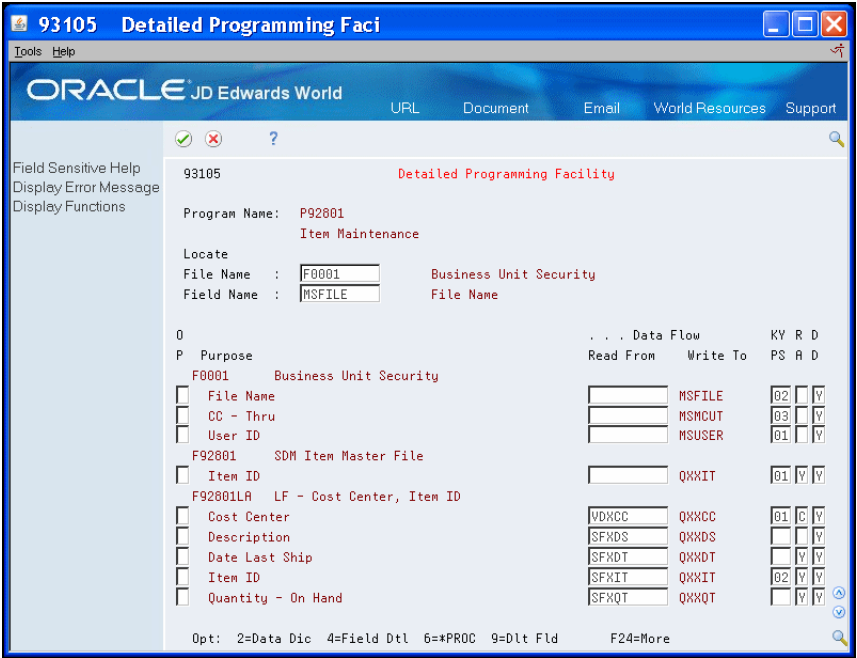
To access the Detailed Programming Facility, you enter 1 in the Option field on the Define Generator Specification screen.

Figure 11–1 Define Generator Specification (Detailed Programming) screen



The Detailed Programming Facility screen displays.

Figure 11–2 Detailed Programming Facility screen



Field	Explanation
File Name	The member ID of the file used by the program.
OP	Allows for selection exits for each field.

Field	Explanation
Data Flow - Read From	Indicates what information is to be loaded into the "Write To" field on the screen. This field is loaded automatically by CAP during the data field generation process initiated by adding files to the file specifications program. It is loaded based upon either a display file or report file data dictionary item name matching with the same data dictionary item name in the specified data base files. If no match occurs for the designated master file fields, this field is loaded with "*SKIP". (An entry of "*" followed by an internal logic module name allows creating standard calculation routines for certain fields.) An entry of *PROC will replace standard code with that generated by PDL.
Key Position	Designates the relative position of the field in the key list. It is used in the program generator to generate key lists (KLIST). You may also define a partial key by blanking out the key position for a particular field. Just remember, partial keys should be defined from the bottom up; for example, don't remove key position 01 if there are 4 keys in the key list.
Right Adjust Parameter	A code of: Y – indicates the field should be right adjusted. N – indicates the field should NOT be right adjusted. C – indicates the field is a business unit and should be left filled with blanks instead of zeros. A – indicates the field is an account number and the account number edit routine will be used for editing. Can only be used when the Read From field is a video field and the Write To field is a data base field.
Data Dictionary Validation	Designates whether the Program Generator will generate all the editing logic specified in the Data Dictionary for the particular data item. Enter a Y if this editing is desired, otherwise enter an N to bypass the Data Dictionary editing. Y is the default.

11.1.1 Available Options

Value	Description
2 - Data Dictionary	Access the Data Dictionary Repository for the data item.
4 - Select/Work With	Access the Full Data Field Parameters screen for more detail on the field.
6 - Data Formula Entry (*PROC)	Access the Data Item Formula Revisions screen where you enter Program Design Language (PDL) code.
9 - Delete Record	Allows you to delete a field from the Detailed Programming Facility.

11.1.2 Function Exits

Repository Services (F6)

Choose Repository Services (F6) to access a screen of JD Edwards World technical functions or repositories.

Select *PROC Fields On/Off (F10)

Choose Select *PROC Fields On/Off (F10) to toggle between a display of either all fields in the Detailed Programming Facility or the fields with PDL.

11.2 About Full Data Field Parameters

Full Data Field Parameters allows you to create additional source code.

11.2.1 Primary Uses of Full Data Field Parameters

The primary uses of full data field parameters include:

- Loading VC0 description fields: The system can store VC0 (descriptive data) about either a screen (video) or report in another file. Use this screen to enter the file in which you want the system to store the data about the field, the file key, where you want the system to store the description and the field with which you want to associate the field.
- Enabling the database update function for subfiles: Entering N in the Entry Optional field enables the subfile field that controls data base updates
- Creating the *ENTRY PLIST code for a program: Specifies which data fields you want to include in an *ENTRY PLIST statement and the sequence in which they will appear.
- Protecting a field from the system clearing it every time the system executes the S001 routine when you generate the code.
 - You specify N in the Clear After (Y/N) field
 - The system requires this for output only fields that do not have a VC0 prefix
 - The system requires this for key fields in RRN program types
- Adding error message the user creates
- Suppressing edits in S005 for audit fields
- Creating a partial KLIST for an input file

To access the Full Data Field Parameters screen, you enter 4 in the Option field next to the field for which you want to create additional source code on the Detailed Programming Facility screen.

Figure 11–3 Detailed Programming Facility (Full Data) screen

93105 Detailed Programming Facility

Oracle JD Edwards World

Tools Help

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Repository Services
Select *PROC Fields Or

93105 Detailed Programming Facility

Program Name: P92801
Item Maintenance

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

		Data Flow		KY	R	D
P	Purpose	Read From	Write To	PS	R	D
<input type="checkbox"/>			#####D	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>			#####CFL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>			#####CRC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>			#####RNO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Report Column		#####RCOL	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>			#####RROW	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>			#####SFRNO	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Action Code		ACTION	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Selection Exits		SFSELC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Description	QXXDS	SFXDS	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	Date Last Ship	QXXDT	SFXDT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Item ID	QXXIT	SFXIT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Opt: 2=Data Dic 4=Field Dtl 6=*PROC 9=Dlt Fld F24=More

The Full Data Field Parameters screen displays.

Figure 11–4 Full Data Field Parameters screen

93125 Full Data Field Parameters

Oracle JD Edwards World

Tools Help

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Repository Services
File Field Description W
Clear Screen

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: QXXIT Dictionary Name: XIT
Field Type: S Data Field Use: B
Key Position: PLIST Sequence:
Entry Optional: N 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: Error Index:
Error Indicator: 43 Validation File:
Data Item Type: A 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	Indicates what information is to be loaded into the "Write To" field on the screen. This field is loaded automatically by CAP during the data field generation process initiated by adding files to the file specifications program. It is loaded based upon either a display file or report file data dictionary item name matching with the same data dictionary item name in the specified data base files. If no match occurs for the designated master file fields, this field is loaded with "*SKIP". (An entry of "*" followed by an internal logic module name allows creating standard calculation routines for certain fields.) An entry of *PROC will replace standard code with that generated by PDL.
Dictionary Name	The 4 character data item name from the data dictionary. Used extensively for field editing within the program generator.
Field Type	Used to designate master file field names and display/report file field names within the data field parameter records. M – indicates a master file field P – indicates the field is in the control record portion of a video screen (so if not a subfile, all fields would be a P) S – indicates the field is in the subfile portion of a video D – indicates a field within a report detail format H – indicates a field within a report heading format T – indicates a field within a report total format
Data Field Use	To determine how a data item is used on a video screen or report as far as: I – input only O – output only B – both input and output H – hidden field
Key Position	Designates the relative position of the field in the key list. It is used in the program generator to generate key lists (KLIST). You may also define a partial key by blanking out the key position for a particular field. Just remember, partial keys should be defined from the bottom up; for example, don't remove key position 01 if there are 4 keys in the key list.
PLIST Sequence	The PLIST Sequence field specifies to the Program Generator which data fields you wish to include as passed parameters on a *ENTRY PLIST statement and the sequence in which they will appear. <ul style="list-style-type: none">■ 01 - 32 are valid■ Must enter as 01 and not 1■ If the first parameter is passed a non-blank value, an auto-inquiry will be performed
Entry Optional	Used with subfile maintenance programs to identify the field that controls database updates. <ul style="list-style-type: none">■ One field needs to be designated as Entry Optional: N■ Defaults to a blank

Field	Explanation
Clear After (Y/N)	<p>Designates to the Program Generator 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> <p>Y – indicates the field will be cleared at the end of each transaction entry. The default is Y.</p> <p>N – indicates the field will not be cleared unless specified by the user by pressing the appropriate function key.</p>
Right Adj (Y/N)	<p>A code of:</p> <p>Y – indicates the field should be right adjusted.</p> <p>N – indicates the field should NOT be right adjusted.</p> <p>C – indicates the field is a business unit and should be left filled with blanks instead of zeros.</p> <p>A – indicates the field is an account number and the account number edit routine will be used for editing.</p> <p>Can only be used when the Read From field is a video field and the Write To field is a data base field.</p>
Center (Y/N)	A code of Y will center the data within the field when it is displayed.
Description File	<p>Used in conjunction with loading a VC0 description field.</p> <ul style="list-style-type: none"> Identifies the file that contains the description
Descr. File Key	<p>Specifies the key field name to use for retrieving the data description from the designated description file. Enter the field name used to chain to that file. If you need to use a KLIST, enter the KLIST name.</p> <p>If this description is coming from the User Defined Codes file, enter the field that contains the "code" portion of the User Defined Codes key. For example, to retrieve state description using the Address Book file, you would enter ABADDS not DRKY01. The system code and record type will be retrieved from the data dictionary item for state code (ADDS).</p> <p>The program generator will produce the code to chain to the appropriate file and move (left justified) the description to the specified output field (usually VC0xxx).</p>
Dictionary Edit	<p>Controls the generation of data dictionary editing for fields in the master file.</p> <ul style="list-style-type: none"> Defaults to Y Specifying N will result in no data dictionary editing for the value that is moved to a master file field Is useful for audit fields such as User ID that can be loaded from the Program Status Data Structure and need no editing.
Error Msg No	<p>Identifies a custom error message to use when errors are detected on a screen field.</p> <ul style="list-style-type: none"> Loads the value in array EMK of subroutine S999
Validation File	Specifies the file name to use for validating the current data field contents. This file name is automatically provided from the data dictionary if it exists.

Field	Explanation
Error Indicator	Used to designate the error controlling indicator for a data item on a video screen. This indicator controls the standard error notification attributes for video screens (reverse image, high intensity and position cursor).
Error Index	The Error Message Index field is the array index where a special error message number is loaded in the error message array. Each of the data item parameters which uses external file validation can override the standard error message (0002). A new index must be entered for these types of changes. Error indexes 1 through 20 are reserved for the program generator. Error indexes 21 through 30 are reserved for file validation. Error indexes 30 through 64 can be used for anything else.
Data Item Type	This defines the type of data to be stored in the field. The data item types are defined in User Defined Codes, system code '98', record type 'DT'. Note: All amount fields should be entered as 15 bytes, 0 decimals, and data item type should be P (packed).
Data Item Size	The field size of the data item. NOTE: All amount fields should be entered as 15 bytes, 0 decimals, and the data item type should be P (packed).
Decimal Pos	The number of positions to the right of the decimal of the data item.
Edit Code	Determines how data is printed or displayed. Depending on the code, you can change the appearance of the fields as follows (standard IBM edit codes): <ul style="list-style-type: none"> ■ Show commas - 1, 2, A, B, J, K, N, or O ■ Show decimal point - 1, 2, 3, 4, A, B, C, D, J, K, L, M, N, O, P, Q ■ Show sign for negative - A, B, C, D ("CR") or J through Q ("-") ■ Suppress leading zeros - 1 through 4, A through D, J through Q, Y and Z Refer to user defined codes (system 98/ type EC) for all valid codes, including additional JD Edwards World edit codes.

11.2.2 Function Exits

File Field Description Screen (F16)

Choose File Field Description Screen (F16) to access the File Field Description Screen. This function exit is field sensitive.

- If the cursor is in the Description File Key field, and this field:
 - Contains a file name, the system enters the fields on the screen with the fields from the description file.
 - Is blank, the system displays a blank File Field screen for you to enter a file name and then displays the fields from that description file.
- If the cursor is not in the Description File field, the system displays a blank File Field screen for you to enter a file name and then displays the fields from that description file.

11.3 Loading VC0 Description Fields

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

11.3.1 Example

In the following example, QXXCC is a field in the Business Unit Master table (F0006) that contains the business unit value. The F0006 table also contains descriptions of the business units. MCDL01 is the field in the F0006 table that contains the business unit description and the system enters this description into VC0001.

Figure 11–5 Full Data Field Parameters (Example 1) screen

11.3.2 Example: User Defined Code

If you are accessing a description for a user defined code (UDC) field, you enter F0005 (User Defined Codes table) in the Description File and the value for the field for which you are accessing the description in the Description File Key field.

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.

Figure 11–6 Full Data Field Parameters (Example: UDC) screen

In the example above, QXXTY is a UDC field in the User Defined Codes table. F0005 contains descriptions of UDCs. DRDL01 is the field in F0005 that contains the UDC description and the system enters this description into VC0002.

Because a server program accesses the F0005 table, it is not necessary to include it in the File Specifications.

Loading field descriptions using this approach only works if the system enters the field description into a VC0 field.

Specifying a file does not guarantee that the system enters the file you specify into the File Specifications. You must review the File Specifications to ensure the files from which you want to retrieve descriptions are present. The exception is for files that you access with a server program.

To load the VCO Description fields

Complete the following fields:

- Source of Data
- Description File
- Descr. File Key

Figure 11–7 Full Data Field Parameters (VCO) screen

Field	Explanation
Source of Data	Indicates what information is to be loaded into the "Write To" field on the screen. This field is loaded automatically by CAP during the data field generation process initiated by adding files to the file specifications program. It is loaded based upon either a display file or report file data dictionary item name matching with the same data dictionary item name in the specified data base files. If no match occurs for the designated master file fields, this field is loaded with "*SKIP". (An entry of "*" followed by an internal logic module name allows creating standard calculation routines for certain fields.) An entry of *PROC will replace standard code with that generated by PDL.
Description File	Used in conjunction with loading a VC0 description field. <ul style="list-style-type: none"> Identifies the file that contains the description
Descr. File Key	Specifies the key field name to use for retrieving the data description from the designated description file. Enter the field name used to chain to that file. If you need to use a KLIST, enter the KLIST name. <p>If this description is coming from the User Defined Codes file, enter the field that contains the "code" portion of the User Defined Codes key. For example, to retrieve state description using the Address Book file, you would enter ABADDS not DRKY01. The system code and record type will be retrieved from the data dictionary item for state code (ADDS).</p> <p>The program generator will produce the code to chain to the appropriate file and move (left justified) the description to the specified output field (usually VC0xxx).</p>

11.4 Enabling the Database Update Function for Subfiles

If you are designing a subfile maintenance program, you must define at least one field in your subfile as a required field.

In the following example, the Item ID (SFXIT) field is the field that controls database updates:

The system performs the following:

- If Item ID is blank, but there is a database record for the subfile record, then the system deletes the database record.
- If Item ID is not blank, then the system saves or updates the database depending on whether the database record exists 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 a SHxxxx field for each key field that is in the subfile record.
- RRN processing for the subfile. The hidden field must be SH#RRN.

To enable the database update function for subfiles

On the Full Data Field Parameters screen, enter N in the Entry Optional field.

Figure 11–8 Full Data Field Parameters (Update) screen

11.5 Creating *ENTRY PLIST Entries

You use PLIST entries to define which data items to include in a parameter list. You can use a maximum of 32 parameters.

11.5.1 Example

The system uses data item VDXCC as the third parameter in the entry list of Subroutine S999. The program generator creates a field name, which is the same data dictionary item with a prefix of ##. The system moves this parameter field to VDXCC from the parameter field.

You must use the data item in the display file, not the database file, for creating PLIST parameters.

Figure 11–9 Full Data Field Parameters (Example 3) screen

If the parameter value is not blank, the system updates the variable \$AUTO with a 1. When the system launches this program, this parameter informs the program to perform an automatic inquiry (S003).

Figure 11–10 Required Program Parameters screen

If the system retrieves this program directly, the CL program retrieving this RPG program must issue a blank parameter.

To create *ENTRY PLIST entries

On the Full Data Field Parameters screen, enter a two-digit number corresponding to the sequence of the parameter in the PLIST Sequence field.

Figure 11–11 Full Data Field Parameters (PLIST) screen

11.6 Protecting Fields from Being Cleared

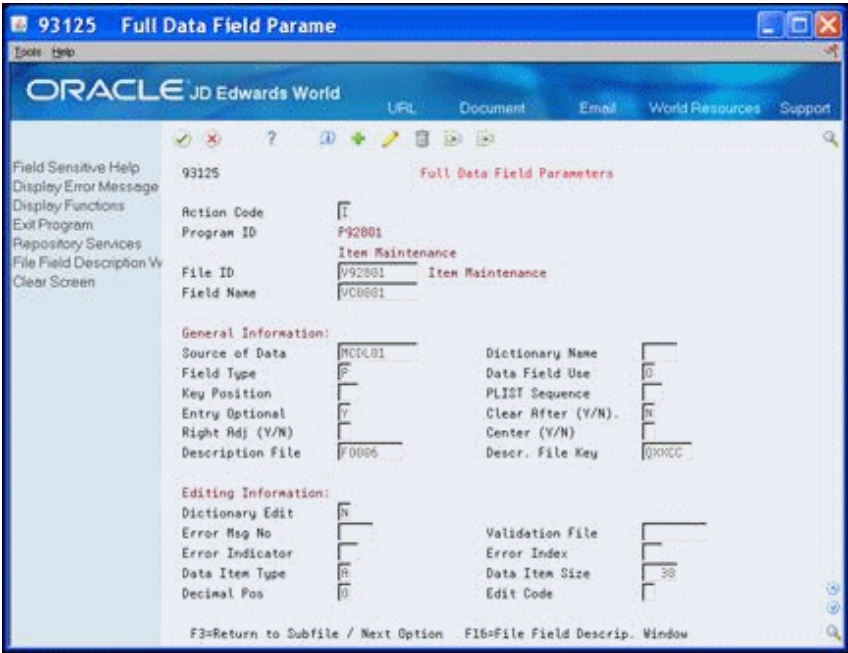
This feature is useful when creating data entry programs with a repetitive data field. For example, when there are multiple occurrences of the date field on a screen, a user only needs to enter a date in the first occurrence of the date field.

- The system clears all fields except those with a prefix of VC0 each cycle in Subroutine S001.
- The default value for this field is Y.
- The function exit F22 clears all fields.

To protect fields from being cleared

On the Full Data Field Parameters screen, enter N in the Clear After field.

Figure 11-12 Full Data Field Parameters (Protect Fields) screen



11.6.1 What You Should Know About

Topic	Description
User Error Messages	<p>Updating the Error Message Number and Error Index fields adds errors to the EMK array in Subroutine S999.</p> <p>For example, in Subroutine S999, the system applies error message 1684 to Error Index 21 of the EMK array. JD Edwards World reserves indexes 1 to 20.</p> <p>You must modify your source to use this error message index.</p>

Figure 11–13 Full Data Fields Parameters (Error Message) screen

93125 Full Data Field Parame

ORACLE JD Edwards World

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Repository Services
File Field Description W
Clear Screen

93125 Full Data Field Parameters

Action Code: 1
Program ID: P92801
Item Maintenance
File ID: V92801 Item Maintenance
Field Name: SFKTY Item Type

General Information:
Source of Data: DIGNTY
Field Type: 5
Key Position: 1
Entry Optional: Y
Right Adj (Y/N):
Description File:

Dictionary Name: DTY
Data Field Use: 8
PLIST Sequence:
Clear After (Y/N): Y
Center (Y/N):
Descr. File Key:

Editing Information:
Dictionary Edit: N
Error Msg No:
Error Indicator: 48
Data Item Type: R
Decimal Pos: 0

Validation File:
Error Index:
Data Item Size: 2
Edit Code:

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

11.7 Disabling Data Dictionary Edits

This feature is useful if you add custom validation through the Source Entry Utility (SEU).

To disable Data Dictionary edits

Enter N in the Dictionary Edit field.

Figure 11–14 Full Data Field Parameters (Disable Data Dictionary) screen

93125 Full Data Field Parame

ORACLE JD Edwards World

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Repository Services
File Field Description W
Clear Screen

93125 Full Data Field Parameters

Action Code: 1
Program ID: P92801
Item Maintenance
File ID: V92801 Item Maintenance
Field Name: SFKDS Description

General Information:
Source of Data: DKKDS
Field Type: 5
Key Position: 1
Entry Optional: Y
Right Adj (Y/N):
Description File:

Dictionary Name: KDS
Data Field Use: 8
PLIST Sequence:
Clear After (Y/N): Y
Center (Y/N):
Descr. File Key:

Editing Information:
Dictionary Edit: N
Error Msg No:
Error Indicator: 44
Data Item Type: R
Decimal Pos: 0

Validation File:
Error Index:
Data Item Size: 39
Edit Code:

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

11.8 Creating a Partial KLIST for a File

The Program Generator displays the full key list. You can change the key position to exclude subordinate elements.

11.8.1 Example

In the example that follows, the key list sequence for the Business Unit security file is:

- User ID
- File Name
- Through Business Unit

If you want to use a partial key to access this file, start with the highest number key field and clear the Key Position (KY PS) field. If you need both the full key list and a partial key list, enter this using the SEU.

To create a partial KLIST for a file

On Detailed Programming Facility, clear the Key Position (KY PS) field, starting with the last element.

In the following example, clear the Key Position (KY PS) field for CC - Thru. The key for F0001 is the User ID and File Name, key positions 01 and 02.

Figure 11–15 Detailed Programming Facility (KLIST) screen

Field	Read From	Write To	KY	R	D
File Name	MSFILE		02		
CC - Thru	MSMCUT		03		
User ID	MSUSER		01		
Item ID	QXKIT		01		
Cost Center	QXKIT		01		
Description	QXKIT				
Date Last Ship	QXKIT				
Item ID	QXKIT		02		
Quantity - On Hand	QXKIT				

Define Processing Options

This chapter contains these topics:

- [Section 12.1, "Overview,"](#)
- [Section 12.2, "Example - Interactive Programs Using Processing Options,"](#)
- [Section 12.3, "Example - Report Program Using Processing Options,"](#)
- [Section 12.4, "Defining Processing Options."](#)

12.1 Overview

Processing options allow individual programs to perform in many different ways. They are analogous to mechanical switches that you set before you run the program. Processing options allow users to enter parameters prior to running a program which cause varied outcomes of the program. Processing options:

- Control which fields appear on the data entry screen
- Control how the program processes data
- Set up certain default values for entry

The Define Processing Options function allows you to define processing options that the program uses. 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 screen.

The system stores information in the DREAM Writer Master Parameter file (F98301). The system retrieves the processing options in the housekeeping subroutine S999 by including a copy member which accesses a program to retrieve the values and enter them into array @OP.

When you define processing options, you must be aware of the following:

- The system makes the following changes to the RPG source code:
 - Creates an O record type in file F98301.
 - Brings in /COPY statement for E81DRPT.
 - Brings in /COPY statement for C81DRPT.
 - Brings in EXSR C81DRPT statement in the housekeeping subroutine S999.
 - Loads processing options to array @OP, which has 99 elements of 25A.
- You must add code manually via the Source Entry Utility (SEU) or Program Design Language (PDL) to use the processing options in a program.

- You add source code in the housekeeping subroutine S999 to move the processing option into a program work field.

Example: `MOVE @OP,1 $PO1 2`

- The system uses the program work field with PDL, or you can manually add source code to the program via SEU. A PDL example follows:

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

- The program needs to have values for Program ID (PSPID) and Version ID (PSVERS) to retrieve the processing option values from the DREAM Writer parameter file:
 - If your program is a report program, the system generates PSPID and PSVERS automatically as PLIST parameters. Therefore, you do not need to do anything.
 - If your program is an interactive program, you must add PSPID(10) and PSVERS(10) as the first two PLIST parameters. Remember to modify any programs that launch this program so that the system delivers these two parameters.

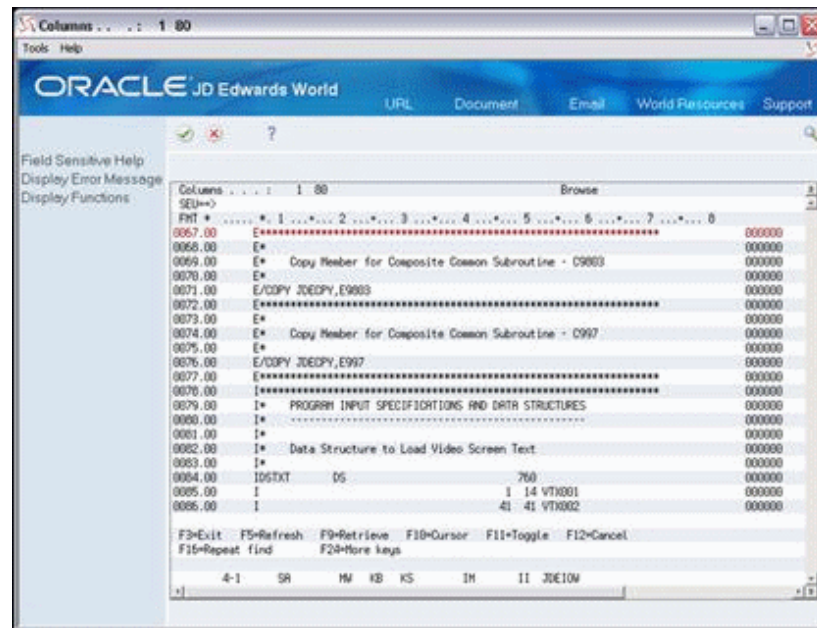
12.1.1 What You Should Know About

Topic	Description
Program Generator	The program generator does not include code that controls interactive processing options. The program generator does control some of the tasks for batch programs. The text you enter on the Processing Options Setup screen does not convey instructions to the program generator.

12.2 Example - Interactive Programs Using Processing Options

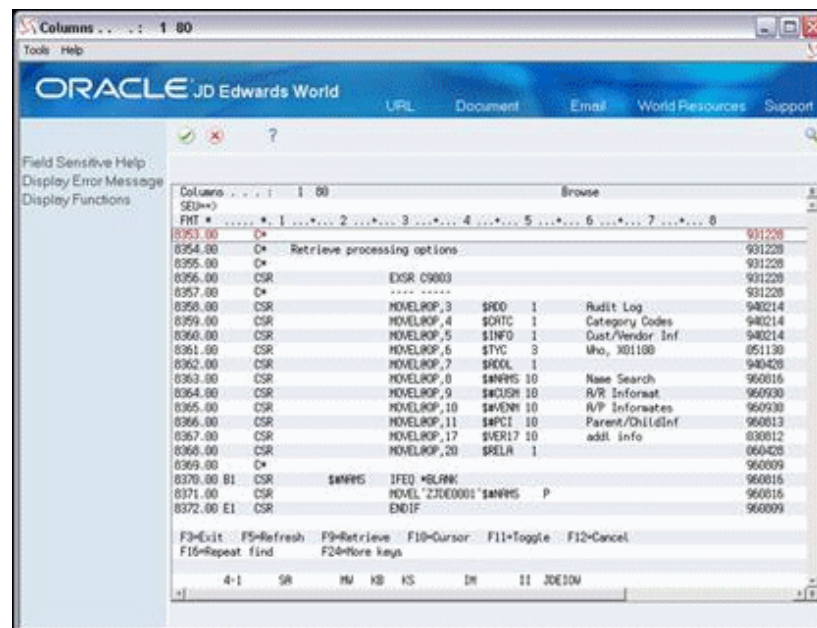
Following is an example of an interactive program containing processing options.

When you search in SEU for string C9803, the following screen displays:

Figure 12-1 Search Results for String C9803 screen

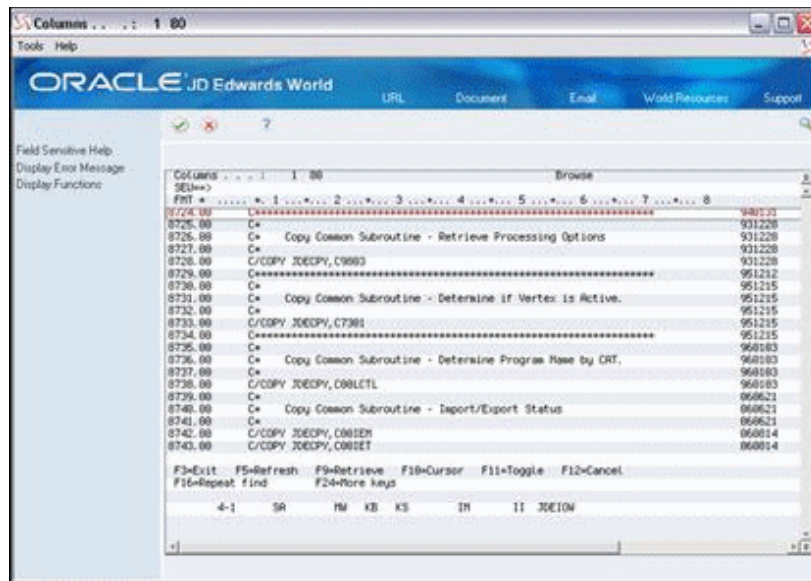
This code copies the E Specs that relate to the common subroutine. This code defines all necessary arrays and tables for the copy module C9803. Typically, you add copy statements such as this at the end of the E Specs.

The next section of code in this program that relates to processing options is as follows:

Figure 12-2 Search Results for String C9803 (Next) screen

The system loads the @OP array for the processing options. @OP1 contains the first of 99 processing option values, which is the value the user enters in the processing option you assign to position 1. The system then moves it into another program field to use.

The next section of code that relates to processing options is as follows:

Figure 12-3 Search Results for String C9803 (@OP1) screen

This code copies the C Specs for C9803. This copy module contains code for the actual subroutine C9803. The code accesses a program that retrieves the processing option values for the DREAM Writer version you execute.

12.3 Example - Report Program Using Processing Options

The program generator builds segments of code that the system requires to processing the processing options. The code that relates to report processing options exists in two locations. The locations are where the program generator copies the:

- Extension Specifications that relate to the common subroutine for retrieving processing options.
- Calculation Specifications that relate to the common subroutine for retrieving processing options.

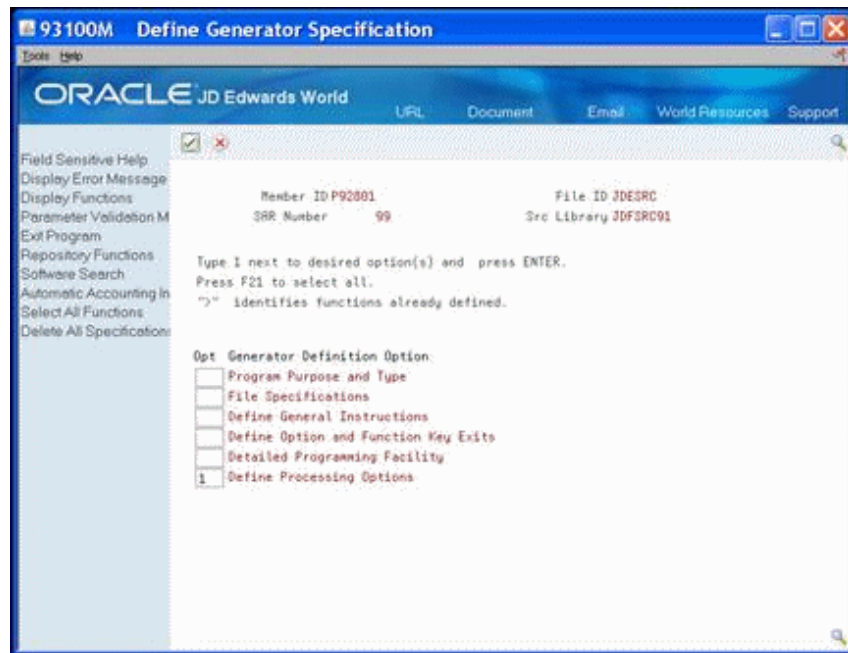
12.4 Defining Processing Options

This section includes the following tasks:

- To enter processing options
- To view code for the processing options

To enter processing options

1. On Define Generator Specifications, enter 1 in the Option field to access Define Processing Options:

Figure 12-4 Define Generator Specification (Processing Options) screen

2. Complete the following fields on Processing Options Setup:

- Sequence
- Text
- Option Number
- Date (0/1/
- RJ (Right Justify)
- Text Only
- DL (Display Level)
- Field Name

Figure 12–5 Processing Options Setup screen

Field	Explanation
Seq	Specifies how the processing option text lines should be ordered on the screen. Not input capable.
Text	The descriptive text for the processing option.
Opt Nbr	<p>The Processing Option Number field specifies for DREAM Writer processing options the array index position for each processing option. This number should never change once assigned. The sequence number of processing options may be changed to allow for better presentation on the Processing Options Entry program, however, you should never change the processing option number because the program includes code specific to the array position for the Processing Option value.</p> <p><i>Screen-specific information</i></p> <p>You can change the sequence number of processing options to allow for better presentation on the Processing Options Entry program, however, you should never change the processing option number because the program includes code specific to the array position for the Processing Option value.</p>
Date (0/1) (0/1/2)	<p>The Date Field specifies whether or not the processing option refers to a date.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> 0 – Indicates that the information is not a date. 1 – Indicates that a date is to be stored in the processing option as a Gregorian date in month, day and year format. 2 – Indicates that a date is to be stored in the processing option as a Julian date in century, year and day format. 3 – Indicates the same as a "2" with the exception that the display AND entry format is "YYYY/MM/DD" (full four digit year). <p>NOTE: All data entry for date information is entered in SYSTEM FORMAT with the exception of the "3".</p>

Field	Explanation
R J(Right Justify)	<p>Determines if the entry field is right-justified. Valid values are:</p> <p>0 – Information is not right-justified</p> <p>1 – Information to be entered is numeric and should be right-justified</p> <p>2 – Information to be entered is to be right-justified and left-filled with blanks</p>
Text Only	<p>The Text Only field is used to specify whether the text line is text only or a processing option value entry line. This allows you to specify multiple lines of text to document each processing option. The values for this field are</p> <p>1 – for text only</p> <p>0 – for a value entry line.</p> <p>Each separate processing option can have only one input value, or "0" value.</p>
D L (Display Level)	<p>This field controls which processing options are displayed to a user based upon the user's Level of Display (LOD) value in the JD Edwards World User Information file. If the User's LOD is equal or greater, PO is displayed.</p>
Field Name	<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"> ■ This is a required field ■ Use #S01 - #S15 for options ■ Use #F01 - #F15 for function keys <p><i>Program-specific information</i></p> <p>The data dictionary item name. Examples include F#MMCO or F#CO for company; F#MMCU or F#MCU for business unit; and F#RP01-30 for business unit category codes 01 through 30.</p> <p>Special characters are not allowed as part of the data item name, with the exception of #, @, \$.</p> <p>If you want to create protected data names without JD Edwards World' interference, use \$xxx and @xxx, with xxx being user-defined.</p> <p>DREAM Writer NOTE: Within the Processing Options Setup screen, the field name is used during data entry to edit field size and other field attributes.</p> <p>Used to validate against the data dictionary.</p> <p><i>Screen-specific information</i></p> <p>Note: On the Processing Options Setup screen, the system uses the field name during data entry to edit field size and other field attributes for DREAM Writer.</p>

To view code for the processing options

1. On Software Versions Repository, locate the program for which you are adding processing options.
2. Enter 1 in the Option field next to the line in the subfile for the program.
The code for the program displays on the Browse screen.

3. Scan for the following instances within the code:
 - Where you instruct the compiler to retrieve the requisite source for the Extension Specification that relate to the C9803 subroutines.
 - Where you interpret and act upon the values in the processing options.
 - Where you instruct the compiler to copy the source for the calculation specifications that relate to the C9803 subroutine.

12.4.1 Function Exits

Repository Services (F6)

Choose Repository Services (F6) to access a screen with a list of JD Edwards World repositories.

Language Preference Text (F18)

Choose Language Preference Text (F18) to access a screen that you use to enter language specific processing options.

Part III

Program Design Language

This part contains these chapters:

- [Section 13, "Overview to Program Design Language,"](#)
- [Section 14, "About PDL Statements and Syntax,"](#)
- [Section 15, "Understand Additional PDL Operations."](#)

Overview to Program Design Language

This chapter contains these topics:

- [Section 13.1, "Objectives,"](#)
- [Section 13.2, "About PDL."](#)

13.1 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

13.2 About PDL

Use PDL to 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 source code.

Enter PDL code prior to the standard code that the program generator creates. If you want the PDL code to follow the standard code for a field, enter the PDL code on the field immediately following the field with which it is associated. The program generator creates all source code for fields in alphabetical order.

CASE stores PDL in the User Defined Procedures file (F93109) with one record per formula. The User Defined Procedures Detail file (F93110) divides the F93109 file into statements. 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 PDL uses:

- Data Item Formula Revisions screen
- 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

About PDL Statements and Syntax

This chapter contains these topics:

- [Section 14.1, "About PDL Statements,"](#)
- [Section 14.2, "About Blocks of Statements,"](#)
- [Section 14.3, "About Comments,"](#)
- [Section 14.4, "About Assignments,"](#)
- [Section 14.5, "About Database Operations,"](#)
- [Section 14.6, "About Program Calls,"](#)
- [Section 14.7, "About Loops,"](#)
- [Section 14.8, "About Conditions,"](#)
- [Section 14.9, "About Miscellaneous Keywords and Syntax."](#)

A Data Item Formula consists of Program Design Language (PDL) statements. PDL statements form the following types of operations.

- Blocks of statements
- Comments
- Assignments
- Database operations
- Program calls
- Loops
- Conditions
- Miscellaneous keywords and syntax

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

14.1 About PDL Statements

A PDL statement combines one or more of the following elements:

- Keywords
- Variables
- Database Files
- Operators

- Constants
- Punctuation

14.1.1 Keywords

Keywords are the vocabulary of PDL. They identify the type of operation the statement performs.

14.1.2 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

You can use Data Dictionary fields in PDL. The system uses the data type and size as they are defined in the Data Dictionary.

- Indicators

You can use indicators by using the names IN01 to IN99. You can also use INLR. You can use both of these 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: In PDL, the system does not use * with indicators. That is, you specify indicator 01 as in01 and not *in01.

- Program Workfields

Any name that PDL recognizes as a variable, but is not a database field name, Data Dictionary field, screen or report field, or indicator, the system considers as a program work field. PDL will prompt you to define its data type.

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

Be aware that if you did not yet generate the source code, PDL is not able to search the source code to find a definition.

14.1.3 Database Files

You must first define a database file name in the File Specifications before you can use it in one of the database I/O statements. PDL does not add file names to the specifications.

14.1.4 Operators

You define the valid assignment and arithmetic operators.

14.1.5 Constants

You specify alpha constants by enclosing them in single quotes. You specify numeric constants without quotes.

Examples:

```
vc0001 := 'Proof Mode';
```

```
$#am1 := 0;
```

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

14.1.6 Punctuation

The basic PDL punctuation is a semi-colon (;), which you must use to separate PDL statements.

14.2 About Blocks of Statements

14.2.1 Keywords and Syntax

Keywords	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

14.2.2 Rules

- You must enter all Data Item Formulas within a Begin...End block. A comment statement may precede the Begin statement. For example:

```
\ Use system date as default. \
```

```
Begin
```

```
If vdttrdj = ' ' Then
```

```
    vdttrdj := $$edt;
```

```
End
```
- You must separate all statements within a Begin...End block by a semicolon. For example:

```
\ Load A/B name to vc0 field. \
```

```
Begin
```

```
aban8 := q3an8;
```

```
chain f0101la;
```

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

```
vc0003 := abalph;
```

```
End
```

- You can nest Begin...End up to a maximum of 50 levels. For example:

```
\ Computer counter. \
```

```
Begin
```

```
If zaclst = '900' Then
```

```
    Begin
```

```
        rr#nin := '1';
```

```
        $#nin := 1;
```

```
    End;
```

```
If zaclst < '900' Then
```

```
    Begin
```

```
        rr#nin := '0';
```

```
        $#nin := 0;
```

```
    End
```

```
End
```

14.3 About Comments

14.3.1 Keywords and Syntax

Syntax	Explanation
\ (backslash)	Initiates and terminates a comment. The syntax is: \ text \ You must enclose all comments within a pair of backslashes.

14.3.2 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
```

14.4 About Assignments

14.4.1 Operator and Syntax

Operators	Explanation
<code>:=</code>	The assignment operator. The system assigns the first variable the value of the variable or expression following the operator. The syntax is: variable := expression;
<code>+</code>	Add
<code>-</code>	Subtract
<code>*</code>	Multiply
<code>/</code>	Divide
<code> </code>	Concatenate
<code> ></code>	Blank and Concatenate
<code> <</code>	Truncate and Concatenate
<code>SST</code>	Substring The syntax is: variable := SST (field,n1,n2) n1 = start position n2 = length of string

14.4.2 Rules

You can use standard notation using parentheses for arithmetic operations.

For example:

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

14.5 About Database Operations

14.5.1 Keywords and Syntax

Keywords	Explanation
Chain	Provides for random data base processing. The syntax is: CHAIN file;

Keywords	Explanation
Delete	Provides the ability to delete the current 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 or greater than the key value you specify. 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 you specify. The syntax is: POSGT file;
Read	Provides for sequential data base processing by reading the next record in the file you designate. The syntax is: READ file;
Readc	Provides for processing of workstation files to obtain the next record change in a subfile. They syntax is: READC file;
Reade	Provides for sequential data base processing by reading the next record in the file with a key equal to the one you specify. The syntax is: READE file;
Readp	Provides for sequential data base processing by reading the record previous to the record read in the file you designate. They syntax is: READP file;
Update	Provides the ability to update the current data base record. The syntax is: UPDATE file;
Write	Provides the ability to add a new data base records. The syntax is: WRITE file;

14.5.2 Rules

You must first define the file in the program using the File Specifications before you enter it in the statement.

The Chain, Poseq, Posgt, and Reade statements use the default KLIST name that the system generates for the file you specify.

You should assign a value to each field of the KLIST prior to entering the statement.

Specify indicator 98 in the statements to signify that the system did not retrieve a record for the program.

Specify indicator 99 in the statements to signify that a database operation error took place.

\ Load A/B name to vc0 field. \

Begin

aban8 := q3an8;

chain f01011a;

If in98 = '0' Then

vc0003 := abalph;

End

14.6 About Program Calls

14.6.1 Keywords and Syntax

Keywords	Explanation
Call	Allows you to execute another program. The syntax is: CALL variable;
Parm	Allows you to deliver parameters to a program that the program call statement executes. The syntax is: PARM variable;

14.6.2 Rules

Neither the Call statement nor the Parm statement allows 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 launch, and load one or more variables with the values of the parameters.

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

14.7 About Loops

14.7.1 Keywords and Syntax

Keywords	Syntax
Until	Provides for loop processing where the system evaluates a condition at the bottom of the loop. <ul style="list-style-type: none"> ■ Translates to DOU in the RPG code. The syntax is: UNTIL (condition) DO (Statement)

Keywords	Syntax
While	Provides for loop processing where the system evaluates a condition at the top of the loop. <ul style="list-style-type: none"> Translates to DOW in the RPG code. The syntax is: WHILE (condition) DO (Statement)
Do	An integral part of the loop statement.

14.7.2 Rules

The Do keyword is an integral part of the loop statement.

The statement following Do can be a single statement, or a block of statements within a Begin...End block.

The action is simply two expressions that you separate.

For example:

Begin

While in98 = '0' Do

Begin

\$#xtp := q2xqt * q2uncs;

\$#am1 := \$#am1 + \$#xtp;

reade f59422;

End

End

14.8 About Conditions

14.8.1 Keywords and Syntax

Keywords	Explanation
If	Provides for conditional processing. <ul style="list-style-type: none"> § The condition is two expressions that you separate by a relationship. § The data types of the expressions have to match. For example, alpha to alpha, numeric to numeric.
Then	Specifies the starting point for all actions the system takes when the condition of the If statement is met.
Else	Enter these statements following the If and Then statements. The system executes these statements when the condition of the If statement is not met.

The Then keyword is an integral part of the If statement.

- The statement following the Then keyword can be a Begin/End block to allow for a block of statements when the condition is met.

- The Else statement can follow the statements you enter with If (condition) and Then (statement).
- The syntax is: IF (condition) THEN (statement) ELSE (statement)

14.8.2 Symbols

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

14.8.3 Rules

You do not have to enter the semicolon (;) to end the statement following the Else, or the Then when there is no Else.

For example, a simple If...Then statement:

Begin

If sftrdj = ' ' Then

sftrdj := \$\$edt

End

For example, an If...Then...Else statement

Begin

If in98 = '0' Then

vc0003 := abalph

Else

vc0003 := 'NOT DEFINED'

End

If you nest Begin/End blocks between the Then and Else statements, you should use the semicolon after each individual statement but not following the End.

For example, an If...Then with a Begin...End statement

Begin

If zaclst = '999' Then

Begin

rr#nin := '1';

\$#nim := 1;

End

End

For example, an If...Then...Else with Begin...End statement

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

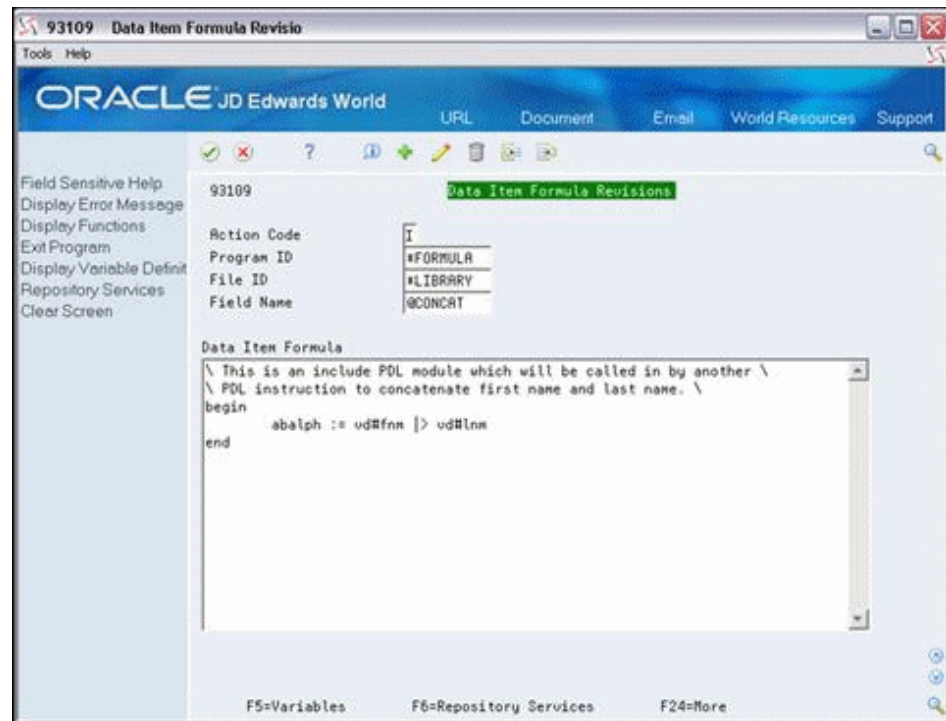
14.9 About Miscellaneous Keywords and Syntax

14.9.1 Keywords and Syntax

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

14.9.2 Rules for Include

You can keep PDL modules in the form of a copy book by designating *FORMULA in the Program ID field and *LIBRARY in the File ID field on the Data Item Formula Revisions screen.

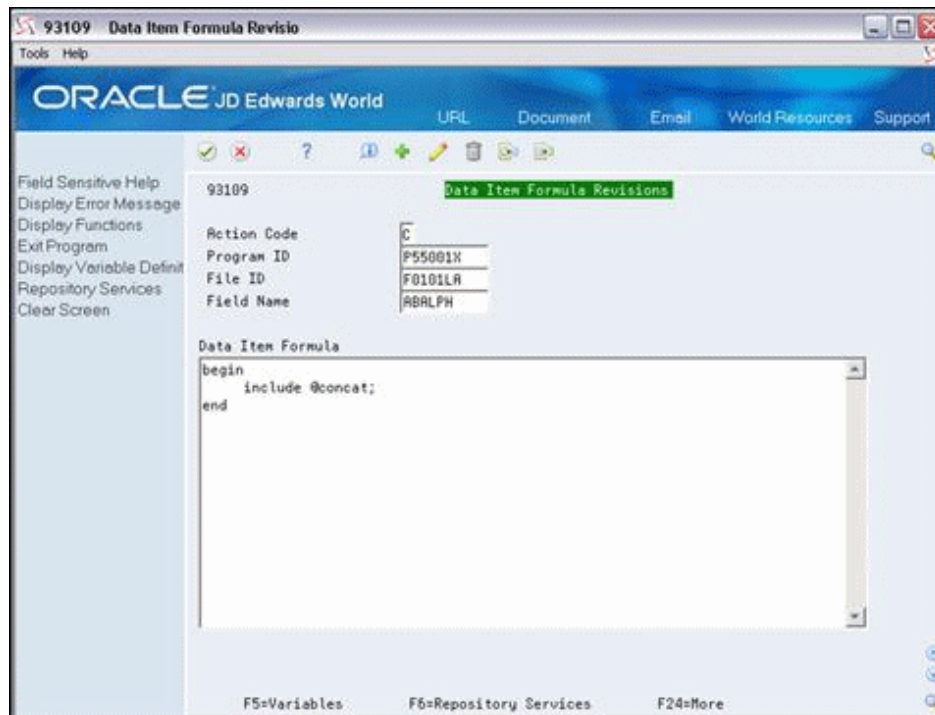
Figure 14-1 Data Item Formula Revisions screen

The include module should have a unique name, for example @CONCAT.

It is JD Edwards World naming convention to begin module names with the @ symbol.

The keyword include causes the Program Generator to automatically generate the appropriate code for the include module. This prevents the need to reenter user calculations that are necessary in numerous programs.

Following is an example of an include module and the include statement that calls the module.

Figure 14–2 Data Item Formula Revisions (Include Module) screen

14.9.3 Rules for Return

Specifying the Return keyword is the same as entering *PROC in the Read From field in the Detail Programming Facility.

The system omits all standard processing for this data field. In other words, by specifying the Return keyword, the system uses the code the PDL generates instead of any standard logic.

For example:

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

Understand Additional PDL Operations

This chapter contains these topics:

- [Section 15.1, "Editing,"](#)
- [Section 15.2, "Parsing,"](#)
- [Section 15.3, "Source Code Generation,"](#)
- [Section 15.4, "Add PDL to a Field,"](#)
- [Section 15.5, "Function Exits."](#)

15.1 Editing

The Data Item Formula is one long continuous field. If there is an error, the entire field displays in reverse image, and the system places the cursor in the field following the error. You can display the error messages by choosing Display Error Message (F7).

15.2 Parsing

The system stores the Data Item Formula in the File Specifications database in two forms:

- The generator stores the unparsed form in the User Defined Procedures file (F93109), with one record for one formula.
- The generator stores the parsed form in the User Defined Procedures Detail file (F93110), with multiple records for each formula. Each record corresponds to an RPG operation code.

15.3 Source Code Generation

The generator merges the PDL code into the program based on the field you enter 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 generator places the PDL code before the standard code Program Generator code for the field in the Write To field. If you want the code the PDL generates to replace the standard code, then enter *PROC in the Read From field.

Note: Use caution when performing this as the system performs no editing or formatting of the field, except what you enter in the Data Item Formula.

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

The generator lists the fields in the Detailed Programming Facility in alphabetical order, and you cannot change the order.

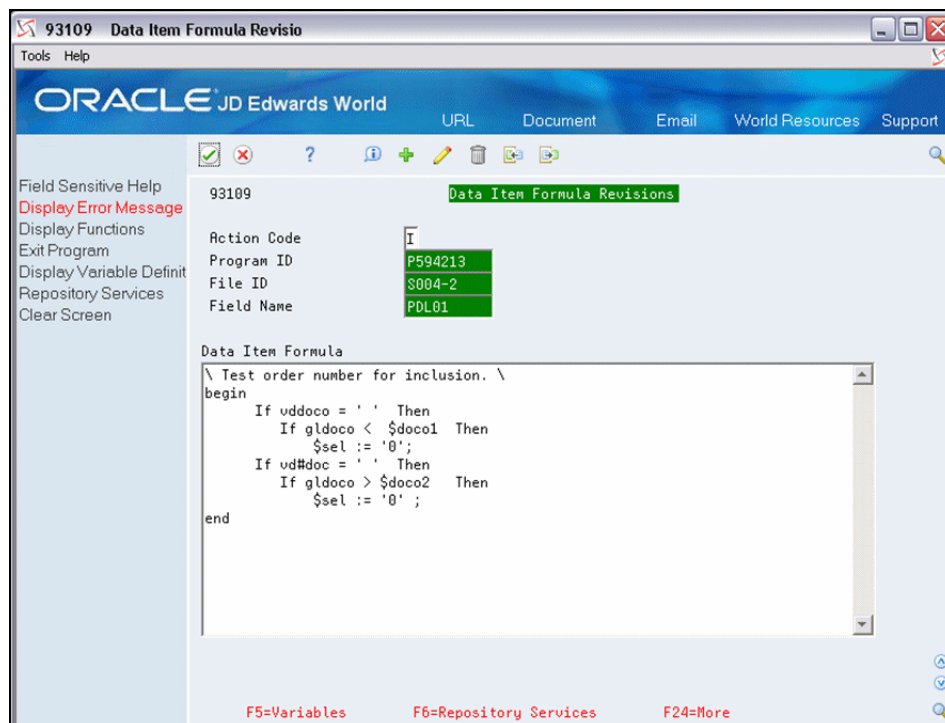
15.3.1 Data Item Formula Examples

Two examples illustrate the PDL statements and syntax. Both are from an inquiry program with a subfile.

15.3.1.1 Example User Defined PDL

The example illustrates a data item formula for 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 (\$SEL = 0 to omit). The program work fields \$doco1 and \$doco2 contain the lower and upper values for the inquiry search fields with a subfile.

Figure 15–1 Data Item Formula Revisions (User Defined PDL) screen



This example also illustrates the following types of PDL statements:

Type of PDL Statement	Description
Assignment	\$sel := '0'
Blocks	begin...end
Comment	\ Test order number for inclusion. \
Condition	If q1doco < \$doco1 Then \$sel := '0';

Additionally, this example illustrates the nesting of conditions:

If vddoco = ' ' Then

If q1doco < \$doco1 Then

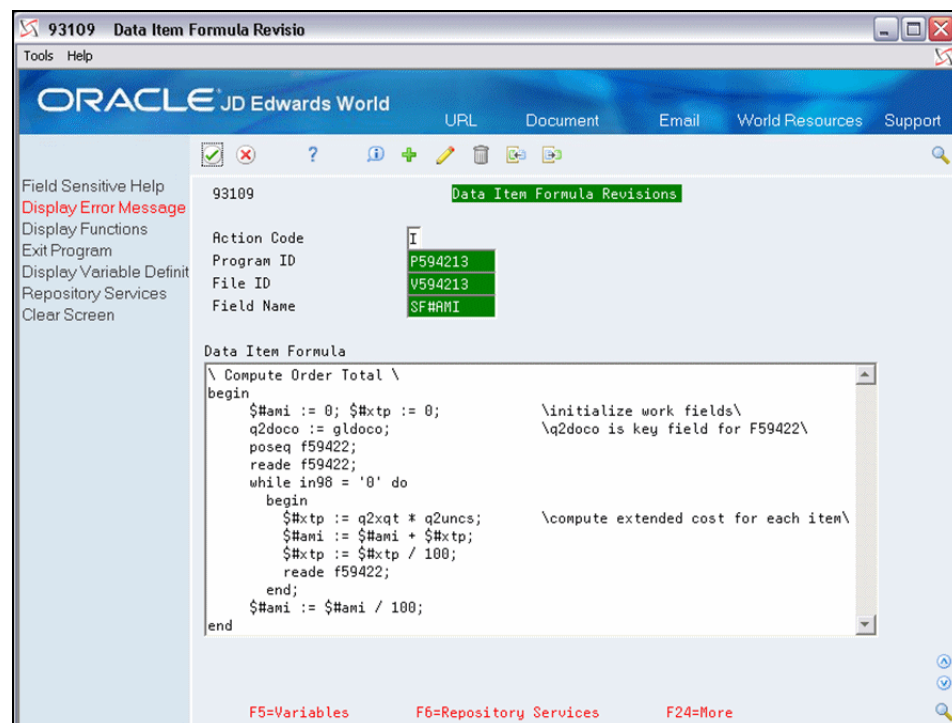
??\$sel := '0';

The statement separator, the semicolon, is not necessary until the outermost If...Then statement is complete.

15.3.1.2 Example Subfile Field

This example illustrates a data item formula for a subfile field that is a computed field. The program is locating sales order header records. The computed field is the order total and is based on the sales order detail records in F59422.

Figure 15–2 Data Item Formula Revisions (Subfile Field) screen



This example illustrates the following types of PDL statements:

Type of PDL Statement	Description
Assignment	\$#ami := 0; \$#xtp := 0; q2doco := gldoco;
Blocks	Notice the begin...end nested within the while...do
Comment	Notice the embedded comments as well as the heading comment
Database	poseq f59422; reade f59422;
Loops	while in98 = '0' do begin...end

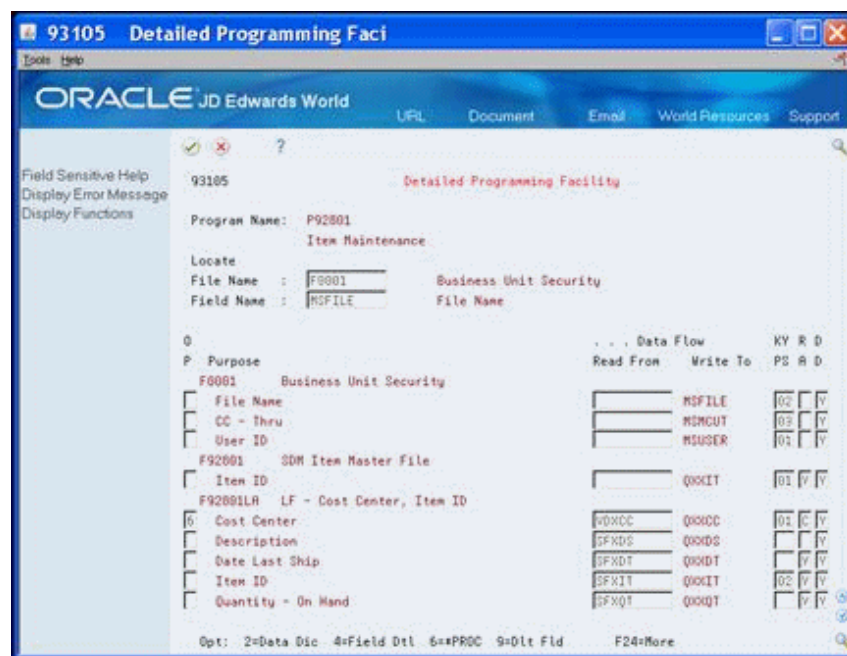
You can separate statement separators in the loop statements within the Begin...End block, and then follow the end statement with a separator.

15.4 Add PDL to a Field

To work with Data Item Formula Revisions

1. On Software Versions Repository, locate a program and then access Define General Specifications.
2. On Define General Specifications, enter 1 the Option field to access Detailed Programming Facility.
3. On Detailed Programming Facility, enter 6 in the Option field next to the field for which you want to add PDL.

Figure 15–3 Detailed Programming Facility (Revisions) screen



The Data Item Formula Revisions screen displays.

Figure 15–4 Data Item Formula Revisions (Add PDL) screen

4. Enter the PDL statements for the field in the Data Item Formula area.

Field	Descriptions
Program ID	The RPG program name defined in the Software Versions Repository Master table.
File ID	The member ID of the file used by the program.
Field Name	This 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.

15.5 Function Exits

Display Variable Definitions (F5)

Choose Display Variable Definitions (F5) to access a screen with a list of variable definitions.

Repository Services (F6)

Choose Repository Services (F6) to access a screen with a list of JD Edwards World technical functions or repositories.

Part IV

Source Modifications

This part contains these chapters:

- [Chapter 16, "Overview to Source Modifications,"](#)
- [Chapter 17, "Change Generated Source Code,"](#)
- [Chapter 18, "Regenerate Source Code,"](#)
- [Chapter 19, "Work with Model Control Language Programs."](#)

Overview to Source Modifications

This chapter contains these topics:

- [Section 16.1, "Objectives,"](#)
- [Section 16.2, "About Source Modifications."](#)

16.1 Objectives

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

16.2 About Source Modifications

After you generate source code, you can make modifications to it by using the Program Generator steps or using Source Entry Utility. When you make changes 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

This chapter contains the topic:

- [Section 17.1, "Pre-SEU and Post-SEU Process."](#)

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 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.

There are two different methods to change generated source code.

17.1 Pre-SEU and Post-SEU Process

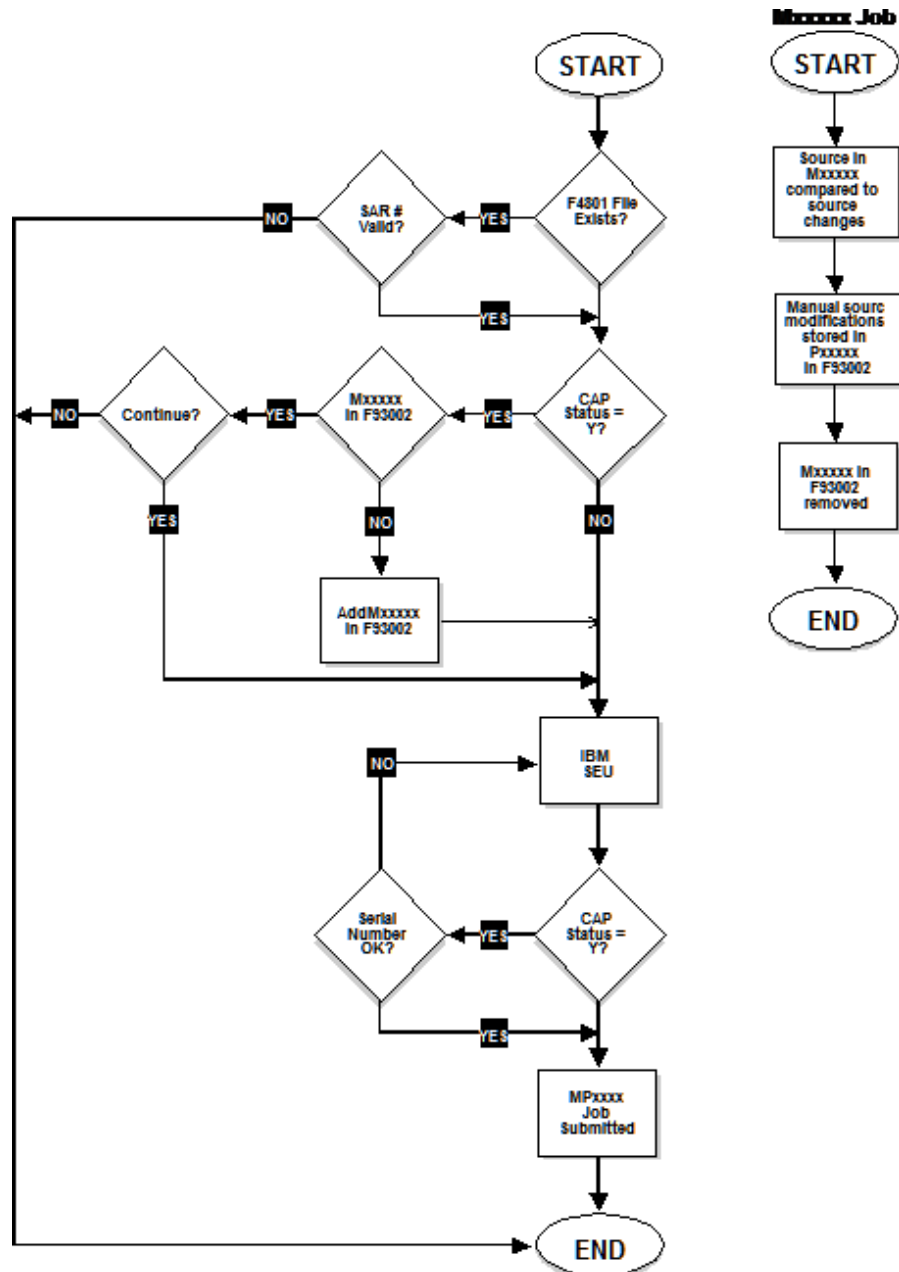
A front-end JD Edwards World program, MPxxxxx, monitors the changes on the SEU. The MPxxxxx job does not have to finish before you recompile. The system automatically merges the changes on the SEU when you generate the program, not when you compile. The system stores all changes on the SEU in the Pxxxxx member in the Additional Help/Modifications Master file (F93002).

For source code lines that you move or copy, you must clear the serial number from column 80 onward.

You can view all changes on the SEU by entering 30 in the Option field on the Software Versions Repository.

Columns 1 and 2 include a 21 for lines you add, 22 for changes, and 23 for lines you delete.

Figure 17-1 Pre-SEU and Post-SEU Processes

**To change generated source code**

From the Software Versions Repository, locate a program and perform one of the following:

1. Enter 2 in the following field to access the JD Edwards World SEU feature
 - Option
2. On SEU, make your changes.

When you change your program using SEU, you do not have to regenerate the code. You only need to recompile the changes.

Alternatively, after you locate a program on Software Versions Repository you can:

1. Access Define General Specifications.
2. On Define General Specifications, enter 1 the following field to access Detailed Programming Facility.
 - Option
3. On Detailed Programming Facility, enter 6 in the following field next to the field for which you want to change PDL.
 - Option
4. On Data Item Formula Revisions, make your changes.

Regenerate Source Code

This chapter contains these topics:

- [Section 18.1, "When to Regenerate Source Code,"](#)
- [Section 18.2, "Changing CAP Status,"](#)
- [Section 18.3, "Resolving CAP Status Invalid Error."](#)

When regenerating source code you should know:

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

18.1 When to Regenerate Source Code

You should regenerate a program whenever you modify a program specification. 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 enter the new information into the Detailed Programming Facility.

- 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 did not have any, or delete all Processing Options
- Change the Lockout Action field values

When you add, change, or remove a file in the program or change the program type, you must access the File Specifications screen from the Define Generator Specification screen and click Enter to submit the Detailed Field Specifications interactive job.

You can view all changes on the SEU by entering 30 in the Option field on the Software Versions Repository.

Columns 1 and 2 include a 21 for lines you add, 22 for changes, and 23 for lines you delete.

18.2 Changing CAP Status

If you change the CAP Status field, the system deletes the changes in the Source Entry Utility that it stores in the P member of the Additional Help/Modifications Master file (F93002). JD Edwards World recommends that you do not change the CAP Status field unless the changes you make to your program become unmanageable. When the CAP Status field is set to Y, you can regenerate your program from one JD Edwards World release to the next.

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

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

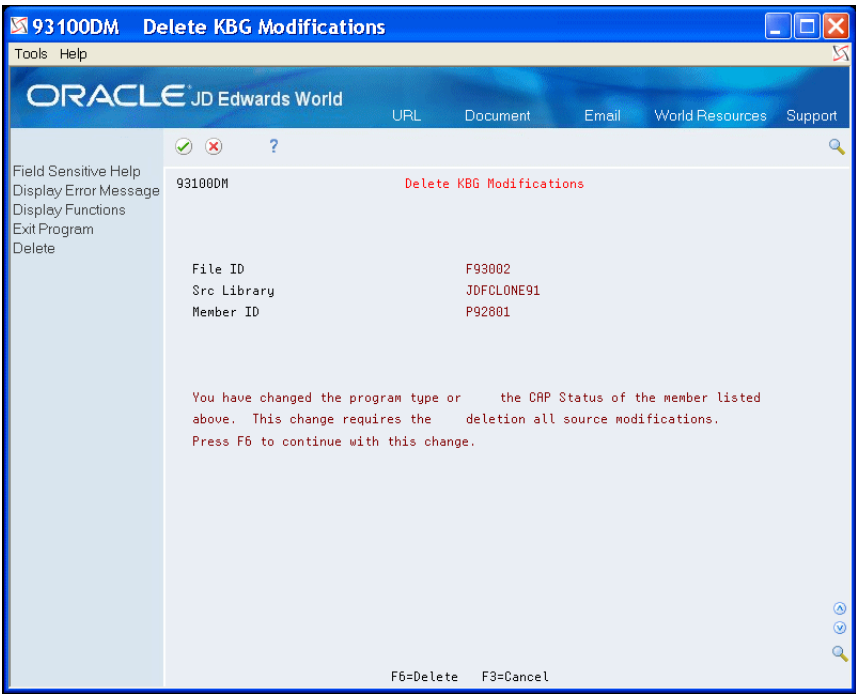
To change CAP status

1. On Define Generator Specification, enter 1 in the following field next to Program Purpose and Type.
 - Option
2. On Program Purpose and Type, enter N in the following field.
 - CAP Status

Figure 18–1 Program Purpose and Type screen

The Delete KBG Modifications screen displays.

Figure 18–2 Delete KGB Modifications screen



- 3. To remove the modifications member, choose Delete (F6).
The Define Generator Specification screen displays.

18.3 Resolving CAP Status Invalid Error

The Program Generator verifies that the job completes normally before each source generation. When the program generator does not complete normally or if you delete the specifications for a program, the system sends an error message to your workstation that states: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 the record, or field AGSRCS is not found, verify that the JDESRC file in your source library exists and has a length of 142 and 8 fields.

To resolve the CAP Status Invalid error, perform any of the following:

Possible Resolution	Description
Ensure the CAP status is set to Y on the Program Purpose and Type screen.	<p>Any job that prevents the MPxxxxx job from completing normally will change the CAP Status to N.</p> <ul style="list-style-type: none">Allow the MPxxxxx job to complete.Do not cancel it in the job queue. <p>If you change the CAP Status field to N, the system deletes the changes in the Source Entry Utility that it stores in the P member of the Additional Help/Modifications Master file (F93002).</p> <p>If the File Specifications ends abnormally, the system changes the value in the CAP Status field to D. Change the value in the CAP Status field to Y and process the file specifications.</p>

Possible Resolution	Description
Ensure the Pxxxxx member exists in the Additional Help/Modifications Master file (F93002).	The Pxxxxx member must exist in order to generate a program. The system initially creates the Pxxxxx member during the Program Purpose and Type definition step.
Ensure 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.
Ensure that one step of the generation process completes before you start the batch job of another step.	NA

Work with Model Control Language Programs

This chapter contains these topics:

- [Section 19.1, "Working with CL Models,"](#)
- [Section 19.2, "JD Edwards World Model CL 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.

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

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

Note: You can view only the source code if the source code resides on your machine.

19.1 Working with CL Models

To copy a model CL

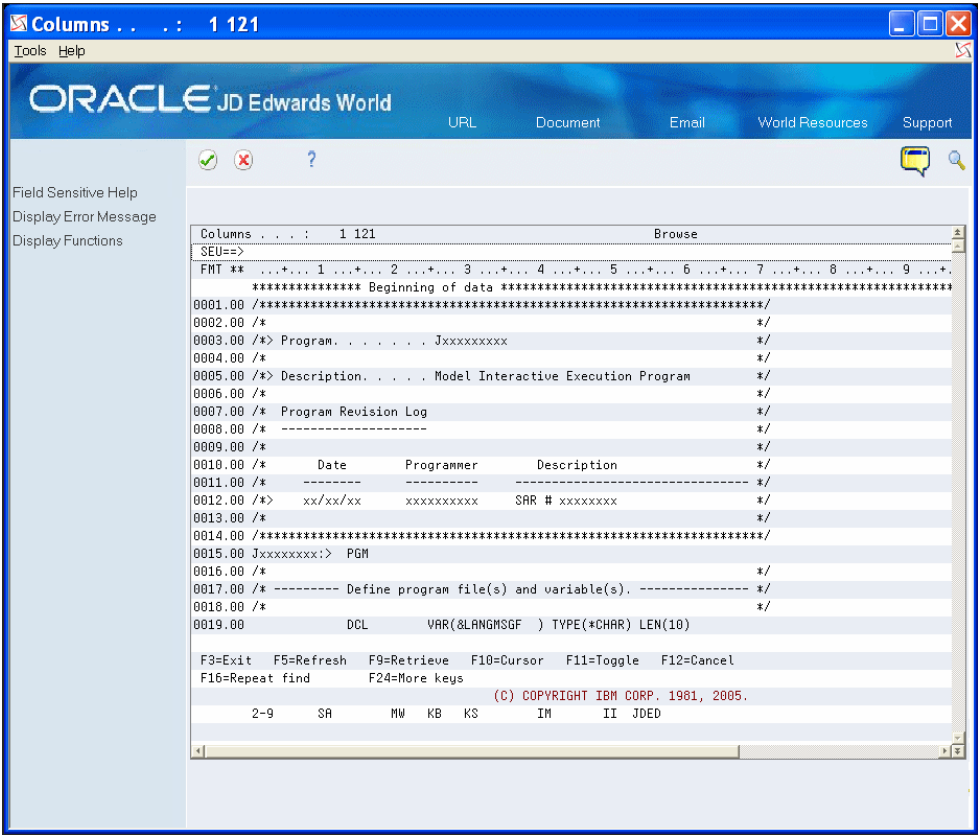
1. On Software Versions Repository, locate a model.
2. Enter 3 in the Option field next to the program.
3. Click Enter in the Copy Source Prompt window.
4. On Software Versions Repository, enter 2 in the Option field next to the program.

The source code displays.

5. On the Source Entry Utility, enter the new program name.

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

Figure 19–1 Source Code for J98MODEL1 screen



To customize a CL model

1. On the Source Entry Utility, locate the lines that contain lower-case xx and make changes.

The lines in the model that require changing contain lower-case xx. This design allows you to easily scan the code for the xx and enter your changes.
2. Exit and save the CL program.
3. Compile the program.

19.2 JD Edwards World Model CL Programs

JD Edwards World includes a series of model CL programs that you can copy and customize to meet your programming needs. The following table describes each model CL program.

Model CL Programs	Description
J98MODEL1	Serves as a template for all interactive programs that do not retrieve processing options in the CL code.
J98MODEL2	Serves as a template for batch programs that need the DREAM Writer but have no printer file.
J98MODEL3	Serves as a template for interactive programs that need a prompt for parameters.

Model CL Programs	Description
J98MODEL4	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.
J98MODEL5	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.
J98MODEL6	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.
J98MODEL7	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.
J98MODEL8	Serves as a template for batch programs that have a control file.

You can create certain model CL programs using the Quick Start CL Generator. See [Chapter 25, "Work with Quick Start CL Generator"](#) for more information.

Part V

CASE Programs

This part contains these chapters:

- [Chapter 20, "Overview to CASE Programs,"](#)
- [Chapter 21, "Overview to Subfile Inquiry Programs,"](#)
- [Chapter 22, "Overview to Subfile Maintenance Programs,"](#)
- [Chapter 23, "Create Report Programs."](#)

Overview to CASE Programs

This chapter contains these topics:

- [Section 20.1, "Objectives,"](#)
- [Section 20.2, "About CASE Programs."](#)

20.1 Objectives

- To create CASE programs

20.2 About CASE Programs

Perform the following tasks:

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

Note: The Report Design Aid is currently not available in the Java platform for JD Edwards World software. You must create reports using the green screen platform of the software.

Overview to Subfile Inquiry Programs

This chapter contains these topics:

- [Section 21.1, "Program Type Description,"](#)
- [Section 21.2, "Display File Definition,"](#)
- [Section 21.3, "CL Program Definition,"](#)
- [Section 21.4, "File Specifications,"](#)
- [Section 21.5, "Detailed Programming Facility,"](#)
- [Section 21.6, "Special Considerations,"](#)
- [Section 21.7, "Quick Start Generation."](#)

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

The use and values you enter to create a typical Interactive subfile inquiry program follow.

21.1 Program Type Description

Use the Program Type Description to create an interactive subfile program that is for inquiry purposes only. This program type processes a single master file by key. You do not use Lockout Action Codes fields. Create a display file prior to generating this program type.

21.2 Display File Definition

The Display File Definition program type validates and changes, where necessary, the data a user enters (scrubs) the key fields in the control format of the display file prior to processing the master file. You denote the key fields by enter K in the Edited Field in the Field Definition screen of Screen Design Aid (SDA). If you are using the Data Base Field Selection feature in SDA, the system updates the key fields.

You do not need to define the Action Code, it is an optional field. Define a default cursor location if there is no action code.

21.3 CL Program Definition

Using the CL Program Definition, you can copy and revise the J98MODEL1 model CL Program and create a CL program for use with program type A0010. You can also use the Quick Start CL Generator to create your CL programs.

21.4 File Specifications

The File Specifications program type requires that you define a single master file and a display file. The master file contains a value of M or 1 in the Input column. The display file begins with V and the selection columns are blank. You can add files to retrieve descriptions, if necessary.

21.5 Detailed Programming Facility

The Detailed Programming Facility allows you to 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 on the Detailed Programming Facility to include only those data items which are necessary. This key list should match your key field definition from the control format of the display file.

21.6 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 loads.

21.7 Quick Start Generation

You can generate this program type using Quick Start.

Overview to Subfile Maintenance Programs

This chapter contains these topics:

- [Section 22.1, "Program Type Description,"](#)
- [Section 22.2, "Display File Definition,"](#)
- [Section 22.3, "CL Program Definition,"](#)
- [Section 22.4, "File Specifications,"](#)
- [Section 22.5, "Detailed Programming Facility,"](#)
- [Section 22.6, "Special Considerations,"](#)
- [Section 22.7, "Quick Start Generation."](#)

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

The use and values you enter to create a typical Interactive Subfile Maintenance Program follow.

22.1 Program Type Description

Use the Program Type Description program type to create 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 exits are optional.

22.2 Display File Definition

The Display File Definition program type validates and changes, where necessary, the data a user enters (scrubs) the key fields in the control format of the display file prior to processing the master file. You denote the key fields by enter K in the Edited Field in the Field Definition screen of Screen Design Aid (SDA). If you are using the Data Base Field Selection feature in SDA, the system updates the key fields.

You must define the Action Code and the Lockout Action Codes are optional.

This subfile maintenance program type includes special logic which permits the system to delete individual subfile records. This logic performs when you enter a C in the Action Code. The system compares the previous value with the current value and deletes the record if the current value is blank. The system stores the previous value in a hidden field at the subfile record level using the Display All Defined Fields in SDA.

22.3 CL Program Definition

Using the CL Program Definition, you can copy and revise the J98MODEL1 model CL program to create a CL program for use with program type D0040. Use the Quick Start CL Generator to create your CL program.

22.4 File Specifications

The File Specifications program type requires that you define a single master file and a display file. The master file contains a value of M or 1 in the Update column. The display file begins with V and the selection columns are blank. Add files to retrieve descriptions, if necessary.

22.5 Detailed Programming Facility

You use the Detailed Programming Facility to access the Full Data Field Parameters screen, which contains details for the subfile field controlling the database update. By entering N in the Update the Entry Optional Y/N field, this informs the generator that the user must enter a value in this field before the system updates the database.

22.6 Special Considerations

The Special Considerations 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 optional entry field.

22.7 Quick Start Generation

You can generate this program type using Quick Start.

Create Report Programs

This chapter contains these topics:

- [Section 23.1, "Understanding RDA Special Use Fields,"](#)
- [Section 23.2, "Creating a Total Format,"](#)
- [Section 23.3, "Defining a Subheading,"](#)
- [Section 23.4, "Understanding DREAM Writer Considerations."](#)

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

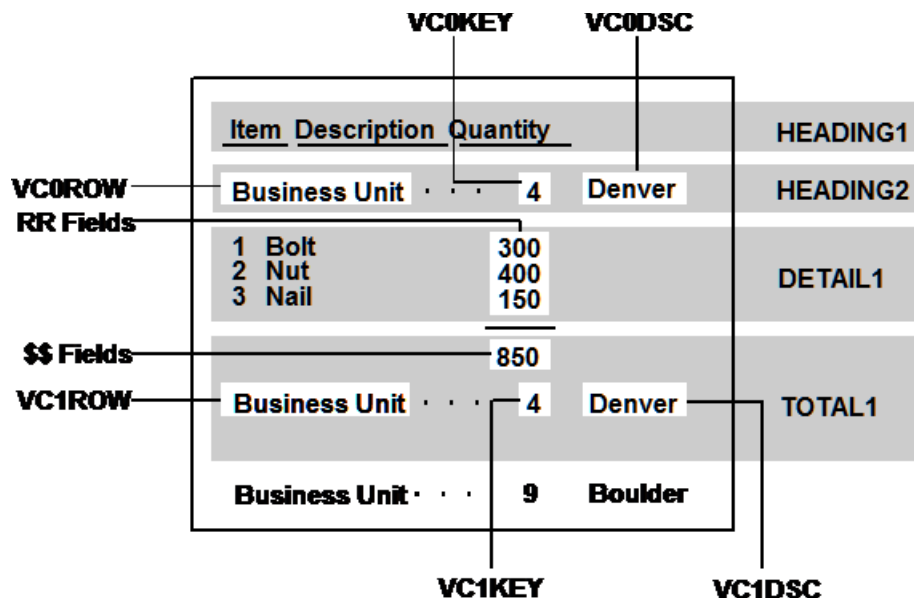
You should be familiar with the definition and use of report totals and subheadings when using the CASE tools. You should also be aware of some DREAM Writer considerations. See *Work with DREAM Writer* in the *JD Edwards World Technical Foundation Guide* for more information.

Note: The Report Design Aid is currently not available in the Java platform for JD Edwards World software. You must create reports using the green screen platform of the software.

23.1 Understanding RDA Special Use Fields

The system uses certain fields in RDA when generating reports that contain dynamic (hierarchical) totaling and subheadings. The following figure and tables illustrate how the system uses these fields within a report.

Figure 23-1 RDA Special Use Fields



The system uses the following fields in the TOTAL1 format:

Field	Explanation
VC1ROW	Prints the data dictionary row description of the level break field. Default length is 30.
VC1KEY	Prints the value of the level break field. Default length is 12.
VC1DSC	Prints the description of the value of the break field. Default length is 30. Only works with the following fields: <ul style="list-style-type: none"> ■ User defined codes ■ Company Number ■ Address Book Number ■ Business Unit

The system uses the following fields in only the HEADING2 format and therefore it uses them in only a C0020 or C0025 program type - Report w/Subheadings.

When you use subheadings, the system automatically underlines them.

Field	Explanation
VC0ROW	Prints the data dictionary row description of the level break field. Default length is 30.
VC0KEY	Prints the value of the level break field. Default length is 12.
VC0DSC	Prints the description of the value of the break field. Default length is 30. Only works with the following fields: <ul style="list-style-type: none"> ■ User defined codes ■ Company Number ■ Address Book Number ■ Business Unit

In programs you generate using CASE; the level breaks are soft coded. The DREAM Writer setup determines this.

23.2 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 level break
- The description of that value

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

Business Unit 5 San Francisco

To create a total format

1. On Software Versions Repository, locate the report for which you want to add a total format.
2. On Software Versions Repository, enter 10 in the Option field to access the Report Design Aid.
3. On Report Design Aid, choose Record Formats List (F10).

Figure 23–2 *Record Formats List* report

[illegible]

4. On the Record Formats List screen, complete the following fields to enter the TOTAL1 format:
 - Format Name
 - Type
 - Start/End Lines
 - Fld Pfx (Field Prefix)
5. Press enter to return to Report Design Aid.
6. On Report Design Aid, enter an asterisk (*) in the column and row position to begin the total description.

The Field Definition screen displays.

Figure 23–3 Field Definition screen

```

928400                                00000000000000000000000000000000
                                Inventory by Cost Center
                                00000000000000000000000000000000
                                00000000000000000000000000000000

Business      Cost Center      It      Item Type
Unit          Description      T T      Description      u
-----
000000000000 00000000000000000000000000000000 00 0000000000000000000000000000 0
*
                                00000000000000000000000000000000

Report: R928400      Field Definition      Format: TOTAL1
Dict Name      Text
Data Type      Field Name VC1ROW      Cond Ind
Row/Column 11 55 Field Use 0      Highlight =
Size 30 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

```

7. On Field Definition, enter VC1ROW in the Field Name field.

8. Click Enter twice.

The description for the total field replaces the asterisk (*).

In the sample report that follows, when you print the report, the field contains the descriptive text Business Unit.

9. On Report Design Aid, enter an asterisk (*) in the column and row position to display the key value.

The Field Definition screen displays.

10. On Field Definition, enter VC1KEY in the Field Name field.

Figure 23–4 Field Definition (VC1KEY) screen

```

00000000000000000000000000000000      Page - . . . 6666
Inventory by Cost Center      Date - . . . 66666666
00000000000000000000000000000000
00000000000000000000000000000000

Item Type      Item      Item      Ship      Quantity
Description    umber r    Description    Date      On Hand      U U-
-----
00000000000000000000000000000000 00000000 00000000000000000000 00000000 000000000000 00
00000000000000000000000000000000* 000 00000000000000000000000000000000 00000000000000

Report: R928400      Field Definition      Format: TOTAL1
Dict Name      Text
Data Type      Field Name VC1KEY      Cond Ind
Row/Column 11 52 Field Use 0      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

```

11. Click Enter.

In the sample report that follows, when you print the report, the field contains the value of 5.

- The Field Definition screen displays.

- 13.** On Field Definition, enter VC1DSC in the Field Name field:

Figure 23–5 *Field Definition (VC1DSC) screen*

```
000000000000000000000000000000000000
Inventory by Cost Center                               Page - . . . 6666
Date = . . . 66666666

000000000000000000000000000000000000
000000000000000000000000000000000000
```

Item Type Description	Item number r	Item Description	Ship Date	Quantity On Hand	U U
----- 000000000000000000000000000000000000	00000000	000000000000000000000000000000000000	00000000	00000000000000	00
-----	000	000000000000000000000000000000000000		0000000000000*	

Report: R928400 Field Definition Format: TOTAL1

Dict Name	_____	Text	
Data Type	A	Field Name	VC1DSC
Row/Column	11 66	Field Use	- Highlight
Size	30	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

- The description for the key value replaces the asterisk (*).

In the sample report that follows, when you print the report, the field contains the value San Francisco.

- The field that contains data for the Quantity on Hand column is RRXQTY. The system places the total amount of Quantity On Hand 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 following illustrates the report.

Figure 23–6 Inventory by Business Unit Report

928400		J.D. Edwards & Company				Page No. . . . 2	
Inventory by Business Unit Report							
Date - 12/02/17		It		Item		Ship	
Bus Unit	Description	Ty	Description	Number	Description	Date	Quantity
5 San Francisco Branch		N	Non-Refrigerated	2524	1 Inch Nail	06/01/17	100.00 BX
5 San Francisco Branch		N	Non-Refrigerated	2532	2 Inch Nails	06/15/17	250.00 BX
5 San Francisco Branch		N	Non-Refrigerated	2541	2 1/2 Inch Nails	05/31/17	75.00 BX
5 San Francisco Branch		N	Non-Refrigerated	2559	3 Inch Nails	07/20/17	51.00 BX
Business Unit				5	San Francisco Branch		476.00

23.3 Defining a Subheading

You can define a subheading before you choose the type of detail you want in the report.

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
- 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 screen opens. Enter the field name and any other appropriate information.

To define a subheading

1. On Software Versions Repository, locate the report for which you want to define subheadings.
2. On Software Versions Repository, enter 10 in the Option field to access the Report Design Aid.
3. On Report Design Aid, choose Record Formats List (F10).
4. On the Record Formats List, enter HEADING2 on the first blank line in the Format Name field:

Figure 23–7 Record Formats List (Define a Subheading) report

[illegible]

5. Complete the following fields:

- Type
- Start/End Lines
- Fld Pfx (Field Prefix)

The system accommodates the placement of the fields on the report.

The following is an example of a report using a HEADING2 format. The system creates this report using a C0020 program type.

Figure 23–8 Inventory by Business Unit Report (HEADING2)

928400	J.D. Edwards & Company									
	Inventory by Business Unit Report									
	Page No. . . . 2									
	Date - . . . 12/02/17									
Bus	Description	It	Description	Item	Description	Ship	Quantity	On Hand	UM	
Unit				Number		Date				
5 San Francisco Branch		N	Non-Refrigerated	2524 1	Inch Nails	06/01/17	100.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2532 2	Inch Nails	06/15/17	250.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2541 2	1/2 Inch Nails	05/31/17	75.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2559 3	Inch Nails	07/26/17	51.00	RE		
Business Unit				5	San Francisco Branch		476.00			
							476.00			

Program type C0025 report prints the subheadings above the column titles as follows. You use the same steps to define this report as you use for the C0020 program type.

Figure 23–9 Inventory by Business Unit Report (Type C0025)

928400	J.D. Edwards & Company									
	Inventory by Business Unit Report									
	Page No. . . . 2									
	Date - . . . 12/02/17									
Business Unit	5	San Francisco Branch								
Bus	Description	It	Description	Item	Description	Ship	Quantity	On Hand	UM	
Unit				Number		Date				
5 San Francisco Branch		N	Non-Refrigerated	2524 1	Inch Nails	06/01/17	100.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2532 2	Inch Nails	06/15/17	250.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2541 2	1/2 Inch Nails	05/31/17	75.00	RE		
5 San Francisco Branch		N	Non-Refrigerated	2559 3	Inch Nails	07/26/17	51.00	RE		
Business Unit				5	San Francisco Branch		476.00			
							476.00			

The report program adds the grand totals automatically because it utilizes 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, see *Understand DREAM Writer* in the *JD Edwards World Technical Foundation Guide*.

23.4 Understanding DREAM Writer Considerations

When compiling your report, use the PRTF command to print a cover page. PRTS does not print a cover page when the system finishes compiling the report.

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

The title fields the system included on the cover page are in the following example:

Figure 23–10 Cover Page With Title Fields in DREAM Writer

Program ID . . . 9928400	The Organization's Name	Report Date. . . 12/02/17
Version. . . . 002	Inventory by Business Unit Report	Report Time. . . 9:35:50
	San Francisco Branch	
	Additional Line of Text if Required	

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.

Figure 23–11 Report Headings Using Field Names in DREAM Writer

928400	J.D. Edwards & Company Inventory by Business Unit Report	Page No. . . . 2 Date - . . . 12/02/17
--------	---	---

On the Additional Parameters screen in the DREAM Writer version, you must enter 2 in the Type Report Totaling field. This enables you to specify your total level fields on the Data Sequence screen.

Part VI

Additional Tools

This part contains these chapters:

- [Chapter 24, "Overview to Additional Tools,"](#)
- [Chapter 25, "Work with Quick Start CL Generator,"](#)
- [Chapter 26, "Work with the Quick Start Application Tool,"](#)
- [Chapter 27, "Work with Action Diagramming."](#)

Overview to Additional Tools

This chapter contains these topics:

- [Chapter 24.1, "Objectives,"](#)
- [Chapter 24.2, "About Additional Tools."](#)

24.1 Objectives

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

24.2 About Additional Tools

You can quickly create programs, forms, 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 the Action Diagramming feature.

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

This chapter contains the topic:

- [Section 25.1, "Working with Quick Start CL Generator."](#)

25.1 Working with 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 does not add the new CL program to a menu. You must perform that task manually.

Perform the following tasks:

- To create a program using the Quick Start CL Generator
- To compile a CL program

Navigation

From Computer Assisted Programming (CAP) (G93), choose Quick Start CL Generator

To create a program using the Quick Start CL Generator

1. On Quick Start CL Generator, complete the following fields and click Enter:
 - Description
 - Program Name
 - Screen or Report Name
 - Select a Program Type(1-4)
 - Master File
 - Source File Name
 - Source Library Name
 - Object Library Name
2. Perform one of the following:

- Click Exit (F3) to return to the menu.
- Click Enter to compile the program.

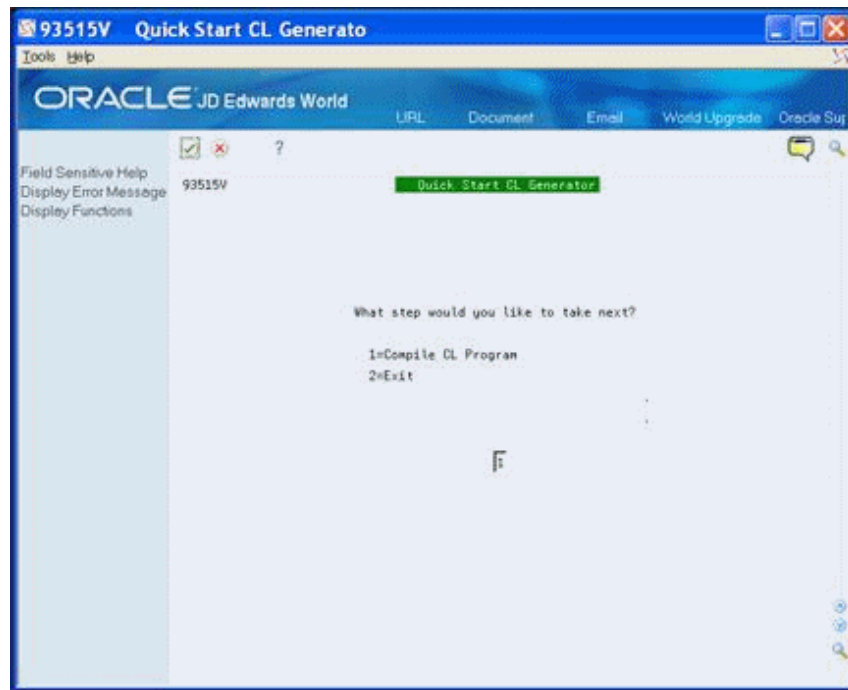
Figure 25–1 Quick Start CL Generator screen

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.
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.
Source File Name	The member ID of the file used by the program.
Source Library Name	File and library that contains the file source.
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.

To compile a CL program

On Quick Start CL Generator, enter 1 to choose Compile CL Program.

Figure 25–2 Quick Start CL Generator (Compile) screen



Work with the Quick Start Application Tool

This chapter contains these topics:

- [Section 26.1, "Quick Start Process,"](#)
- [Section 26.2, "Defining the Application,"](#)
- [Section 26.3, "Selecting Data Fields,"](#)
- [Section 26.4, "Browsing or Updating the Screens or Reports \(Optional\),"](#)
- [Section 26.5, "Compiling the Screens or Report \(Optional\),"](#)
- [Section 26.6, "Modifying Specifications \(Optional\),"](#)
- [Section 26.7, "Submitting the Program to Compile \(Optional\),"](#)
- [Section 26.8, "Updating the Data Dictionary and Glossary."](#)

The Quick Start Application Tool allows you to quickly create initial versions of programs, screens, and reports. After you create a new version, you can access the Screen or Report Design Aid or the Program Generator and make adjustments to the version.

The tool provides an easy way for you to create a prototype. This program allows you to:

- Create the program that you associate with the screen or report.
- Select fields dynamically from the master and detail files, as well as other database files.
- Compile your screen or report.
- Create specifications for the Program Generator.
- Create and compile your source code, optionally.
- Create a Control Language (CL) program to launch your new screen or report program.

The Quick Start Application Tool recognizes whether the program is a subfile. Additionally, this tool:

- Adds a hidden field to the screen for subfile maintenance.
- Sets the Entry Optional field to N for subfile maintenance.

Quick Start cannot:

- Define which VC0 fields to use as defaults in the version.
- Add the CL program to a menu.

- Add a Fold Area.

26.1 Quick Start Process

The Quick Start Application Tool is a set of steps that allow you to:

1. Define the Application
 - Define the type of program you want to create
 - Define the screen options
 - Define the report options
 - Define the files and libraries
 - Define the source file to use to create the application
2. Select Data Fields
 - Select the individual data fields to display on the screen or report using JD Edwards World Screen/Report Design Aid
 - Sequence the fields any way you choose.
3. Browse or update the screens or report you are creating (optional).
4. Compile screens or the report (optional).
5. Modify Specifications (optional).
 - Using the File Specifications, the Detailed Programming Facility, and the Help Instructions based on the program type you select.
 - Compile the program. Even if you compile the screen or report in a previous step, the system prompts you to do so again.
6. Compile the program (optional).
7. Update Data Dictionary and Glossary.

Note: As you create a version, you should continue through the steps and complete the entire process. The optional steps allow you to perform additional functions that relate to the process. If you exit the process and access the tool at a later time, the system enters all of the information on the Quick Start Application Tool screen from the previous version you created if you did not sign off the system.

Navigation

From Computer Assisted Programming (CAP) (G93), choose Quick Start Application Tool

26.2 Defining the Application

You create versions of programs, screens, and reports by defining the initial criteria.

The system enters all of the information on this screen from the previous version you created if you did not sign off the system.

To define the application

On Quick Start Application Tool, complete the following fields:

- Description
- Program Name
- Create Program(Y/N)?
- Screen or Report Name
- Select a Program Type(1-4)
- Action Code
- Selection Option
- Report Detail Subheadings
- Report Total Subheadings
- Report Totals
- Master File
- Library Name
- Detail File(optional)
- Library Name
- Source File Name
- Source Library Name
- Object Library Name

Figure 26–1 Quick Start Application Tool screen

Field	Explanation
Description	Use this field to enter a short one-line description of the program you are creating.

Field	Explanation
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.
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 file name associated with the program. This field is only required for program type 2.
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
Action Code	Enter Y if you are creating a screen with an Action Code field. Enter N if you are not creating a screen with an Action Code 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 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.
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	The member ID of the file used by the program.
Source Library Name	File and library that contains the file source.
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.

26.3 Selecting Data Fields

The system displays key fields from each data file. You can select, deselect or resequence the fields that you want to use in your program.

The fields from the primary data file display first in the list, followed by the fields from the secondary data file, if you entered one.

To select data fields

- On Quick Start Field Selection, perform any of the following:
 - To select a field, enter 1 in the Option field to the left of the field name and click Enter.
 - To specify heading or subfile fields enter 1 or 2, respectively, in the column to the right of the selection and sequencing column.

This field only displays if the program you are creating is a transaction processor.

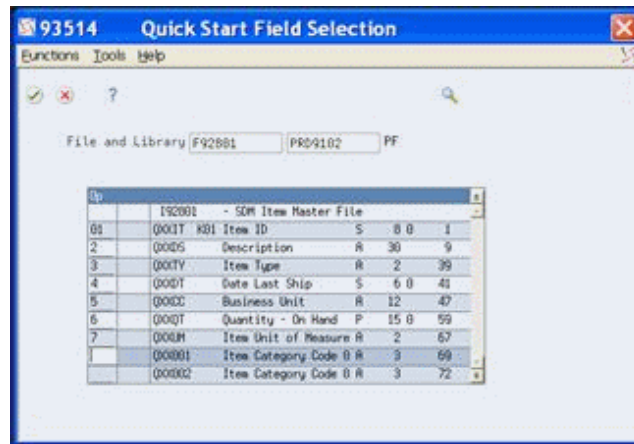
- To sequence a field, enter the sequence number in the Option field to the left of the fields you want to use in your program and click Enter.
- To add fields, enter the names of additional fields on the screen.

2. Press (F3) to continue.

3. The system prompts you perform one of the following:

- Exit (F3) to exit the program.
- Replace (F6) to continue to the next step in the process.

Figure 26–2 Quick Start Field Selection screen



26.4 Browsing or Updating the Screens or Reports (Optional)

At this point in the process, you can view or update the screens or reports you are create using the browse or update mode.

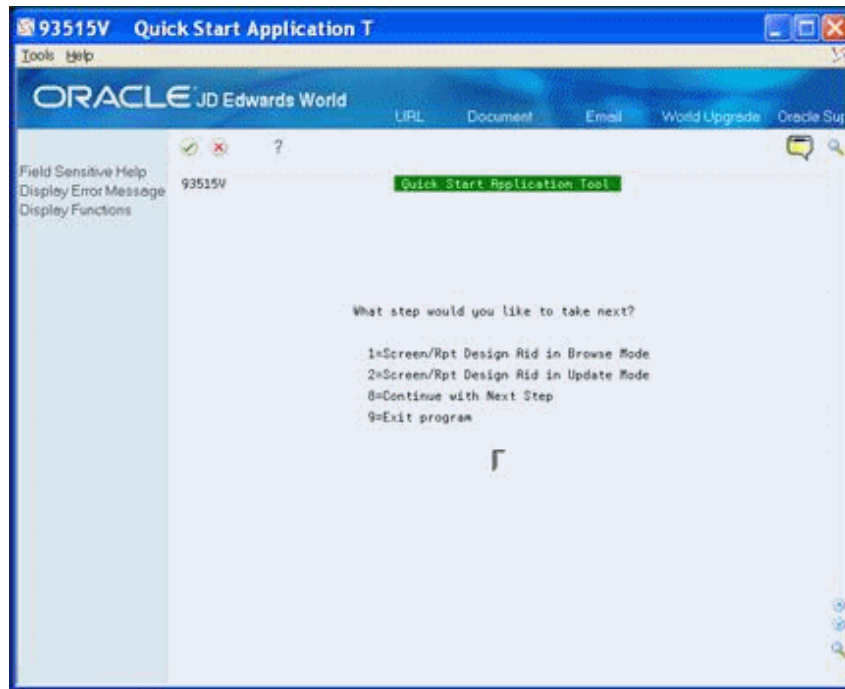
To browse or update screens or reports

On Quick Start Application Tool, enter 1 to browse or 2 to update.

To continue the steps to create a version, enter 8.

To exit the program, enter 9 and return to the Computer Assisted Programming (CAP) menu.

Figure 26–3 Quick Start Application Tool (Browse or Update) screen



26.5 Compiling the Screens or Report (Optional)

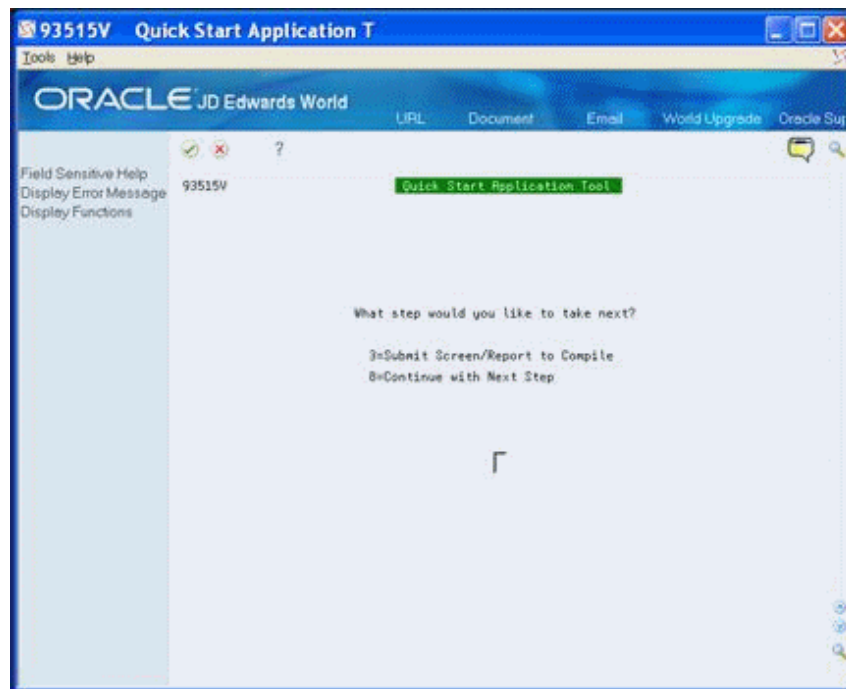
At this point in the process, you can compile the screens or report. Before you compile the program, you must complete this step.

To compile the screens or report

On Quick Start Application Tool, enter 3 to compile the screens or reports.

The system retrieves the object library for the compile from the CASE Profiles.

To continue the steps to create a version, enter 8.

Figure 26–4 Quick Start Application Tool (Compile) screen

26.6 Modifying Specifications (Optional)

At this point in the process, you can access the Program Generator Specifications screen to modify specifications.

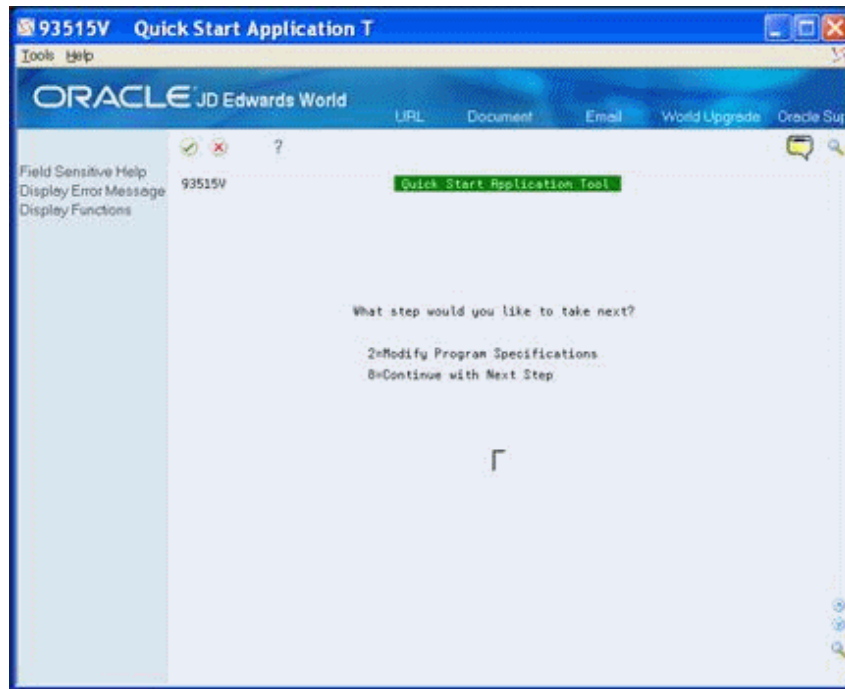
To modify specifications

On Quick Start Application Tool, enter 2 to modify program specifications.

The Program Generator Specifications screen displays.

To continue the steps to create a version, enter 8.

Figure 26–5 Quick Start Application Tool (Modify) screen



26.7 Submitting the Program to Compile (Optional)

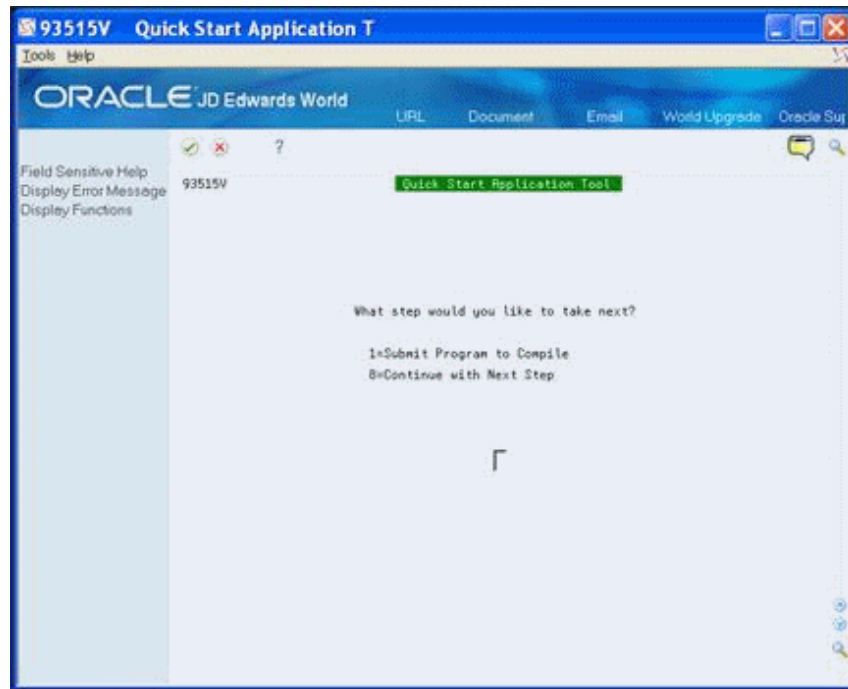
Do not submit the program to compile until the screens or report successfully compile.

To submit the program to compile

On Quick Start Application Tool, enter 1 to compile the screens or reports.

To continue the steps to create a version, enter 8.

Figure 26–6 Quick Start Application Tool (Submit to Compile) screen



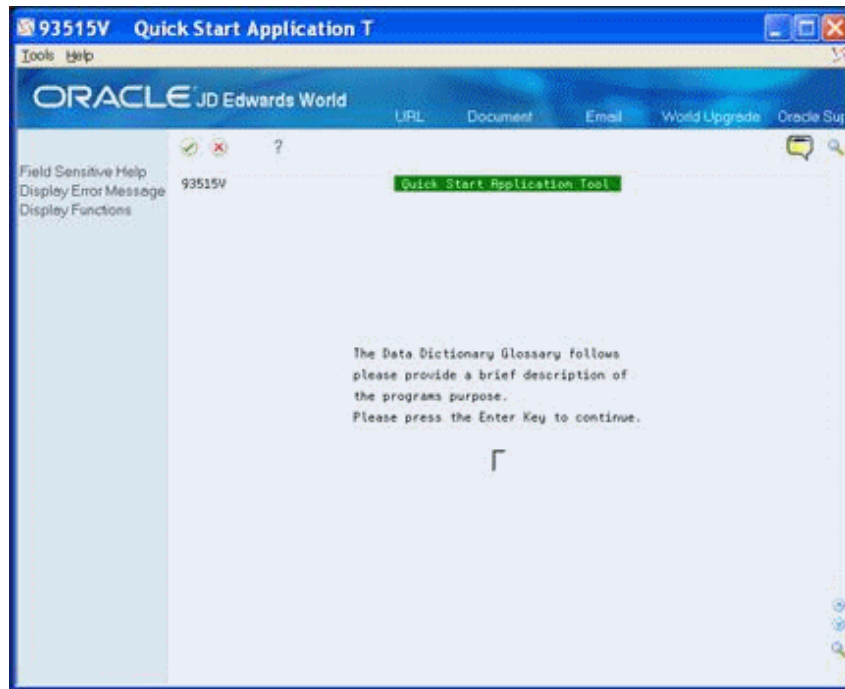
26.8 Updating the Data Dictionary and Glossary

You must enter the purpose of the program on the Data Item Glossary Revisions screen.

To update the data dictionary and glossary

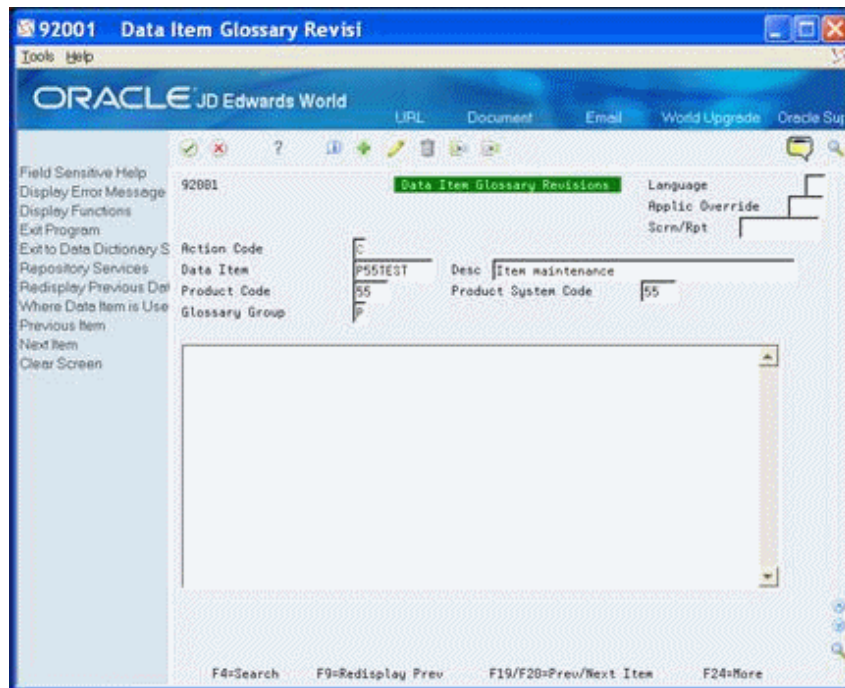
1. On Quick Start Application Tool, click Enter to continue.

Figure 26–7 Quick Start Application Tool (Update Data Dictionary) screen



2. On Data Item Glossary Revisions, enter the description of the program's purpose that displays in the online help instructions.

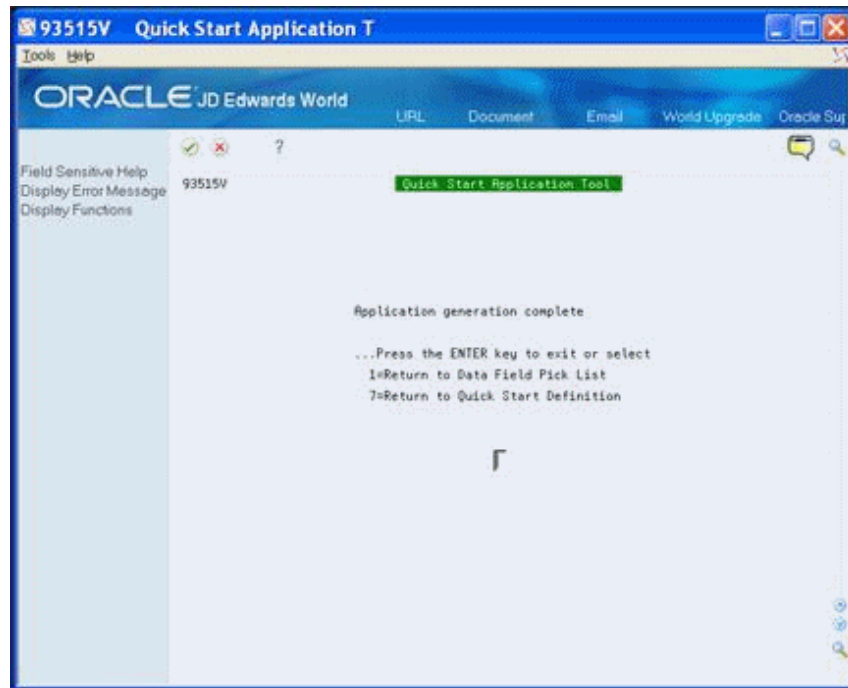
Figure 26–8 Data Item Glossary Revisions screen



3. Click Add to add the program purpose statement.
4. Click Exit (F3).
5. On Quick Start Application Tool, perform one of the following:

- Click Enter to exit the program and return to the menu.
- Enter one of the following:
 - 1 to Return to the Data Field Pick List
 - 7 to Return to Quick Start Definition

Figure 26–9 Quick Start Application Tool (Exit) screen



Work with Action Diagramming

This chapter contains these topics:

- [Section 27.1, "Building an Action Diagram,"](#)
- [Section 27.2, "Viewing an Action Diagram,"](#)
- [Section 27.3, "Accessing the Logic Translation Feature."](#)

The Action Diagramming functionality allows you to produce a diagram which illustrates the different groupings of logic and the interrelationships of code within a program. The system generates the diagrams from the program source code. They provide easy access to more detailed information about the files, fields and programs in the code.

27.1 Building an Action Diagram

The Build Action Diagram program allows you to build the necessary cross reference items to produce the action diagram. Using DREAM Writer as the initial screen to the batch job, you can specify the programs for which you want to build an action diagram.

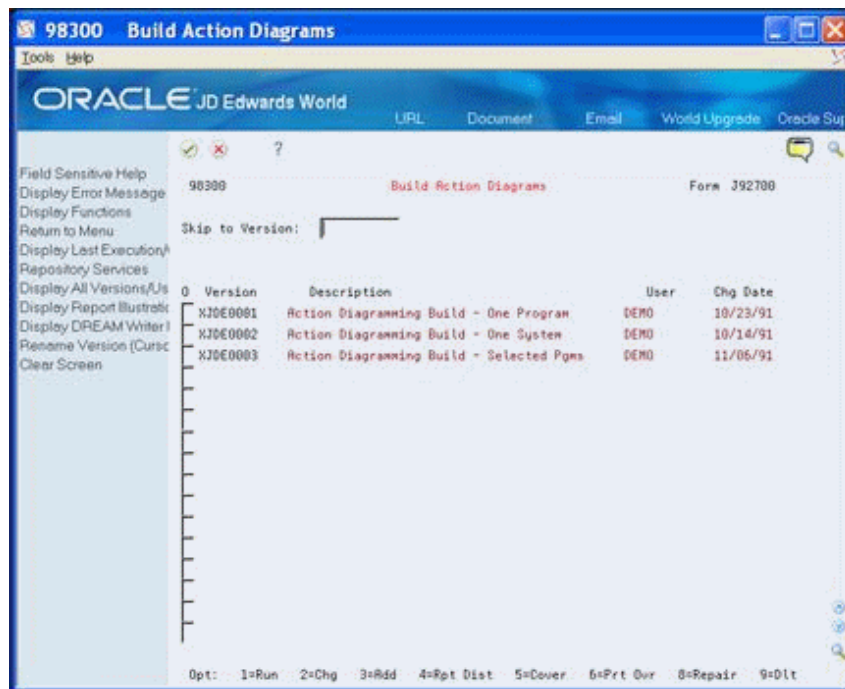
JD Edwards World includes sample Action Diagrams with the software but you must build the Action Diagram for all other programs. This is not an automatic function.

To build an action diagram

Navigation

From Action Diagramming (G9363), choose Build Action Diagrams

On Build Action Diagram, choose a version.

Figure 27-1 Build Action Diagrams screen

27.2 Viewing an Action Diagram

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

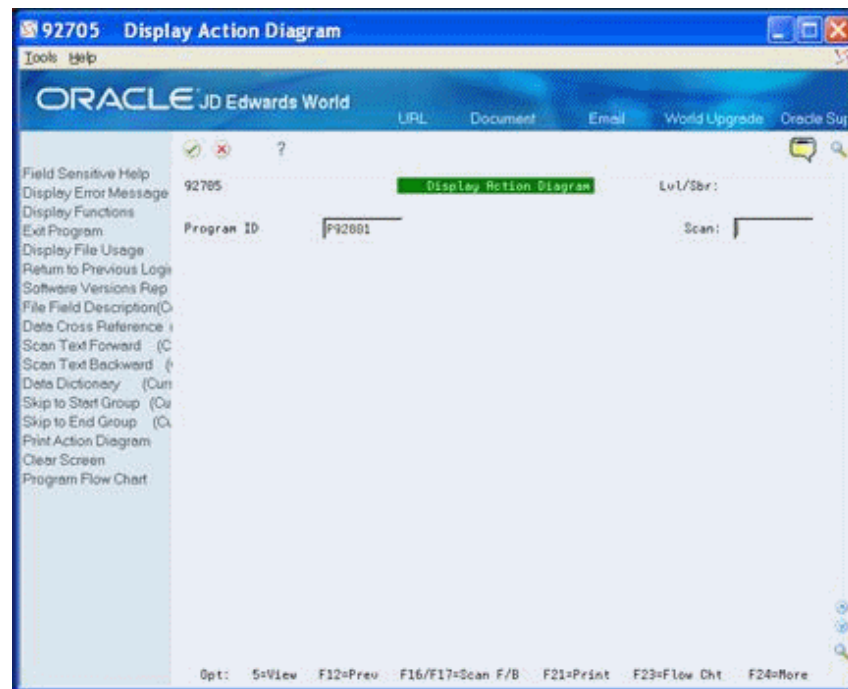
To view an action diagram

Navigation

From Action Diagramming (G9363), choose Display Action Diagram

On Display Action Diagram, enter a program ID in the Program ID field to view an action diagram.

For example, enter P92801.

Figure 27-2 Display Action Diagram screen

The logic groups for the program display.

Group	Description
Lvl/Sbr	Specifies the logic level and subroutine.
Program ID	The program name for the action diagram.
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.

Symbol	Description
===>	Signals the beginning or ending of a loop.
--->	Signals an IF or WHEN statement or their associated end statement.
Blank	Labels are in reverse image.

27.2.1 Function Exits

Display File Usage (F10)

Choose Display File Usage (F10) to view the files in the file specifications of the program.

Return to Previous Logic Level (F12)

Choose Return to Previous Logic Level (F12) to return to the logic level immediately prior to the one that currently displays.

Scan Text Forward (F16)

Choose Scan Text Forward (F16) to enter a value in the Scan field and then scan forward through the code to locate the value.

Scan Text Backward (F17)

Choose Scan Text Backward (F17) to enter a value in the Scan field and then scan backward through the code to locate the value.

Skip to Start Group (F19)

Choose Skip to Start Group (F19) to skip to the beginning (start) of a section of code. The user places the cursor within the section of code and then chooses Skip to Start Group to go to the beginning of that section of code.

Skip to End Group (F20)

Choose Skip to End Group (F20) to skip to the end of a section of code. The user places the cursor within the section of code and then chooses Skip to End Group to go to the end of that section of code.

Print Action Diagram (F21)

Choose Print Action Diagram (F21) to obtain a printout of the action diagram.

Program Flowchart (F23)

Choose Program Flowchart (F23) to view and print, or view, or print a flowchart which illustrates the interaction of files and processes that relate to a single program. You can continue to view lower levels of detail as well.

27.2.2 Cursor Sensitive Function Exits

To access information that relates to fields, files, and programs appearing in the program code, you can use cursor sensitive function exits to access this information by placing the cursor at the beginning of the field, file, or program.

Software Versions Repository (F13)

Choose Software Versions Repository (F13) to access the Software Versions Repository.

File Field Description (F14)

Choose File Field Description (F14) to display the File Field Description screen.

Data Cross Reference (F15)

Choose Data Cross Reference (F15) to access the cross reference program.

Data Dictionary (F18)

Choose Data Dictionary (F18) to access the Data Dictionary program.

The following chart indicates which function exits access relevant information for the different elements.

Element	Function Exit	Description
Fields	Data Cross Reference (F15)	Displays all the programs that use the data item.

Element	Function Exit	Description
Files	Data Dictionary (F18)	Displays the Data Dictionary definition for the data item.
	Display File Usage (F10)	Displays the files within the program.
	Software Versions Repository (F13)	Displays the Software Versions Repository record for the file.
	File Field Description (F14)	Displays the File Field Descriptions for the file.
	Data Cross Reference (F15)	Displays all the programs that use the file.
Programs	Software Versions Repository (F13)	Displays the Software Versions Repository record for the program.
	Data Cross Reference (F15)	Displays all the programs that launch the program.

27.2.3 Option Field Values

View (5)

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

27.3 Accessing the Logic Translation Feature

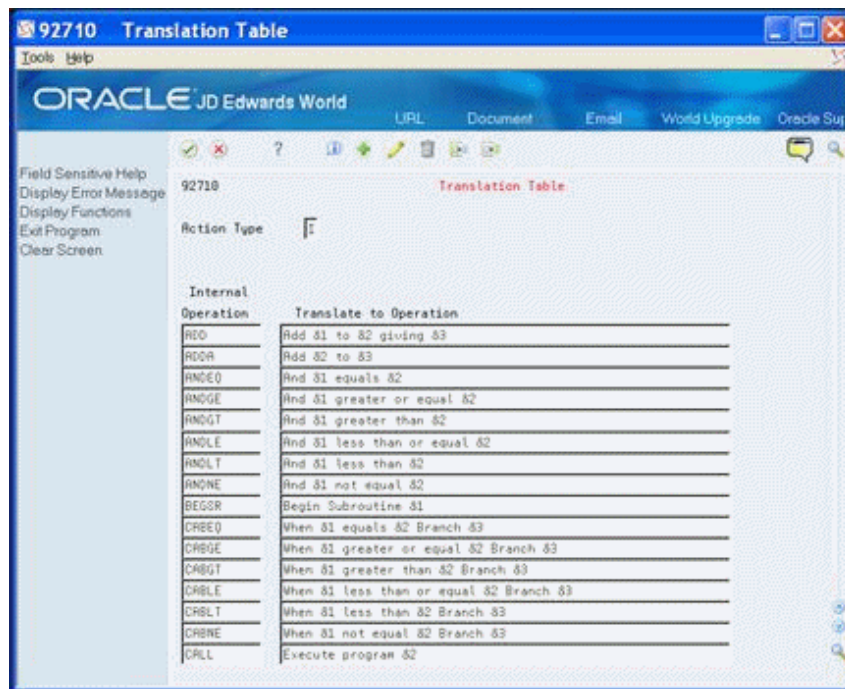
The Logic Translation feature allows you to view how the Action Diagram feature translates the RPG code of a program into an Action Diagram.

To access the logic translation feature

Navigation

From Action Diagramming (G9363), choose Translation Table

The system displays the RPG operation in the first column and how it translates that operation within an action diagram in the second column.

Figure 27–3 Translation Table screen

Part VII

Source Code Inventory and Database

This part contains these chapters:

- [Chapter 28, "Overview to Source Code Inventory and Database,"](#)
- [Chapter 29, "Understand Source Sequence,"](#)
- [Chapter 30, "Working with Program Types,"](#)
- [Chapter 31, "Work with Logic Modules,"](#)
- [Chapter 32, "Understand Directives,"](#)
- [Chapter 33, "Work with the Question and Answer System,"](#)
- [Chapter 34, "Create User Defined PDL."](#)

Overview to Source Code Inventory and Database

This chapter contains these topics:

- [Section 28.1, "Objectives,"](#)
- [Section 28.2, "About the Source Code Inventory and Database."](#)

28.1 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

28.2 About 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

Understand Source Sequence

This chapter contains these topics:

- [Section 29.1, "Source Serial Numbers,"](#)
- [Section 29.2, "Source Sequence Line Structure,"](#)
- [Section 29.3, "Structure of the Serial Number."](#)

When you use the program generator, it is important that you understand how the system manages the source code in the program. The topics in this chapter include the key elements that the system assigns.

29.1 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, and then rennumbers the entire source.







29.2 Source Sequence Line Structure

The source sequence line structure includes six elements:

Element	Description
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 you add or change the code within the source sequence line.

The following illustrates the parts of the source sequence line.

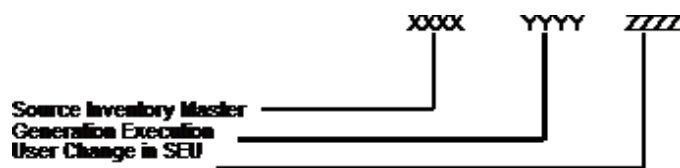
Figure 29–1 Parts of the Source Sequence Line

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.

29.3 Structure of the Serial Number

Figure 29–2 Structure of the Serial Number



29.3.1 Source Inventory Master File (F93001) - XXXX

- Assigns numbers to the first sequence of the serial number.
- Increments by 10 to allow you to insert lines as the Program Generator Source Inventory Master file changes.
- Allows a maximum of 9999 lines.

29.3.2 Generation Execution - YYYY

- Assigns numbers when the system generates the program.
- Represents lines that are part of a detail logic module.
- Increments by 10 to allow you to insert lines.
- Allows a maximum of 9999 lines.

29.3.3 User Change in SEU - ZZZZ

- Represent lines of code that the user inserts via SEU.
- Allows a maximum of 9999 lines.

Working with Program Types

This chapter contains these topics:

- [Section 30.1, "Reviewing Abbreviations for Program Types,"](#)
- [Section 30.2, "Reviewing Program Types Index,"](#)
- [Section 30.3, "Reviewing Program Types Cross Reference,"](#)
- [Section 30.4, "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 within the program type.

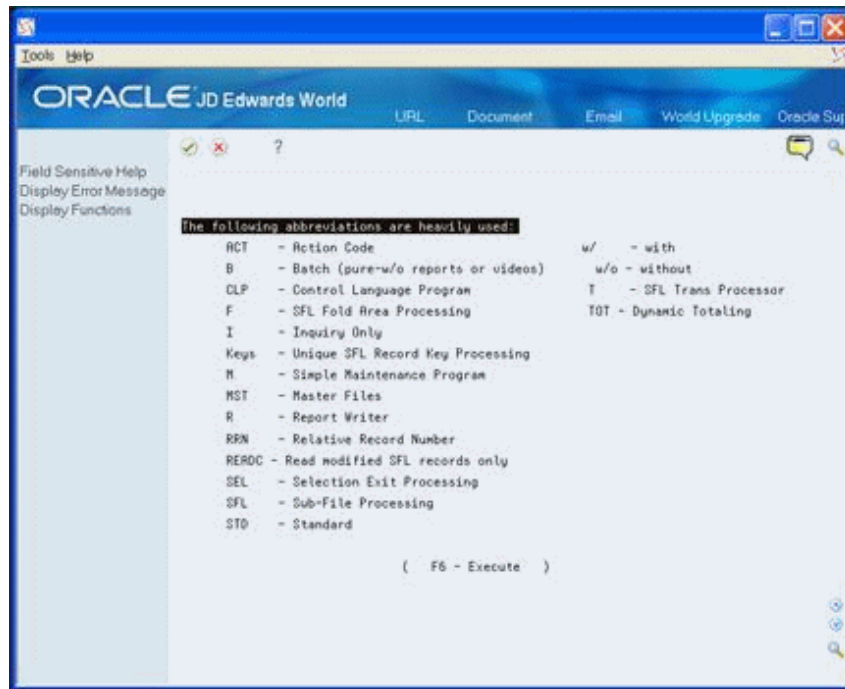
JD Edwards World provides you with program types for the most common programs. You can create your own program types for your organization's needs.

30.1 Reviewing Abbreviations for Program Types

You can use the Index to review abbreviations for data that displays on the Create Modify screen.

Navigation

From Model Program Design Menu (G9361), under PROGRAM TYPES, choose Index

Figure 30–1 Index of Abbreviations for Program Types screen

30.2 Reviewing Program Types Index

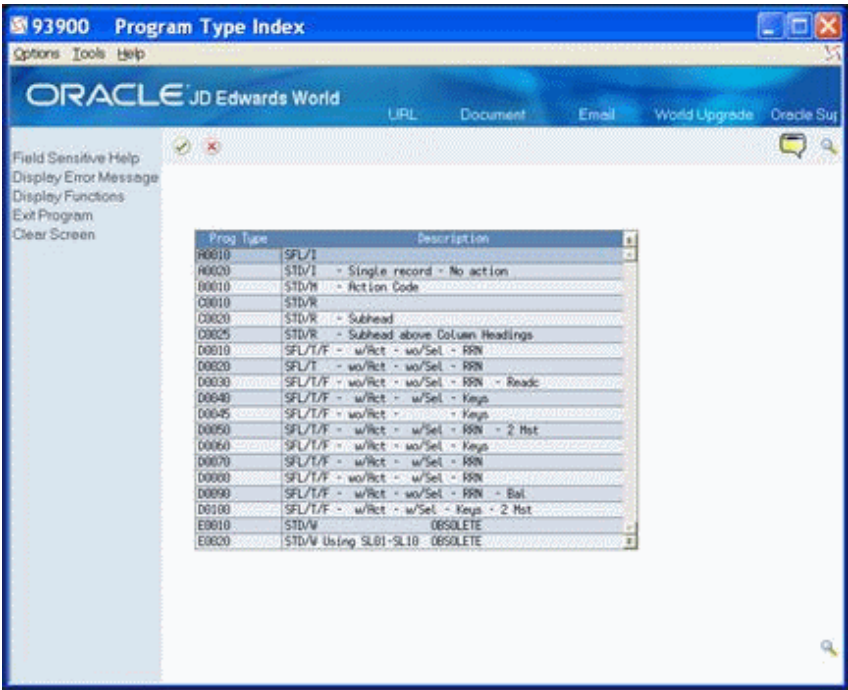
You can use the Program Type Index program (P93900) to locate program types. You can access this from the Program Types Cross Reference screen, Index screen, as well as the Create/Modify screen.

30.2.1 Available Options

You can right click on any program type and choose one of the following options:

- Display Full Logic - Access the Data Item Glossary Revisions screen
- Print Logic Class Source
 - Prints the generic source of the shell program without any of the specifics (detail logic modules).
 - Use this if you are creating your own program types and you want to review them.
- Display Logic Class Source
 - Displays the generic source of the shell program without any of the specifics (detail logic modules).
 - Use this if you are creating your own program types and you want to review them.
- Return Selected Program Type - Retrieves the program type when the system accesses it from another program.
- Display Logic Cross Reference - Displays the programs you create using this logic type.
- Revise Logic Class - Displays a bill of materials list for the program type.

Figure 30-2 Program Type Index screen

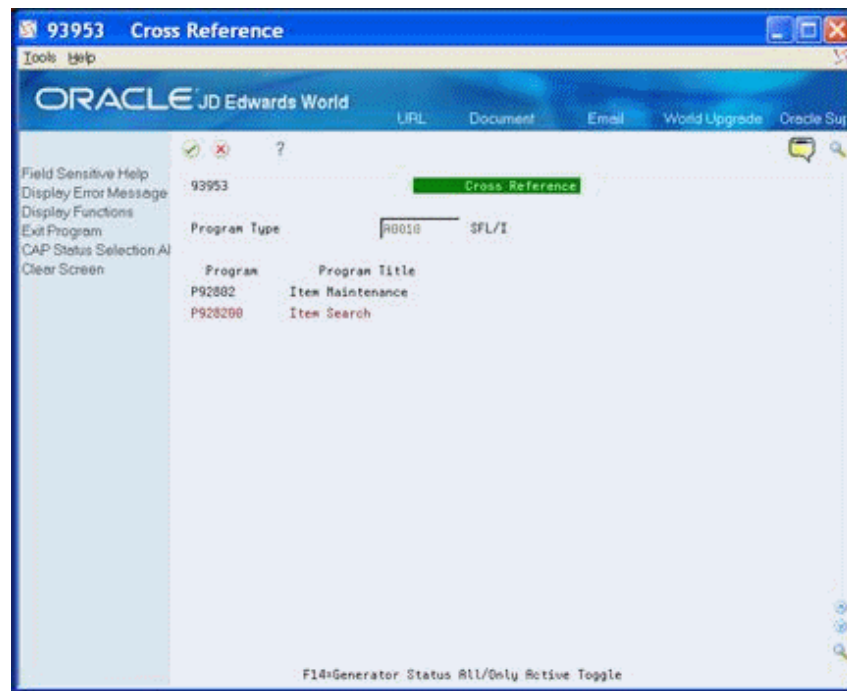


30.3 Reviewing Program Types Cross Reference

The Program Types Cross Reference screen allows you to review additional information about program types.

Navigation

From Model Program Design Menu (G9361), under PROGRAM TYPES, choose Cross Reference

Figure 30–3 Cross Reference screen**Clone Status All/Only Active Toggle (F14)**

You can choose CAP Status Selection All/Only Active (F14) to toggle back and forth between viewing all programs using the program type or only the programs with a CAP status of Y.

30.4 Creating or Modifying Program Types

As you create or modify program types, you should be aware of the following:

30.4.1 Program Type

It is the list of the segments of code the system requires to build this type of program. You might consider this a bill of materials list.

30.4.2 Primary Module

This is the main sections of code that the system uses to create the first level of program source.

30.4.3 Glossary K

The system uses these to document logic modules within a program type. When the system generates a program, it validates the field against the Data Dictionary, and adds the glossary for the key as documentation for the logic module.

To create or modify program types

Ensure you are of the alpha order requirement when creating new program types.

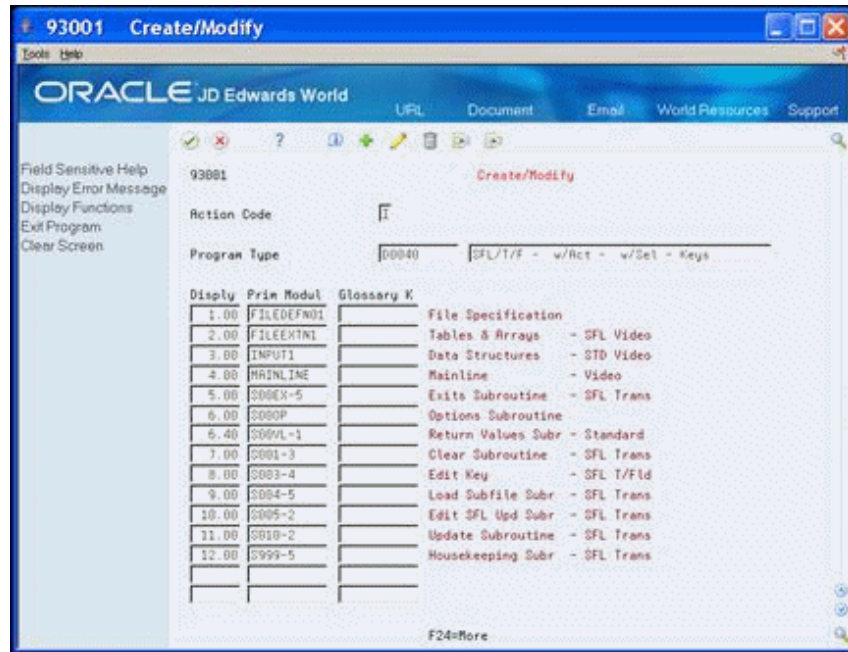
The following is an example using program type D0040.

Navigation

From Model Program Design Menu (G9361), under PROGRAM TYPES, choose Create/Modify

On Create/Modify, locate an existing program type.

Figure 30–4 Create/Modify screen



Work with Logic Modules

This chapter contains these topics:

- [Section 31.1, "Primary Logic Modules,"](#)
- [Section 31.2, "Detail Logic Modules,"](#)
- [Section 31.3, "Generation Options,"](#)
- [Section 31.4, "Viewing the Logic Module Index,"](#)
- [Section 31.5, "Viewing Logic Module Cross Reference,"](#)
- [Section 31.6, "Viewing Logic Module Op Codes,"](#)
- [Section 31.7, "Maintaining the Logic Module File,"](#)
- [Section 31.8, "Creating or Modifying Logic Modules,"](#)
- [Section 31.9, "Creating or Modifying Formula Library Entry,"](#)
- [Section 31.10, "Copying or Moving Program Specifications,"](#)
- [Section 31.11, "Printing Program Generator Specifications,"](#)
- [Section 31.12, "Reviewing Source Modifications,"](#)
- [Section 31.13, "Using Program Generator Updates,"](#)
- [Section 31.14, "Using CASE Specifications Inquiry."](#)

There are two types of logic modules:

- Primary
- Detail

31.1 Primary Logic Modules

Primary logic modules include:

- Main segments of code in the definition of a program type.
- Full sections of a program or subroutines within the program, normally.
- Functional directives to the generation program.

Each primary logic module includes code with a five character directive code. See Columns 1 through 5 in the Source Code Inventory Master file (F93001).

The JD Edwards World CASE software provides approximately 100 different primary logic modules. This includes many variations on mainline logic, field initialization,

update logic, housekeeping, and so forth. Use the Index for logic modules to become familiar with the various types of primary logic modules.

Primary logic modules contain the following:

- Program identification specifications
- Extension specifications
- Data structures
- Mainline calculations
- Default logic from data dictionary
- Subroutine calculations
- Update subroutine
- Housekeeping subroutine

31.2 Detail Logic Modules

Detail logic modules direct the final integration of the database, screen, or report specifications into the primary logic modules that make up the final program type.

Detail logic modules are usually functional or data field-related segments of code. Functional directives reference the detail logic modules which contain substitution directives to the generation program. A prefix of X indicates the system does not use the detail logic module in conjunction with a conditional directive. A prefix of Z indicates the system uses the detail logic module in conjunction with a conditional directive. See [Chapter 32, "Understand Directives"](#) for more information about directives.

31.3 Generation Options

Following are additional programs you can use on the Model Program Design Menu.

31.3.1 Help Instructions Edit/Build

You use this to access the Software Versions Repository to rebuild the Helps for a single program.

31.3.2 All Help Instructions

You use this to submit a job to regenerate the helps for all programs.

31.3.3 Global Program Regeneration

You use this to regenerate all programs that have a CAP Status of Y.

Caution: Use caution when you use this program.

Working with logic modules includes the following tasks:

- View the Logic Module Index
- View the Logic Module Cross Reference
- View Logic Module Op Codes

- Maintain the Logic Module File
- Create or modify Logic Modules
- Create or modify Formula Library Entry
- Copy or move program specifications
- Print Program Generator specifications
- Review source modifications
- Use Program Generator updates
- Use CASE specifications inquiry

31.4 Viewing the Logic Module Index

The system allows multiple logic modules for each subroutine. Depending on the type of program in which you use the subroutine, the same subroutine can appear differently.

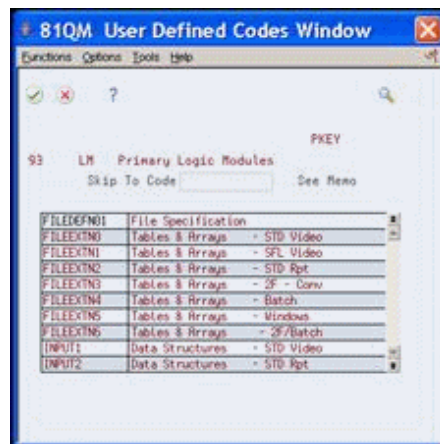
You can review the logic modules on the User Defined Codes window.

To view the logic module index

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Index

Figure 31-1 User Defined Codes Window screen



31.5 Viewing Logic Module Cross Reference

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

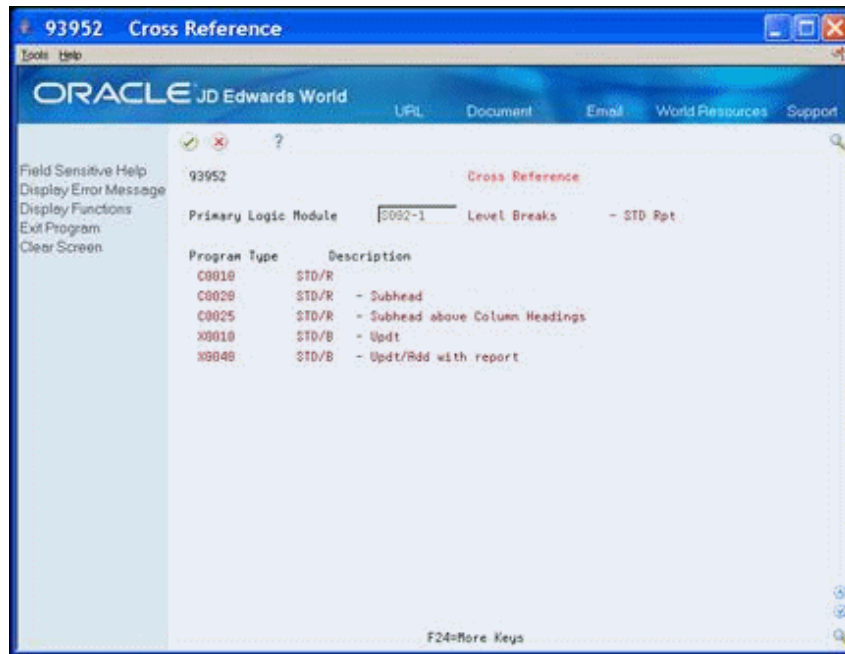
To view the logic module cross reference

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Cross Reference

Enter a primary logic module name.

Figure 31–2 Cross Reference (View Logic) screen



31.6 Viewing Logic Module Op Codes

On the Op Codes screen, the:

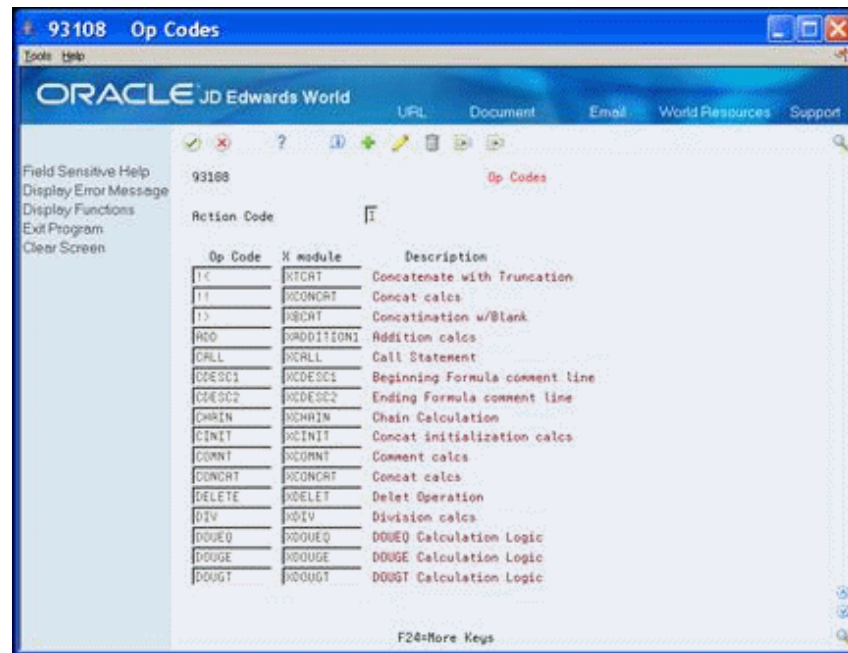
- Left column lists the PDL op codes.
- Right column displays the x-module that the system launches to generate the source code.

If PDL does not generate source code, the Operation Code to Logic Module X-Ref file (F93108) might have been accidentally cleared.

To view the logic module op codes

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Op Codes

Figure 31–3 Op Codes screen

31.7 Maintaining the Logic Module File

The following programs do not appear on a menu and you must access these programs manually.

Caution: Use extreme caution when using these programs.

31.7.1 Resequence Logic Module

Use this program when you need to add several lines to a logic module and resequence the line numbers. If you add or change lines in a logic module, you must manually change or add the serial numbers for the logic module or run this program. The Resequence Master Source program (93998) launches a program to resequence an existing logic module.

Normally, you create and incorporate a new logic module into a new program type. You use the new program type and delete the old program type when there are no longer programs with that program type with a CAP status of Y.

CALL P93998 PARM (logic module name).

31.7.2 Remove Logic Module

You use this program when you no longer use a logic module and want to reduce the amount of source code in the F93001 file. The Remove Logic Module program (P93999) removes lines from F93001 and launches a program to remove an existing logic module.

You must ensure that there are no programs with a CAP status of Y that use a program type with this logic module.

CALL P93999 PARM(logic module name).

31.8 Creating or Modifying Logic Modules

The Create/Modify screen allows you to review only the logic module you want, otherwise all 12,000 lines of code display because the F93001 is a single member file.

You can choose Field Sensitive Help to access a list of logic modules, UDC 93/LM.

When the system accesses the code, it performs three steps:

- Creates a work file in QTEMP/F93001WRK.
- Adds a member to F93001WRK.
- Clears the member in F93001WRK.

You can exit the code without saving your changes.

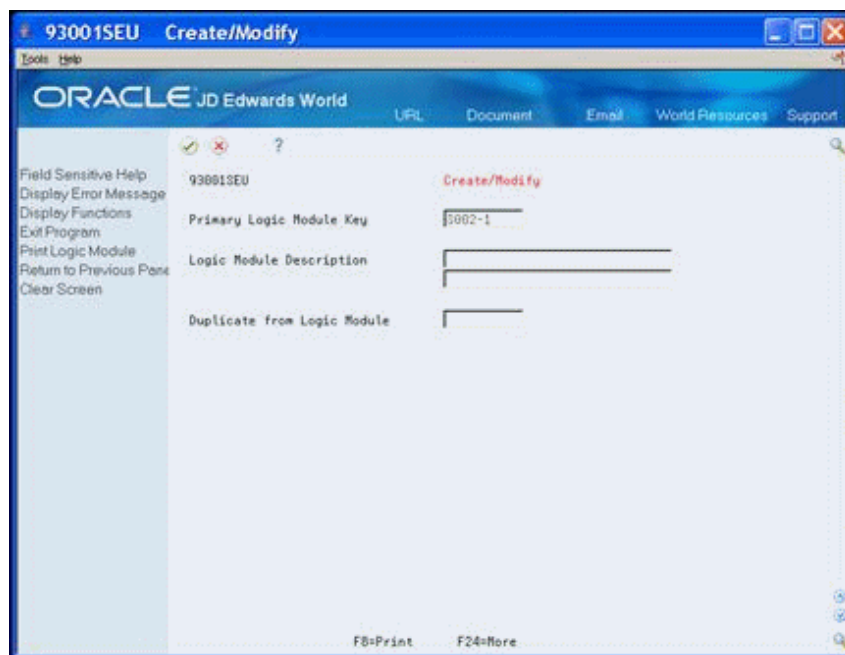
To create or modify logic modules

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Create/Modify

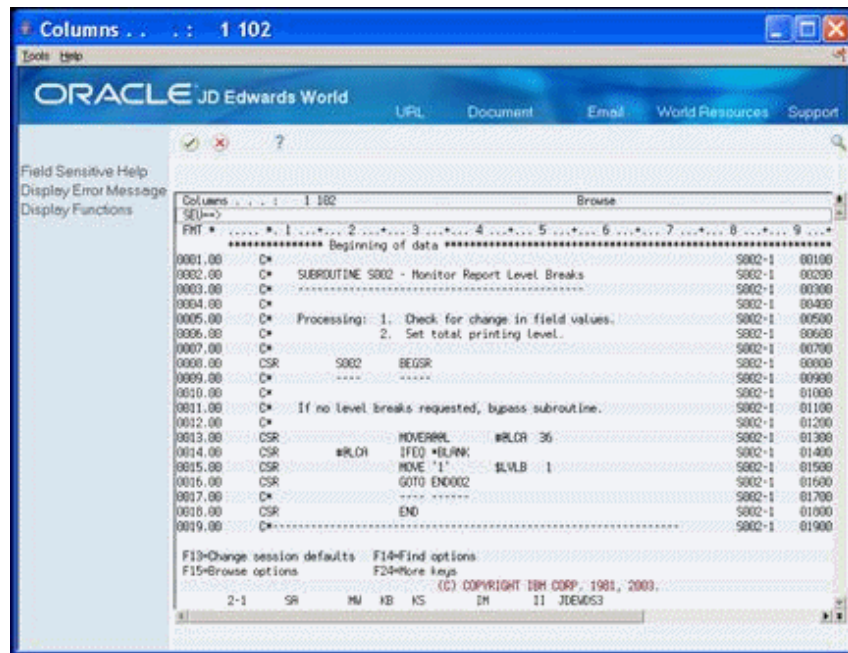
1. Enter a logic module name.

Figure 31–4 Create/Modify (Logic Modules) screen



2. Create or change the appropriate lines of code

Figure 31–5 Lines of Code screen



31.9 Creating or Modifying Formula Library Entry

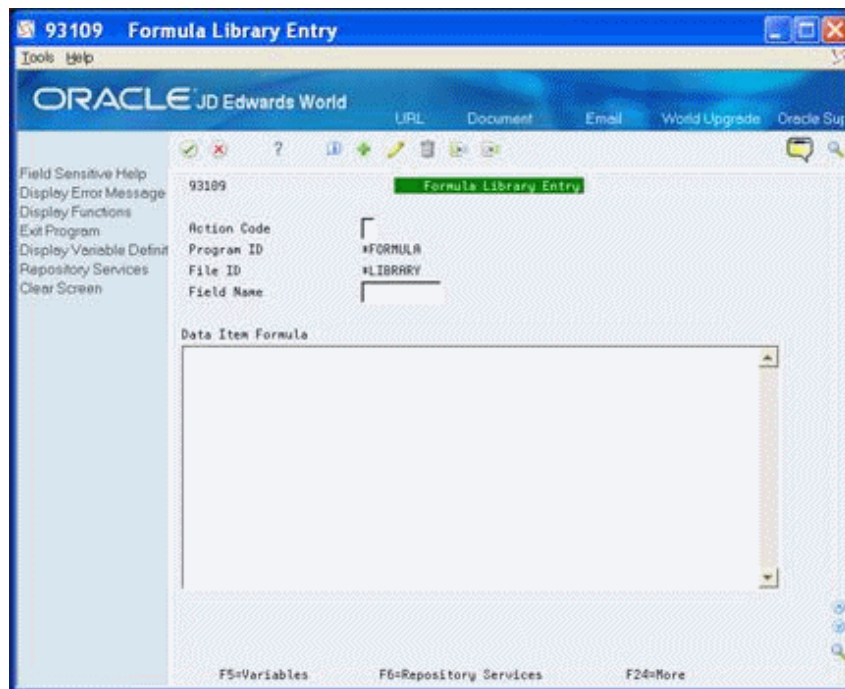
When you choose this menu selection, the system preloads the screen with the keys for entering a formula.

Alternatively, you also access this screen through the Detailed Programming Facility to enter PDL.

To create or modify the formula library entry

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Formula Library Entry

Figure 31–6 Formula Library Entry screen

31.10 Copying or Moving Program Specifications

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

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

Alternatively, you can enter 3 in the Option field on Software Versions Repository to copy Program Generator specifications within a library.

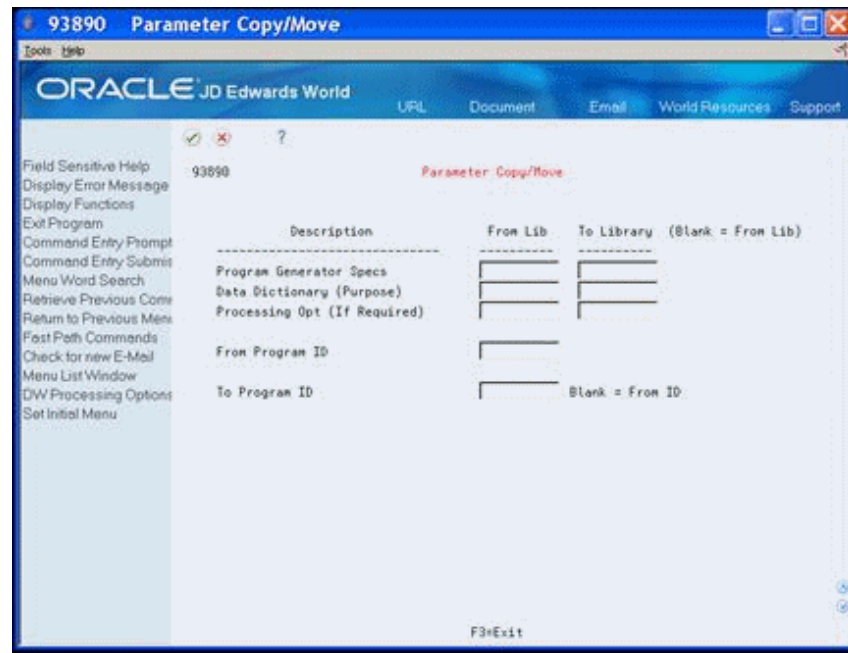
To copy or move program specifications

Navigation

From Model Program Design Menu (G9361), under OTHER TOOLS, choose Parameter Copy/Move

On Parameter Copy/Move complete the following fields:

- Program Generator Specs
- Data Dictionary
- Processing Option
- From Program ID
- To Program ID

Figure 31–7 Parameter Copy/Move screen

31.11 Printing Program Generator Specifications

Use the Print Program Specifications program to print the program specifications. You must use a logical file.

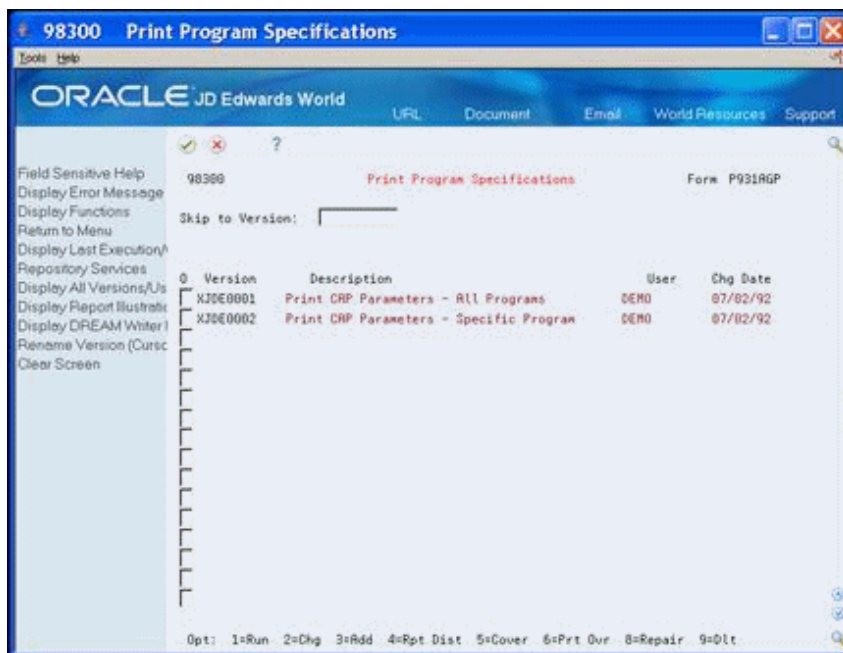
If the print job ends abnormally, review the Additional Parameters screen of the DREAM Writer and ensure that the File Output Type field is set to a Logical File and not Open Query.

To print program generator specifications

Navigation

From Model Program Design Menu (G9361), under OTHER TOOLS, choose Print Program Specifications

Copy the appropriate version and change it to print the version of the specifications you want.

Figure 31–8 Print Program Specifications screen

31.12 Reviewing Source Modifications

The Review Source Modifications program displays the source code that a user adds manually through the Source Entry Utility. You view the Pxxxx member in the Additional Help/Modifications Master file (F93002).

The lines of code are the result of the MPxxxx job that runs and compares the before image of the source code with the source code after the user makes changes and stores the code in the Pxxxx member in the F93002.

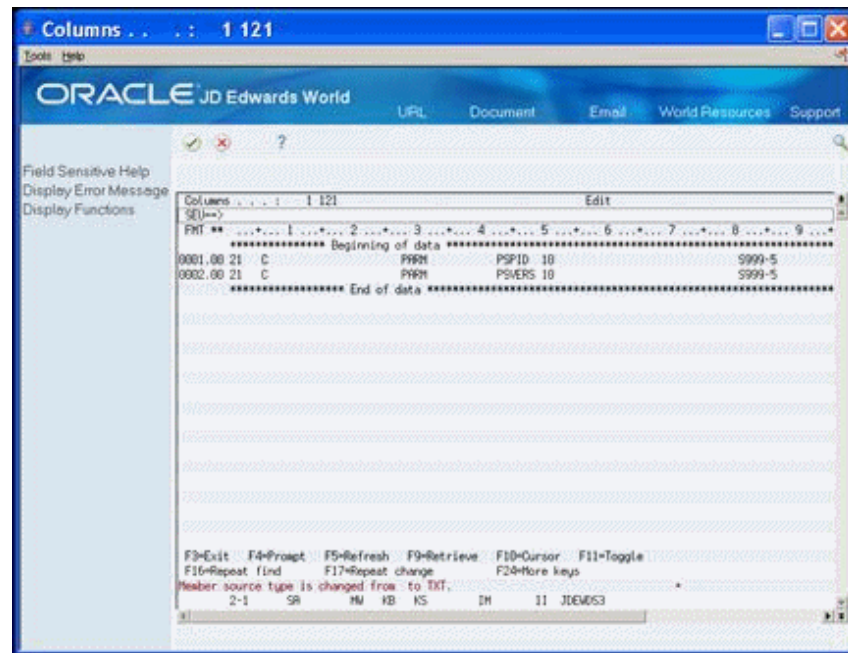
To review source modifications

Navigation

From Model Program Design Menu (G9361), under OTHER TOOLS, choose Review Source Modifications

Alternatively, you can enter 30 in the Option field on Software Versions Repository to access Review Source Modifications screen.

1. On the Review Source Modifications screen, locate the program.
2. Enter 30 in the Option field to view source code modifications.

Figure 31–9 Review Source Code Modifications screen

31.13 Using Program Generator Updates

The both of the Generator Updates merge JD Edwards World updates for the Program Generator.

The system uses these programs during a PTF install.

To use program generator updates

Navigation

From Model Program Design Menu (G9361), choose Generator Updates From Generator Updates (G9366), choose the appropriate Compare/Update

31.14 Using CASE Specifications Inquiry

The CASE Specifications Inquiry allows you to view the programs you design using the JD Edwards World CASE Tools. You can modify and delete CASE Specifications using this utility as well as access the source code in the Software Versions Repository.

To use CASE specifications inquiry

Navigation

From Model Program Design Menu (G9361), under OTHER TOOLS, choose Case Specifications Inquiry

1. On CASE Specifications Inquiry, complete any of the following fields:
 - Program ID
 - System Code
 - CAP Status

- Program Type

The system displays the records that meet your search criteria.

2. Complete the following field:

- Option

Figure 31-10 CASE Specifications Inquiry screen

93130 CASE Specifications Inqui

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
More Details
Clear Screen

93130 CASE Specifications Inquiry

Program ID: P92801 to
System Code CRP Status Program Type

Program ID	Program Title	Prod Cod	Program Type	S t SVR Status
P92801	Item Maintenance	92	00040	Y
P928011	Item Master Information	92	00010	Y
P92802	Item Maintenance	92	00010	Y
P928200	Item Search	92	00010	Y
P928400	Inventory by Business Unit	92	00010	Y
P928401	Inventory by Cost Center w/Sub	92	00020	N
P92910	Copy ROW Files into Production	93	00010	N
P93KBG	Check if member is a KBG progr	98	00010	N
P93KL	File Server Key Lists	93	00010	N
P93801	Create/Modify Program Types	93	00040	N
P938011	Logic Module Compare/Update	93	00020	N
P938012	Program Type Compare/Update	93	00020	N
P938013	Update/Merge Application Gener	93	00030	N
P938014	Print Logic Module	93	00020	N

Opt: I=SVR 2=CRSE Specs F4=More Data F24=More Keys

Understand Directives

This chapter contains these topics:

- [Section 32.1, "Functional Directives,"](#)
- [Section 32.2, "Substitution Directives,"](#)
- [Section 32.3, "Exception Directives,"](#)
- [Section 32.4, "Conditional Directives."](#)

Directives are in the logic modules and instruct the program generator what 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

JD Edwards World supplies all directives and you cannot create your own directives.

32.1 Functional Directives

Functional directives:

- Control major functions within a program.
- Provide the initiation point for creating database specific logic and screen or report file control logic.
- Initiate the inclusion of copy modules into the source code.
- Seize detail logic modules for inclusion.

Functional directives are only found within primary logic modules and cannot reside in a detail logic module.

Following are the functional directives JD Edwards World includes with the software:

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

Directive Code	Detail Logic Module	Source Created	Functional Directive
AUTHR	None	F spec	Program author
*AUTOI	X*ENTRYI	S999	Automatic inquiry at execution test logic
*CLRNR	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		S	Display logic for primary video fields
	XDSPFLD1	004	Format Alpha field for output
	XDSPFLD2	S004	Format Gregorian Date for output
	XDSPFLD3	S004	Format Julian Date for output
	XDSPFLD4	S004	Format VC0 field from VTX
	XDSPFLD5	S004	Format VC0 field from designated description file (field details)
	XDSPFLD6	S004	Format VC0 field from F0005
	XDSPFLD7	S004	Format Alpha 3 or 28
	XDSPFLD8	S004	Repeat of XDSPFLD1
DSP2			Display logic for primary video fields
	XDSPFLD1	S004	Format Alpha field for output
	XDSPFLD2	S004	Format Gregorian Date for output
	XDSPFLD3	S004	Format Julian Date for output
	XDSPFLD4	S004	Format VC0 field from VTX
	XDSPFLD5	S004	Format VC0 field from designated description file (field details)
	XDSPFLD6	S004	Format VC0 field from F0005
	XDSPFLD7	S004	Format Alpha 3 or 28
	XDSPFLD8	S004	Repeat of XDSPFLD1
*EMK	XLOADMK	S999	Load user defined error messages

Directive Code	Detail Logic Module	Source Created	Functional Directive
ENTRY	X*ENTRYYP X*ENTRYM	Various	Load program execution passed parameters
*EXITC	XEXIT-CMD0 XEXIT-CMD1	S00EX	Function key exit execution logic
*EXITS	XEXIT-SEL0	S00OP	Selection exit execution logic
*EXITW	XEXIT-SEL0	S000P	Selection exit execution logic
*FIELD			Active Data Dictionary field validation logic
	XFIELDEDT1	S005	Data Dictionary alpha edit
	XFIELDEDT2	S005	Validation n=Master - Alpha
	XFIELDEDT3	S005	Gregorian edit
	XFIELDEDT4	S005	Julian edit
	XFIELDEDT5	S005	Data Dictionary numeric edit
	XFIELDEDT6	S005	Alpha field size 10
	XFIELDEDT7	S005	User defined code edit
	XFIELDEDT8	S005	No dictionary
	XFIELDEDT9	S005	Validation n = Master - Numeric
	XFIELDEDTA	S005	Account ID
	XFIELDEDTC	S005	Cost center edit
	XFIELDEDTE	S005	Numeric field size 7
	XFIELDEDTR	S005	Right adjust
	XFIELDEDTS	S005	Validation = Master - Alpha
	XFIELDEDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDEDTU	S005	Validation = Master - Numeric
FILES	None	F spec	Program file descriptions
*FLDxx			Active Data Dictionary field validation for primary data
	XFIELDEDT1	S005	Data Dictionary alpha edit
	XFIELDEDT2	S005	Validation n=Master - Alpha
	XFIELDEDT3	S005	Gregorian edit
	XFIELDEDT4	S005	Julian edit
	XFIELDEDT5	S005	Data Dictionary numeric edit
	XFIELDEDT6	S005	Alpha field size 10
	XFIELDEDT7	S005	User defined code edit
	XFIELDEDT8	S005	No dictionary
	XFIELDEDT9	S005	Validation n = Master - Numeric
	XFIELDEDTA	S005	Account ID
	XFIELDEDTC	S005	Cost center edit
	XFIELDEDTE	S005	Numeric field size 7
	XFIELDEDTR	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
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
KEYS			Load master file key fields in subfile format.
	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
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
KEYS2	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 S004 S00EX	Business Unit security logic where xx = master field designation 1 thru 9
*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			Format data for report detail format
	XDSPFLD1	S004	Format Alpha field for output
	XDSPFLD2	S004	Format Gregorian Date for output
	XDSPFLD3	S004	Format Julian Date for output
	XDSPFLD4	S004	Format VC0 field from VTX
	XDSPFLD5	S004	Format VC0 field from description file (field details)
	XDSPFLD6	S004	Format VC0 field from F0005
	XDSPFLD7	S004	Format Alpha 3 or 28
	XDSPFLD8	S004	Repeat of XDSPFLD1
RPTH			Format data for report heading format
	XDSPFLD1	S004	Format Alpha field for output
	XDSPFLD2	S004	Format Gregorian Date for output
	XDSPFLD3	S004	Format Julian Date for output
	XDSPFLD4	S004	Format VC0 field from VTX
	XDSPFLD5	S004	Format VC0 field from description file (field details)
	XDSPFLD6	S004	Format VC0 field from F0005
	XDSPFLD7	S004	Format Alpha 3 or 28
	XDSPFLD8	S004	Repeat of XDSPFLD1
*RPTT			Format data for report total format
	XDSPFLD1	S004	Format Alpha field for output
	XDSPFLD2	S004	Format Gregorian Date for output
	XDSPFLD3	S004	Format Julian Date for output
	XDSPFLD4	S004	Format VC0 field from VTX
	XDSPFLD5	S004	Format VC0 field from description file (field details)
	XDSPFLD6	S004	Format VC0 field from F0005
	XDSPFLD7	S004	Format Alpha 3 or 28
	XDSPFLD8	S004	Repeat of XDSPFLD1
*RTA	XTOTARRAY	E spec	Load totaling arrays
*RTS	None	I spec	Report softcoding array
*RTX	None	I spec	Report softcoding text fields
*RTXI	XVTIDX	S999	Set maximum VTX index to use

Directive Code	Detail Logic Module	Source Created	Functional Directive
*SFFLD			Active Data Dictionary data field validation for subfile fields.
	XFIELDT1	S005	Data Dictionary alpha edit
	XFIELDT2	S005	Validation n = Master - Alpha
	XFIELDT3	S005	Gregorian edit
	XFIELDT4	S005	Julian edit
	XFIELDT5	S005	Data Dictionary numeric edit
	XFIELDT6	S005	Alpha field size 10
	XFIELDT7	S005	User defined code edit
	XFIELDT8	S005	No dictionary
	XFIELDT9	S005	Validation n = Master - Numeric
	XFIELDTA	S005	Account ID
	XFIELDTC	S005	Cost center edit
	XFIELDTE	S005	Numeric field size 7
	XFIELDTR	S005	Right adjust
	XFIELDTS	S005	Validation = Master - Alpha
	XFIELDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDTU	S005	Validation = Master- Numeric
SLDxx			Active Data Dictionary data field validation for subfile data fields. Where xx = specified master file 1 thru 9.
	XFIELDT1	S005	Data Dictionary alpha edit
	XFIELDT2	S005	Validation n = Master - Alpha
	XFIELDT3	S005	Gregorian edit
	XFIELDT4	S005	Julian edit
	XFIELDT5	S005	Data Dictionary numeric edit
	XFIELDT6	S005	Alpha field size 10
	XFIELDT7	S005	User defined code edit
	XFIELDT8	S005	No dictionary
	XFIELDT9	S005	Validation n = Master - Numeric
	XFIELDTA	S005	Account ID
	XFIELDTC	S005	Cost center edit
	XFIELDTC	S005	Numeric field size 7
	XFIELDTR	S005	Right adjust
	XFIELDTS	S005	Validation = Master - Alpha
	XFIELDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDTU	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

Directive Code	Detail Logic Module	Source Created	Functional Directive
*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

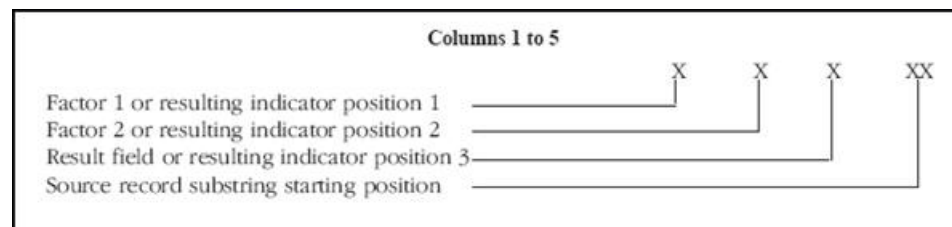
* Automatically include JD Edwards World standards, which are beyond normal requirements.

32.2 Substitution Directives

Substitution directives:

- Control the translation of symbolic names to the actual data field names the system requires for an individual line of source code.
- Substitute information within a line of code.
 - If the system replaces a field, the field it replaces begins with an & (ampersand).
 - If the substitution is positional, this directive informs the program generator where to position the substitution on a line of code.

Figure 32–1 Substitution Directives, Columns 1 to 5



Following are the substitution directives JD Edwards World includes with the software:

Directive	Column Allowed	Column Allowed	Column Allowed	Column Allowed	Function
	1	2	3	45	
@	x	x	x	x	Four character Data Dictionary name
#	x				Primary parameter that passes for *ENTRY
A	x	x	x		Highest VTX field.
B					Unused at this time.

Directive	Column Allowed	Column Allowed	Column Allowed	Column Allowed	Function
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
Q				x	Field name to receive description value

Directive	Column Allowed	Column Allowed	Column Allowed	Column Allowed	Function
R				x	Field name to receive 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

Directive	Column Allowed	Column Allowed	Column Allowed	Column Allowed	Function
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

32.3 Exception Directives

Exception directives:

- Provide unusual option definition to the program generation process.
- Combine two other types of directives. For example:

DSPF &01FILE

Combines a functional directive (DSPF) with a substitution directive (&01FILE), so it is an exception directive.

Example:

You create a line of code for the READ Master file and then substitute the Master file name.

Most exception directives are substitution directives but are out of the normal syntax substitution directives use.

Following are the exception directives JD Edwards World includes with the software:

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

Factor 1	Factor 2	Result	Keyword	Function
x	x		&xxPGCTL	Number of subfile records in 1 page
x			%	Factor 1 intentionally left blank
x	x	x	=	User defined calculation logic result

32.4 Conditional Directives

Conditional directives:

- Are the most flexible and most powerful directives.
- Verify specific conditions exist before determining if the system must perform any action.
- Use positions 1 to 5 to provide directive initiation and use Factor 1, Factor 2, and the Result field to complete the directive definition.

Subroutine S010-11 includes examples of conditional directives. For example:

If SFSELC exists, include code for selection exits.

You can combine conditional directives.

Following are the conditional directives JD Edwards World includes with the software:

Position/Factor/Result	Description
Pos 1	+ Include detail logic module if true
	- Include detail logic module if false
Pos 2-5	FLDN Test existence of data field
	DTAI Test existence of data item
	FILE Test existence of file
	FMT Test existence of file
Factor 1	Name of field, item, file or format to test. Can also contain *ANYx for file test which you can use to test for types of files in a program where x might optionally designate number of files.
Oper (file test only)	DSPF Display file
	PF Physical file only
	LF Logical file only
	PRTF Printer file only
	DB Database file
Factor 2	Name of detail logic module to include into source code. Might also use *AND to produce compound conditions
Result Field Pos 1	@ Any input file
	M Master input file with M in file specifications
	1-9 Master input file with 1 - 9 in field specifications

Position/Factor/Result	Description
Result Field Pos 2	@ Any output file
Result Field Pos 3	@ Any update file
	M Master update file with M in file specifications
	1-9 Master update file with 1 - 9 in field specifications
Result Field Pos 4	@ Any add file

Work with the Question and Answer System

This chapter contains these topics:

- [Section 33.1, "About Simple Question & Answer,"](#)
- [Section 33.2, "Reviewing Questions in a Master Dialogue,"](#)
- [Section 33.3, "Adding New Q & A Dialogue,"](#)
- [Section 33.4, "Working with an Existing Dialogue."](#)

The program generator uses JD Edwards World Question and Answer system as a method of determining the appropriate program type. Based on the answers to certain questions, the system selects a program type for you.

You can create your own questions and answers to produce your own custom program type. You can also modify the questions, known as a dialogue, the program generator uses through this feature.

Navigation

From Model Program Design Menu (G9361), choose Maintain Q/A

From World CASE Q & A Menu (G9364), choose Simple Question and Answer

33.1 About Simple Question & Answer

The Simple Question & Answer screen, from which you begin all Question & Answer tasks, includes the following three fields.

Option	Description
Simple Question and Answer	Access the Question Entry screen.
Add New Q & A Dialogue	Access the Dialogue Descriptions screen.
Update Existing Q & A Dialogue	Access the Dialogue Lists screen.

33.2 Reviewing Questions in a Master Dialogue

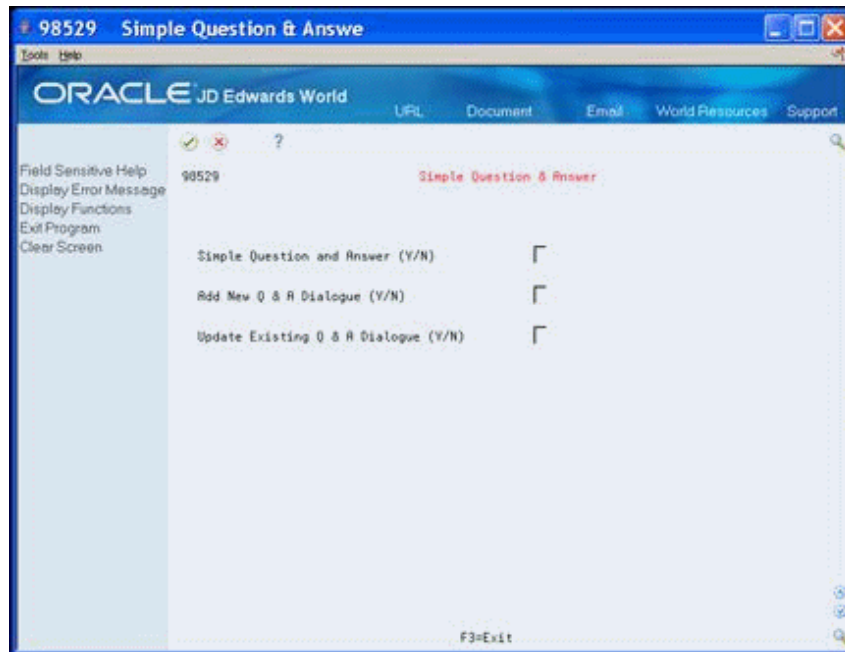
You choose Display Next Question (F19) and Display Previous Question (F20) to review all of the master questions. Additionally, you can make changes to the master question that displays on the Question Entry screen.

To review questions in a master dialogue

1. On Simple Question and Answer, enter Y in the following field:

- Simple Question and Answer

Figure 33–1 Simple Question & Answer screen

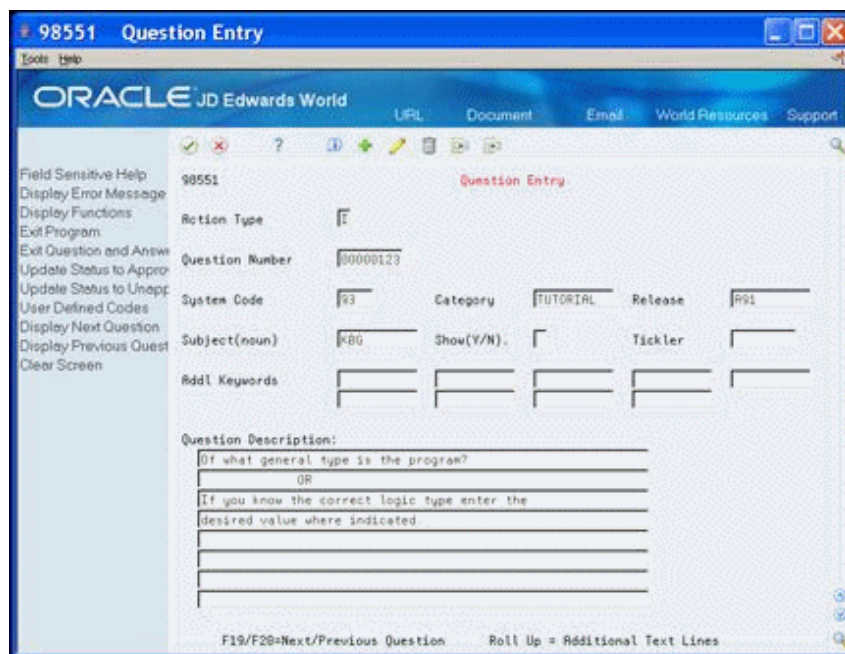


2. On Question Entry, complete the following field:

- Question Number

The question detail displays.

Figure 33–2 Question Entry screen



3. To review the answers to the master question Click Change.

33.3 Adding New Q & A Dialogue

You can create your own questions and answers by completing the Add New Q & A Dialogue field.

The dialogue the CASE tool uses to determine the program type is Primary Key
*Default, Data Item LC.

To add new Q&A dialogue

1. On Simple Question & Answer, enter Y in the following field:

- Add New Q & A Dialogue

The Dialogue Descriptions screen displays.

2. Complete the following fields:

- Dialogue Keys: Primary
- Dialogue Keys: Secondary
- Dialogue Type
- Summary description
- Beginning Question Number
- Dialogue description

Figure 33–3 Dialogue Descriptions screen

The first Question Entry screen displays.

3. Complete the following fields:

- Question Number
- System Code
- Category

- Release
 - Subject
 - Show
 - Tickler
 - Question Description
4. Complete the following field to assist in future searches for this question:
- Additional Keywords

Figure 33–4 Question Entry (New) screen

98551 Question Entry

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Exit Question and Answer
Update Status to Approved
Update Status to Unapproved
User Defined Codes
Display Next Question
Display Previous Question
Clear Screen

98551 Question Entry

Action Type

Question Number

System Code Category Release

Subject(noun) Show(Y/N) Tickler

Addl Keywords

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

The Answer Entry screen displays.

Figure 33–5 Answer Entry screen

5. Complete the following fields:

- Question Number
- Answer Number
- Next Question
- Answer

6. Optionally, complete the Return Value field.

The Return Value field can contain a member name, or *PROMPT. *PROMPT lets the user manually complete the Return Value field.

The following screen illustrates the use of the Return Value field. The return value is the program type for an interactive window program. In this case, there is no next question. The dialogue ends after the system enters the value E0010.

When you click Add the value in the Next Question field changes. There is no value in the Return Value field. For any question, there is either a value in the Return Value field or the Next Question field.

Figure 33–6 Answer Entry (Next Question) screen

For an existing question and answer, you choose Display Next Question (F19) and Display Previous Question (F20) to review all other possible answers for this question.

7. To create a second answer to the question, click Enter.
The screen clears.
8. Enter the number of the next answer in the following field:
 - Answer Number
9. Repeat the previous steps as necessary to enter the next question and answer.
10. To define the next question, click Exit to return to the Question Entry screen.
11. Repeat the previous steps as necessary to enter questions and answers.
12. Click Exit until the Simple Question & Answer screen displays.

33.4 Working with an Existing Dialogue

You can perform the following tasks using the Update Existing Q & A Dialogue feature:

- To review a dialogue flow
- To change a dialogue
- To copy a dialogue
- To rename a dialogue
- To run a dialogue
- To delete a dialogue
- To run a quiz

To perform these tasks, choose Update Existing Q & A Dialogue on the Simple Question and Answer screen to access the Dialogue Lists screen.

On Dialogue Lists, you can use the following values in the Option field:

2 - Change the questions and answers for the Dialogue

3 - Copy one dialogue to another dialogue

5 - Run the Q & A. You can specify the number of responses to allow.

6 - Flow displays the flow of the Q&A and how one question leads to another. You can access Q&A Revisions from the flow.

7 - Rename

9 - Delete

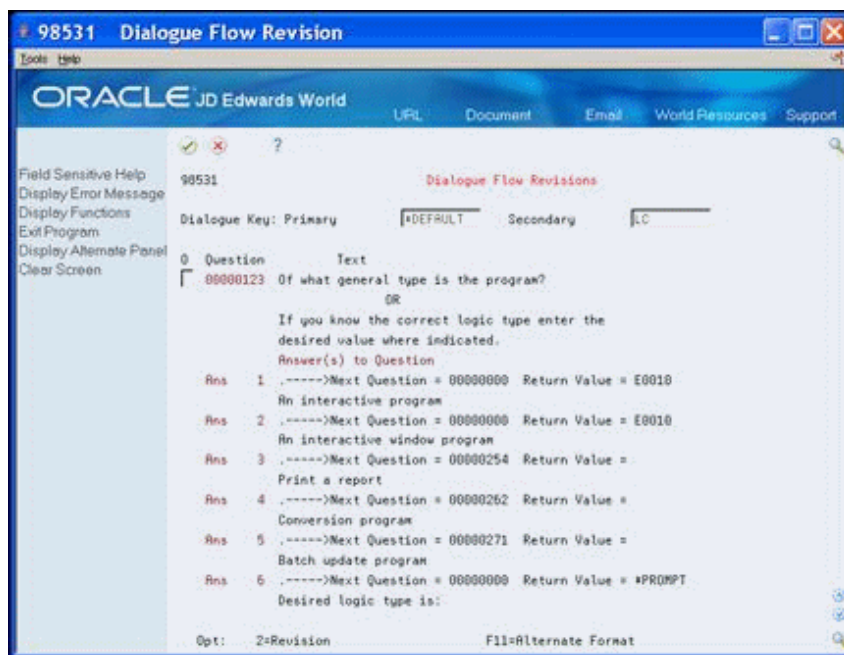
11 - Take a quiz from this screen

To review a dialogue flow

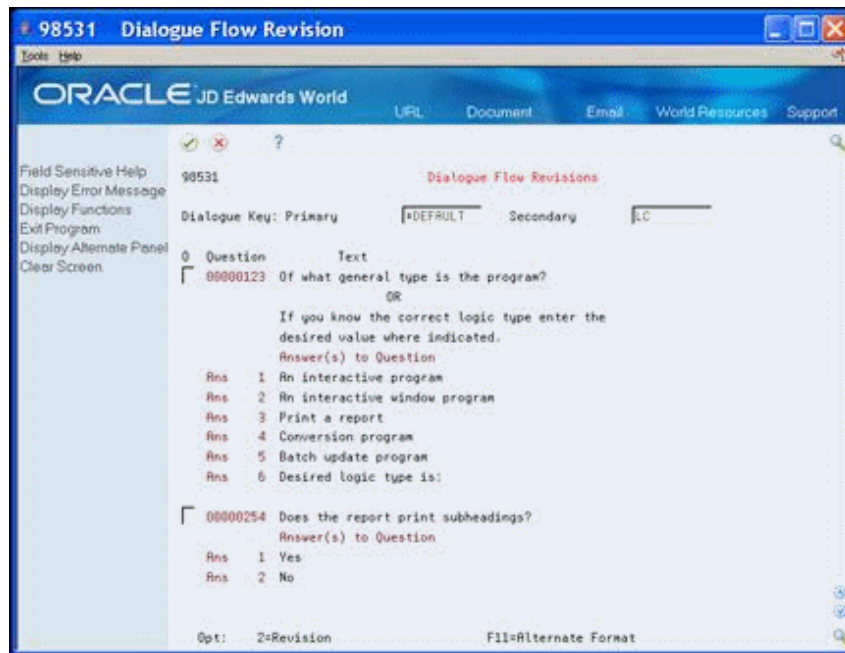
1. On Dialogue Lists, enter 6 in the Option field.

The Dialogue Flow Revisions screen displays.

Figure 33–7 Dialogue Flow Revision screen



2. Choose Alternate Format (F11) to view the Alternate Format.

Figure 33–8 Dialogue Flow Revision (Alternate Format) screen**To change a dialogue**

1. On Dialogue Lists, enter 2 in the Option field next to the dialogue you want to change.

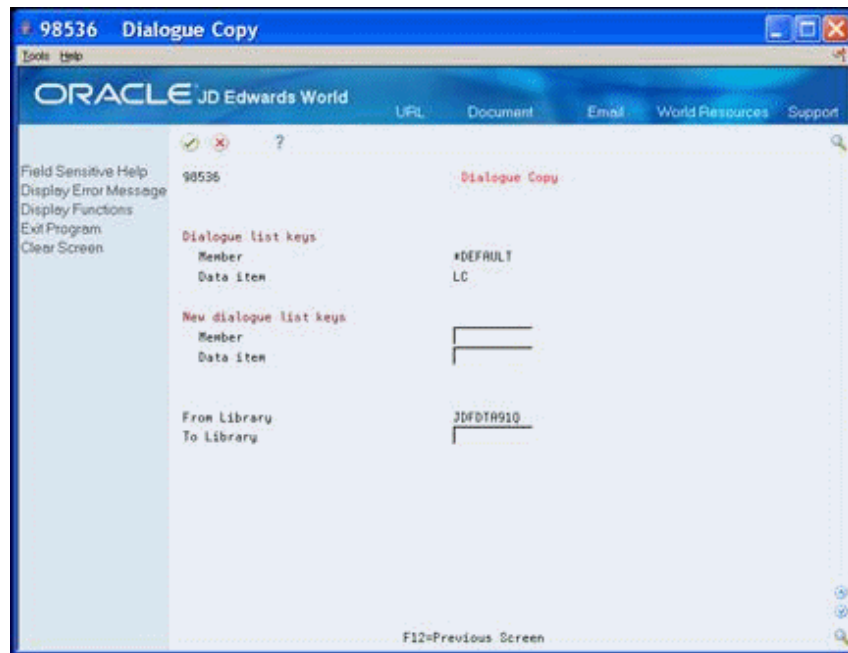
The Question Entry screen displays.

2. Click Change.
3. Make the changes to the questions and answers.

To copy a dialogue

1. On Dialogue Lists, enter 3 in the Option field next to the dialogue you want to copy.

The Dialogue Copy screen displays.

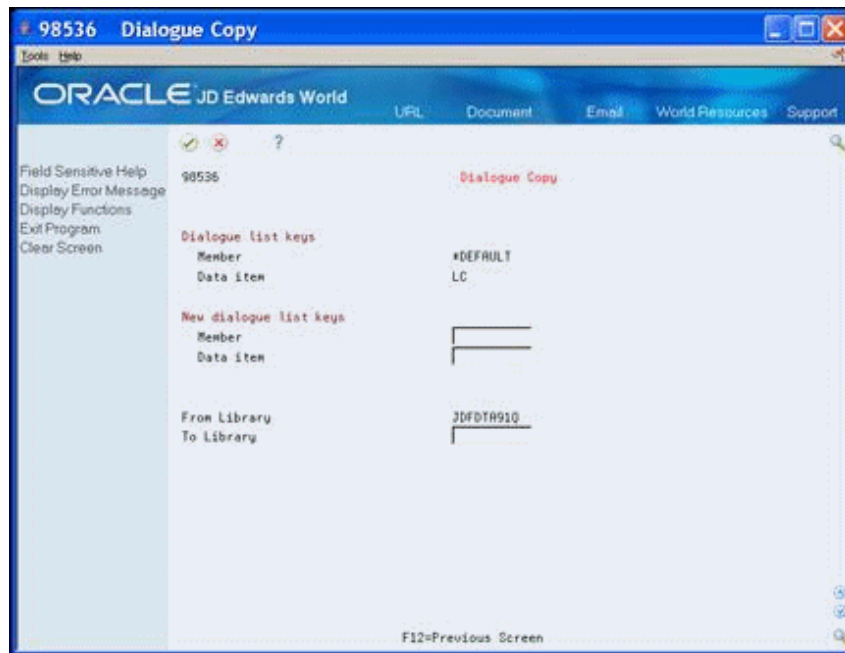
Figure 33–9 Dialogue Copy screen

2. Enter the name of the new dialogue list keys (primary key) in the following field:
 - Member
3. Enter the name of the new dialogue list keys (secondary key) in the following field:
 - Data item
4. Enter the name in which the new dialogue resides in the following field:
 - To Library

The system copies the dialogue.

To rename a dialogue

1. On Dialogue Lists, enter 3 in the Option field next to the dialogue you want to rename.
The Dialogue Copy screen displays.

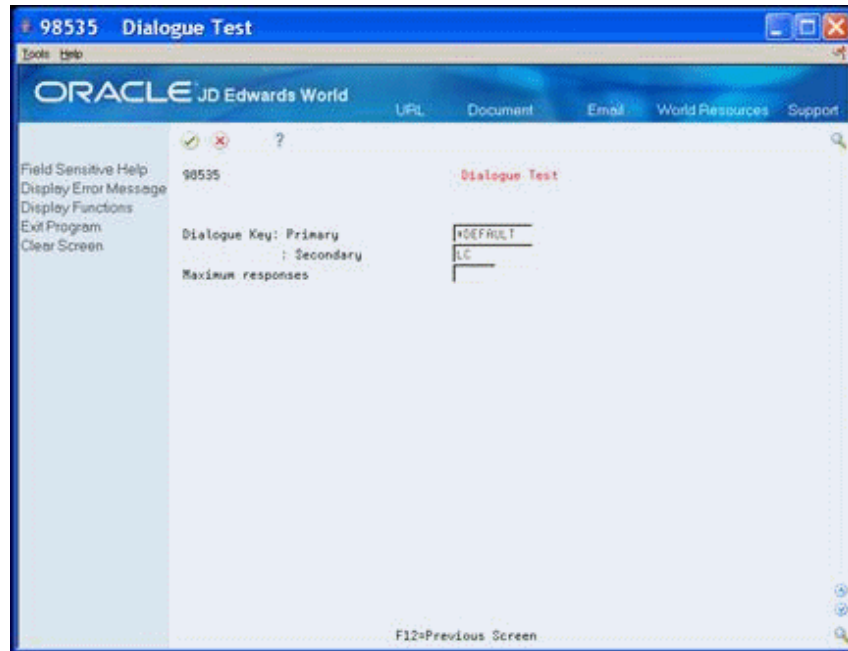
Figure 33–10 Dialogue Copy (Rename) screen

2. Enter the name of the primary key in the following field:
 - Member
3. Enter the name of the secondary key in the following field:
 - Data item
4. Enter the name in which the new dialogue resides in the following field:
 - To Library

The system renames the dialogue.

To run a dialogue

1. On Dialogue Lists, enter 5 in the Option field next to the dialogue you want to run.
The Dialogue Test screen displays.

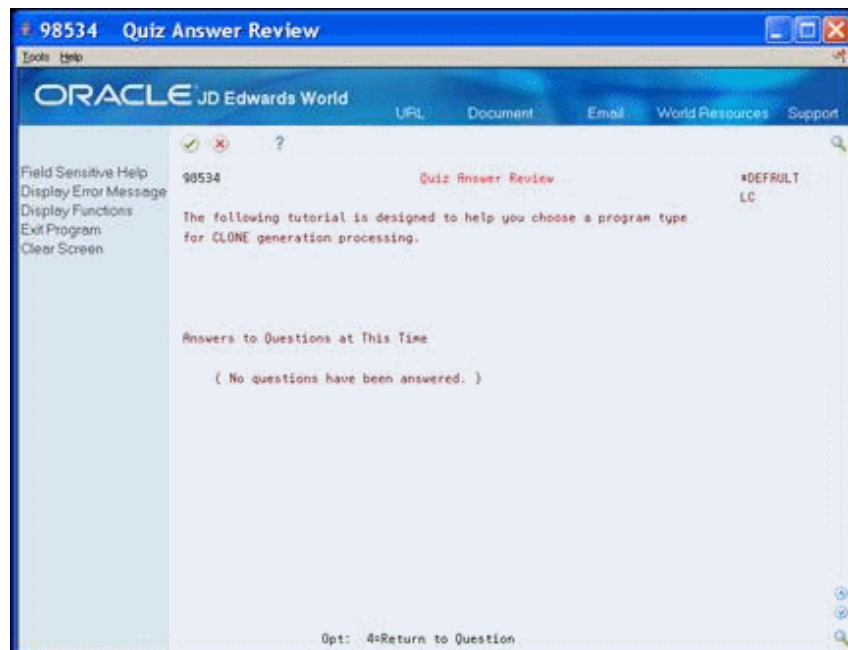
Figure 33–11 Dialogue Test screen

2. Enter a number for the maximum number of times you want to run this dialogue in the following field:

- Maximum responses

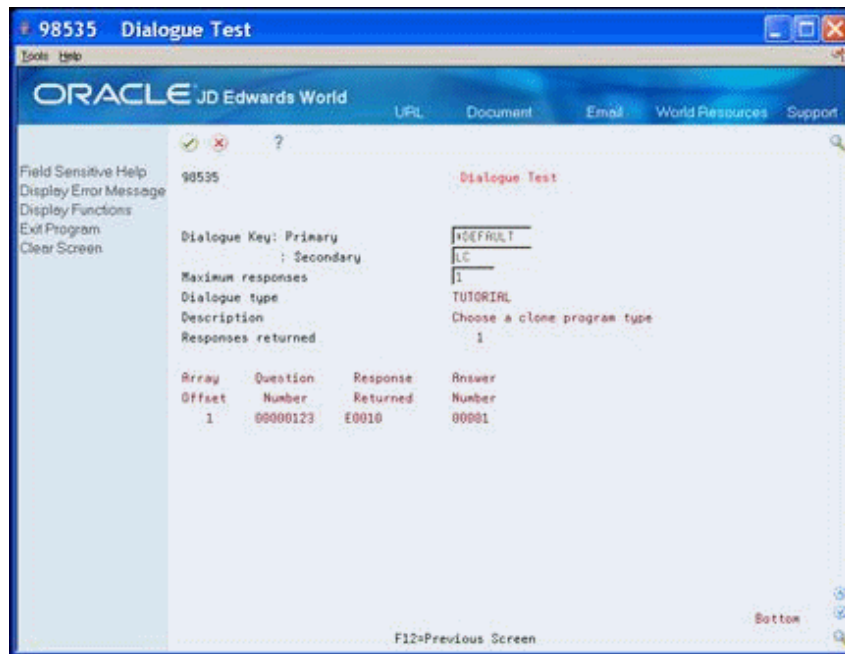
The questions of the dialogue display in sequence. When you reach the last question, a message displays at the bottom of the Dialogue Selection screen.

3. Choose Review Selections (F5) when the last question displays, to access the Quiz Answer Review screen.

Figure 33–12 Quiz Answer Review screen

4. The screen displays with the information about your answers. To review the remaining questions and answers, choose Display Next Question (F19) and Display Previous Question (F20).
5. Enter 4 to return to a specific question.
6. Click Exit on the last question screen to display the Dialogue Test screen.

Figure 33–13 Dialogue Test (Exit) screen



The Dialogue Test screen displays the values for number of times the dialogue was run, the Responses Returned at the end of the dialogue, and the Answer Number of the last question.

7. Click Exit (F3) to return to the Dialogue Lists screen.

To delete a dialogue

On Dialogue Lists, enter 9 in the Option field next to the dialogue you want to delete. The system deletes the dialogue.

To run a quiz

To run a quiz, the dialogue type must be QUIZ.

1. On Dialogue Lists, enter 11 in the Option field next to the quiz you want to run.

The first question of the quiz displays.

2. Answer the questions.

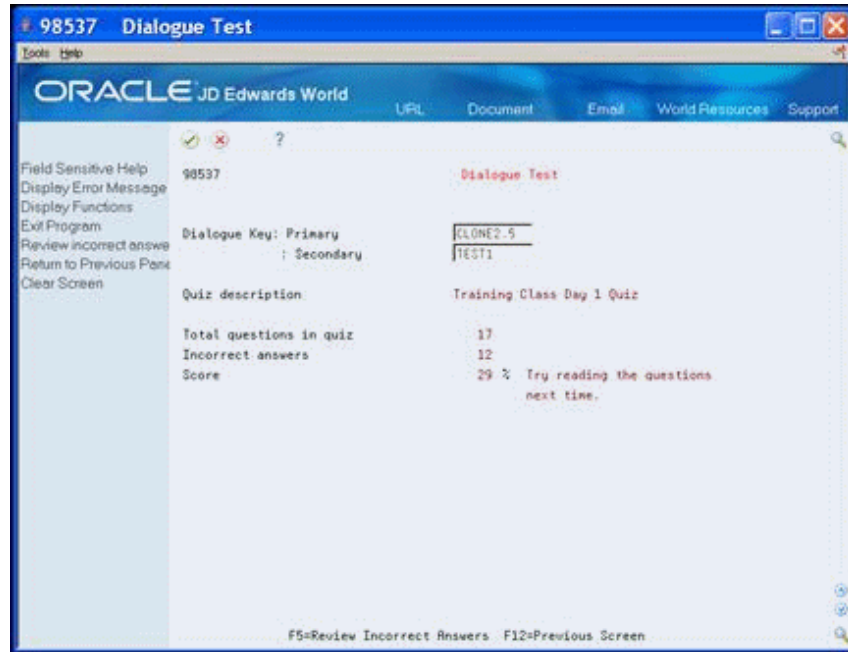
When you finish answering the questions, a message displays at the bottom of the last screen, Question and Answer complete.

3. Perform one of the following:
 - To review choose Review Incorrect Answers (F5)
 - Click Enter

When you choose Review Incorrect Answers, the questions and answers display on the screen. When you click Enter, the system calculates the number of errors and displays your score.

Choose Review Incorrect Answers (F5) from this Dialogue Test screen to review your errors.

Figure 33–14 Dialogue Test (Review Incorrect Answers) screen



Create User Defined PDL

This chapter contains the topic:

- [Section 34.1, "Creating User Defined PDL."](#)

34.1 Creating User Defined PDL

You attach *PROCs to either a master file field or to a device file field (screen or report). If you attach them to a master file field, then the system places the code it generates in S005. If you attach them to a device file field, then the system places the code it generates in S004.

The purpose of User Defined PDL Entry Points is to allow you to create *PROCs in any subroutine and to allow them to exist without attaching them to a master file field or device file field. You define the entry points within subroutines where you enter PDL code using the Detailed Programming Facility. User Defined PDL Entry Points are a functional directive that you can enter into a primary logic module.

The User Defined PDL Entry Points cause the system to create RPG code in the same manner as users entering the PDL using the Detailed Programming Facility; however, you connect it to logic modules instead of fields.

In any primary logic module you can insert up to 99 PDL directives. Ordinarily, you number the first one PDL01, the second one PDL02, and so forth. PDL directives do not have to be in sequential order; however, each PDL directive must have a unique number within that logic module.

For all single record maintenance forms, you create a user defined PDL entry point in the mainline subroutine. Enter PDL to bring in a default value for a constant field.

For example, change the logic module MAINLINE because this module creates the mainline code for all single record maintenance forms.

Before You Begin

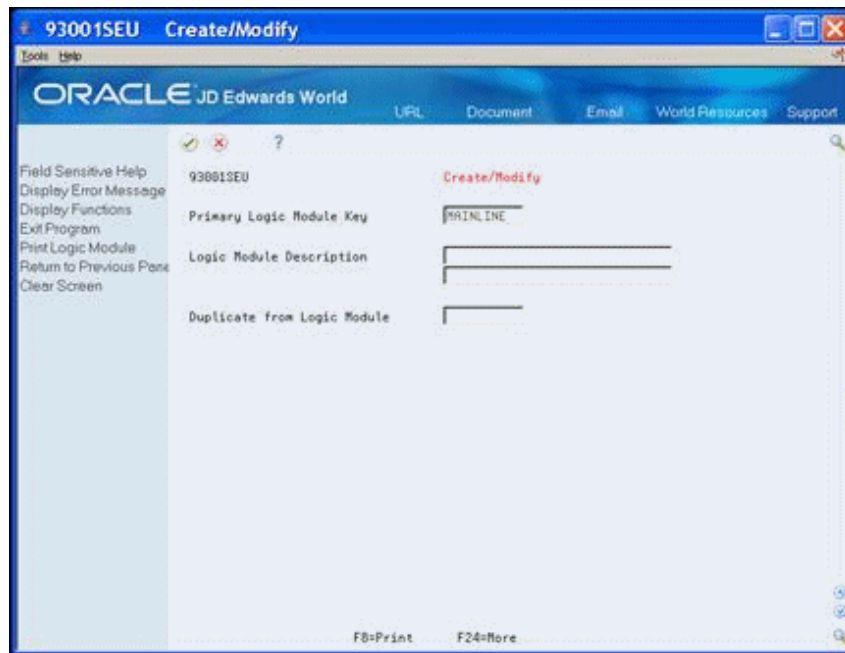
You must be able to locate program types and logic modules. See [Section 30.4, "Creating or Modifying Program Types"](#) and [Chapter 31, "Work with Logic Modules."](#)

To create user defined PDL

Navigation

From Model Program Design Menu (G9361), under LOGIC MODULES, choose Create/Modify

1. On Create/Modify (Logic Module), access the Edit screen.

Figure 34–1 Create/Modify (User Defined PDL) screen

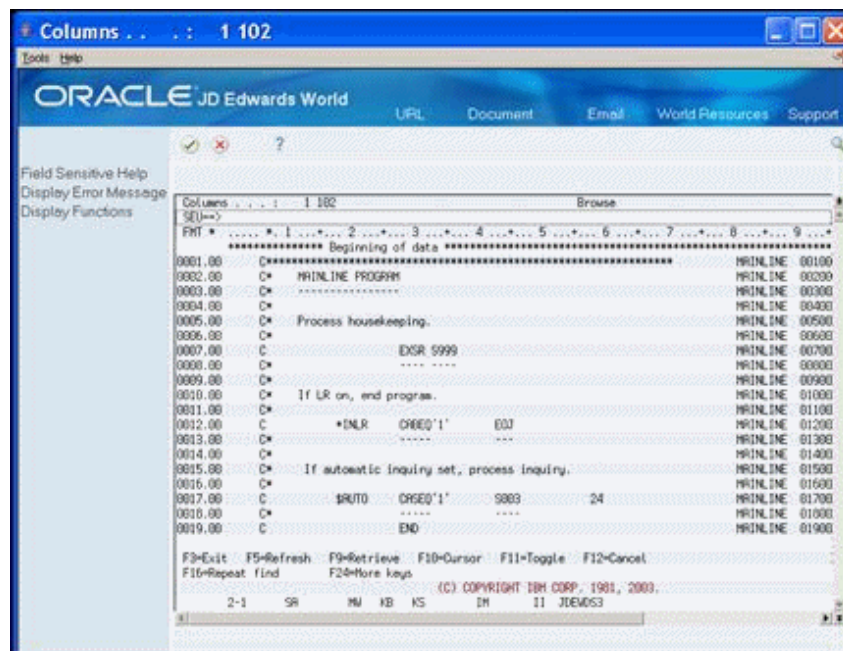
2. On the Edit screen, create the user defined PDL entry points within the affected primary logic modules.

The naming convention for user defined PDL entry points is PDLxx, where xx is a two digit number between 01 and 99.

You can either add the PDL directive to an existing line of code that does not contain a directive, or insert a new line and enter 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. (Move to column 80.)

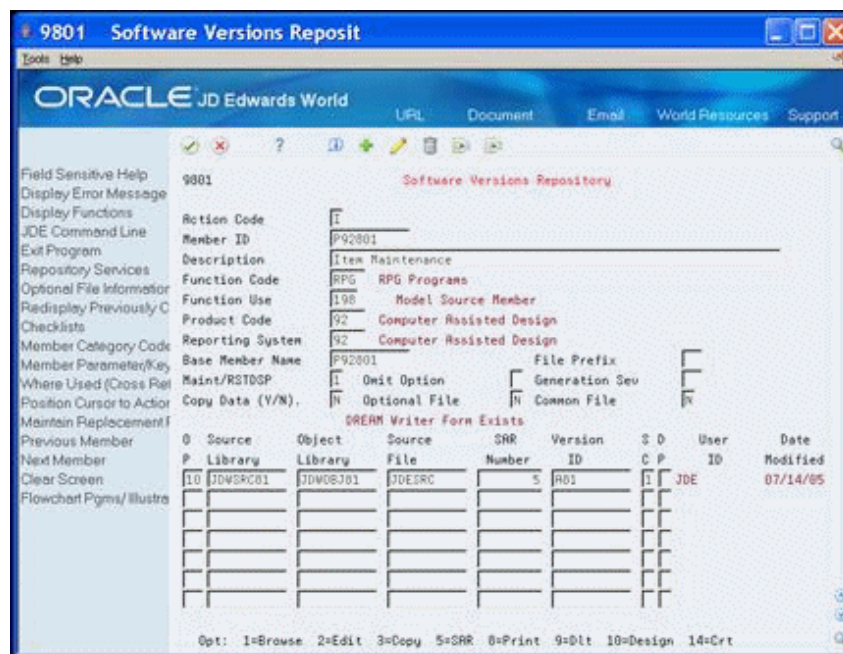
In this example, you enter PDL01 on line 9. Any PDL code that you enter for this entry point will come immediately after the statement EXSR S999 and before the test for *INLR.

Figure 34–2 Edit PDL Code screen



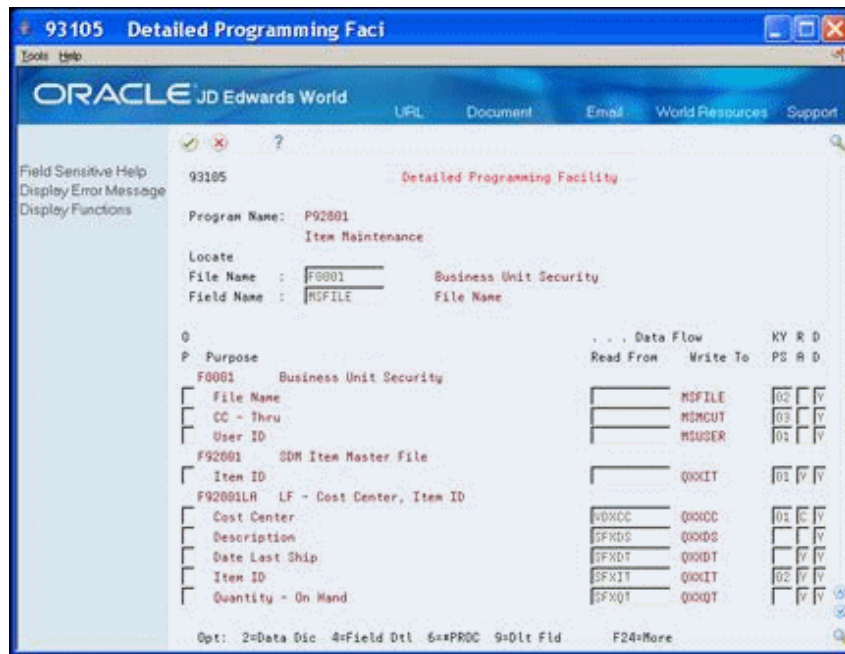
3. Access the Software Versions Repository and locate the member.

Figure 34–3 Software Versions Repository (User Defined PDL) screen



4. Enter 10 (Design) next to the environment in the Option field:
The Program Generator Specification screen displays.

- On Define Generator Specification, enter 1 in the Option field to access the Detailed Programming Facility:
The Detailed Programming Facility screen displays.



6. Enter the PDL code through the Detailed Programming Facility.
All user defined PDL entry points appear after the form or report file fields in the Detailed Programming Facility.

7. On Detailed Programming Facility, enter 6 in the Option field to access Data Item Formula Revisions.
8. Enter the PDL code.

Figure 34–6 Data Item Formula Revisions (User Defined PDL) screen

93109 Data Item Formula Revisio

Tools Help

ORACLE JD Edwards World

URL Document Email World Resources Support

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Display Variable Definit
Repository Services
Clear Screen

93109 Data Item Formula Revisions

Action Code R
Program ID F92801
File ID F92801LR
Field Name 000000

Data Item Formula

F5=Variables F6=Repository Services F24=More

Program Generator Checklist

This appendix contains these topics:

- [Section A.1, "Data File Design Aid,"](#)
- [Section A.2, "Screen Design Aid,"](#)
- [Section A.3, "Report Design Aid,"](#)
- [Section A.4, "Program Generator."](#)

You can use the following to aid you as you use the Program Generator. It includes topics that you should consider as you use the Program Generator.

A.1 Data File Design Aid

Attempt to create files with keys to avoid having to process by relative record numbers. Processing by relative record numbers is more complex than processing by keys.

A.2 Screen Design Aid

For Subfile Programs, you should define:

- A hidden field for the parts of the file key that the subfile video uses if you create a maintenance subfile.
- The hidden field, SH#RRN if processing by relative record numbers.
- A hidden field for the data structure if processing by relative record numbers.

You should:

- Note the video fields that relate to VC0 fields. You need this information in the Detailed Programming Facility in order to load the VC0 fields.
- Note the error indicators the system assigns to screen fields.
- Prepare final checks:
 - Did you assign the Default Cursor keyword to a video field?
 - Did you allow for upper and lower case on description fields?
 - Did you specify a K in the Edited field for the key fields only?

Note: The Screen Design Aid is currently not available in the Java platform for JD Edwards World software. You must perform these tasks using the green screen platform of the software.

A.3 Report Design Aid

- Change the Start/End lines for format HEADING1 from 1-4 to the length you need. Usually 1-8 will suffice.
- Add DETAIL1 format.
- Add TOTAL1 format if using hierarchical (dynamic) totaling. You must include one or all of the following fields which enable dynamic totaling:
VC1ROW, VC1KEY, VC1DSC, VC0TO2
- Add HEADING2 format if using subheadings. You must include one or all of the following fields that enable subheadings to work properly:
VC0ROW, VC0KEY, VC0DSC

A.4 Program Generator

- All Programs:
 - Ensure the CAP Status is set to Y. If CAP Status is not set to Y, then something could have ended abnormally.
 - Verify the program type.
- File Specifications:
 - You must specify one file with an M. Do not specify one file with an M and another with 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.
 - You must specify a video or report file.
 - You cannot include description files if a field is in the HEADING2 format for a report.
- Option and Function Exits:
 - Ensure the program the system is launching is setup to accept parameters sent by the function key or selection exit.
 - Modify any CL programs that also launch a program to send blank parameters. For example, CALL Pxxxxx PARM()
 - Ensure the program to retrieve exists before using it on this screen.
 - Attempt to send PSxxxx fields instead of VDxxxx or SFxxxx fields.
 - May inadvertently change in the program the system launches.
 - You might have to define and load the PSxxxx fields manually.
- Detailed Programming Facility:
 - Enter N in the Entry Optional field for key fields in a subfile. You specify this for the subfile fields, not the hidden fields as it enables the delete function.
 - Link VC0 fields to description files.

- Use PDL in the Detailed Programming Facility to:
 - Affect subroutine S005 when you enter it with a data base field.
 - Affect subroutine S004 when you enter it with a video field.
 - Use the Return keyword to omit the standard code the Program Generator creates.
- Specify a PLIST sequence if the program is going to receive parameters from another program. Use the video fields for this instead of the file fields.
- Use *OUTPUT to retrieve the row description from the Data Dictionary for fields that you only use in the HEADING2 format and not the DETAIL1 format.

Programming Standards

This appendix contains these topics:

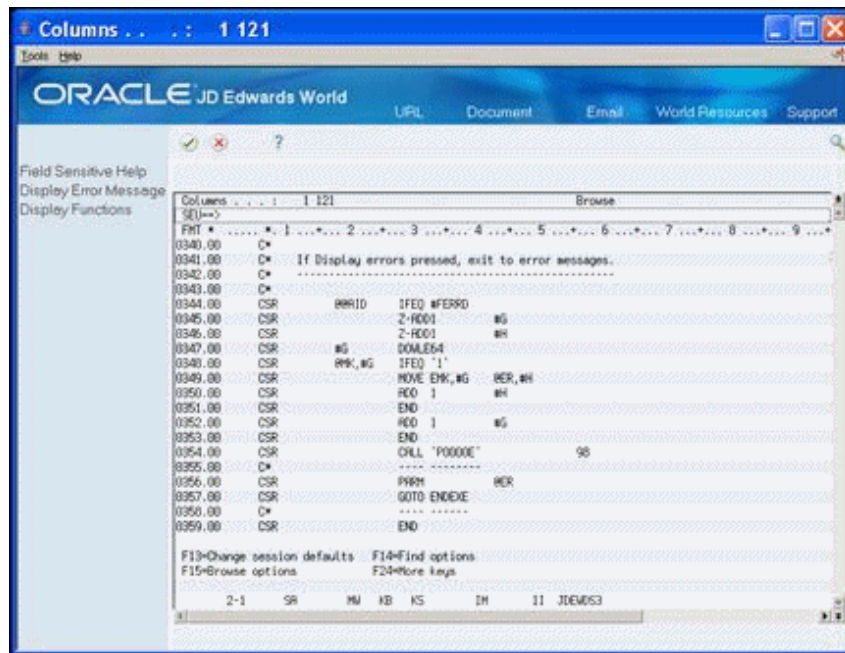
- [Section B.1, "Error Handling,"](#)
- [Section B.2, "Indicator Usage,"](#)
- [Section B.3, "Naming Conventions,"](#)
- [Section B.4, "Key List \(KLIST\),"](#)
- [Section B.5, "Work Fields,"](#)
- [Section B.6, "Current Date and Time."](#)

B.1 Error Handling

JD Edwards World includes an efficient means of handling errors using arrays. The following illustrate the error handling arrays within the Single Record Maintenance Program.

- The EMK array contains the four byte data dictionary name of every error that can occur in this program.
- The @MK array maintains an indicator for each error in EMK. If one of the errors occurs, the system activates the indicator.
- The @ER array loads the error messages when you choose Display Error Message (F7) to view the errors.

The code to launch the error message handling program follows.

Figure B–1 Error Message Handling Program Code screen

If an error indicator 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 (@ER).

The system loads the array which contains every possible error for this program only once (in S999).

B.2 Indicator Usage

There are 99 indicators available for use. JD Edwards World groups them by purpose and includes standards for the use of the indicators that are in the following table. JD Edwards World does not include standards for indicators that are not in the following table.

INDICATOR	DESCRIPTION
01	Causes the Invalid Function Key Pressed message to display
02	Dictates the color palette to use
04	Controls subfile keywords SFLDROP and SFLFOLD for fold areas
20	Controls the clear screen action code
21	Controls the add action code
22	Controls the change action code
23	Controls the delete action code
24	Controls the inquire action code
31	The system uses this in conjunction with subfile processing to initiate the INVITE or SFLCLR keyword
32	The system uses this in conjunction with subfile processing initiating the keyword SFLNXTCHG
37	The system uses this in conjunction with subfile processing to highlight the last record in the display (used only with inquiry subfiles)

INDICATOR	DESCRIPTION
38	The system uses this in conjunction with subfile processing to control the display keyword SFLDSP
42-79	The system uses this for error processing to indicate which fields are in error
40	The system reserves this for errors in the Action Code field
41	The system reserves this for errors in the key fields
80-89	Indicates a general reusable one-time indicators
93	Indicates 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

B.3 Naming Conventions

Use the following characters in the first place of the naming convention to distinguish different item names:

- @ - Array names
- \$ - Field names the program creates for flags and work fields
- # - Fields you define in common subroutines

B.4 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:

- XXXKY01 for physical files where XX = the file prefix. For example: ABKY01
- When a physical file needs to have more than one key list in a program, you denote the successive files by the last character. For example, for three key lists for the physical F0101, the key lists are: ABKY01, ABKY02, and ABKY03.
- XXXKY0x 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 file needs to have more than one key list in a program, you denote the successive files by the penultimate character. For example, the three key lists for the logical F0101LA are: ABKY0A, ABKY1A, and ABKY2A.

B.5 Work Fields

Define work fields only once within a program. JD Edwards World recommends that you use the LIKE DEFN command to define work fields when their attributes directly tie 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,
```

You 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 an indicator, you should assign the prefix \$ and enter descriptive text for the remainder of the name. 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 in the data structure I00DSINX.

B.6 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 on which you compiled the program. You cannot change the date format without recompiling the program. TIME uses the system's date format when it executes the operations code.

Note: The TIME operation requires significant system resources. If possible, use it only once for a program. Typically, this would occur in the Housekeeping Subroutine (S999).

Always use program X0028 to edit dates and format them for output.

CASE Program Types

This appendix contains these topics:

- Section C.1, "Guidelines,"
- Section C.2, "A0010 - Interactive Subfile Inquiry,"
- Section C.3, "A0020 - Interactive Single Record Inquiry,"
- Section C.4, "B0010 - Interactive Single Record Maintenance,"
- Section C.5, "C0010 - Batch Report with Totals,"
- Section C.6, "C0020 - Batch Report with Totals and Subheadings,"
- Section C.7, "C0025 - Batch Report with Totals and Subheadings,"
- Section C.8, "D0010 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number,"
- Section C.9, "D0020 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number,"
- Section C.10, "D0030 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record,"
- Section C.11, "D0040 - Interactive Subfile Maintenance with Action Code, with Options, by Key,"
- Section C.12, "D0050 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number,"
- Section C.13, "D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key,"
- Section C.14, "D0070 - Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number,"
- Section C.15, "D0080 - Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number,"
- Section C.16, "D0090 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance,"
- Section C.17, "D0100 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key,"
- Section C.18, "E0010 - Interactive Window,"
- Section C.19, "X0010 - Batch Update with Report,"
- Section C.20, "X0020 - Batch Update,"

- [Section C.21, "X0030 - Batch Update with Subroutine S001,"](#)
- [Section C.22, "X0040 - Batch Update with Report,"](#)
- [Section C.23, "Y0010 - Conversion, Two Files with Error Report,"](#)
- [Section C.24, "Y0020 - Conversion, One File Update with Error Report,"](#)
- [Section C.25, "Y0030 - Conversion, One File Write with Error Report."](#)

This appendix includes each program type, the requirements for each program type, and its use. Use this as a quick reference for all program types.

C.1 Guidelines

The following are optional:

- General help instructions, however JD Edwards World highly recommends you include these.
- Detail (fold) areas and AAI's within program types.
- Processing Options. Define processing options for batch processing. The step in the program generator which automatically includes the logic to retrieve this information is subroutine S999. Define the special calculations to use the processing options.
- User defined options and function exits for all program types.
- Calculations you can create using Program Design Language in the Detailed Programming Facility.
- VC0 description fields updates.

C.2 A0010 - Interactive Subfile Inquiry

C.2.1 Description

Use this program type to create an interactive subfile program. This subfile program is for inquiry purposes only. This program type processes a single master file by key. You cannot set the Lockout Action Codes fields. Create a display file prior to generating this program type.

C.2.2 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 the Screen Design Aid (SDA) with the value K. If you are using the Data Base Field Selection feature in SDA, the known key field updates automatically.

The use of an Action Code is optional. Enter a default cursor location if there is no action code.

C.2.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program to use with this program type. You can use the Quick Start CL Generator to automatically create your CL program.

C.2.4 File Specifications

This program type requires that you define a single master file and a display file. The master file contains 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.

C.2.5 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 necessary. This key list should match your key field definition from the control format of the display file.

C.2.6 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 scans without a key until the subfile fills.

C.2.7 Quick Start Generation

You can generate this program type using the Quick Start CL Generator.

C.3 A0020 - Interactive Single Record Inquiry

C.3.1 Description

Use this program type to create an interactive single record program. This program is for inquiry purposes only. Create a display file prior to generating this program type. This program type processes a single master file by key.

C.3.2 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 SDA with the value K. If you use the Data Base Field Selection feature in SDA, the known key field updates automatically.

The use of an Action Code is optional. Enter a default cursor location if there is no action code. This program type does not use Lockout Action Codes.

C.3.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.3.4 File Specifications

This program type requires that you define a single master file and a display file. The master file contains 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.

C.3.5 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.

C.3.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.4 B0010 - Interactive Single Record Maintenance

C.4.1 Description

Use this program type to create 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.

C.4.2 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 SDA with the value K. If you are using the Data Base Field Selection feature in SDA, the known key field updates automatically.

You must use an Action Code. Lockout Action Codes are optional.

C.4.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program to use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.4.4 File Specifications

This program type requires that you define a single master file and a display file. The master file contains 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.

C.4.5 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 necessary. This key list should match your key field definition from the control format of the display file.

C.4.6 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.5 C0010 - Batch Report with Totals

C.5.1 Description

Use this program type to create a batch report program that you manipulate using DREAM Writer. Create a printer file prior to generating this program type. This program type processes a single master file. You choose the data for the batch report using DREAM Writer Data Selection and Data Sequence parameters. The Batch Report with Totals program type does not use Lockout Action Codes and user defined options and function exits.

C.5.2 Printer File Definition

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

C.5.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program to use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.5.4 File Specifications

This program type requires that you define a single master file and a printer file. The master file contains 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.

C.5.5 Special Considerations

If you are 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, ensure that the value of 2 is in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

C.5.6 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.6 C0020 - Batch Report with Totals and Subheadings

C.6.1 Description

Use this program type to create a batch report program that you manipulate using DREAM Writer. Create a printer file prior to generating this program type. This program type processes a single master file. You choose the data for the print program using DREAM Writer Data Selection and Data Sequence parameters. The Batch Report with Totals and Subheadings program type does not use Lockout Action Codes and user defined options and function exits.

C.6.2 Printer File Definition

This program type requires that formats HEADING1, HEADING2 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

C.6.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.6.4 File Specifications

This program type requires that you define a single master file and a printer file. The master file contains M or 1 in the Input column. The printer file begins with an R and has blank selection columns. Add files to retrieve descriptions if necessary.

C.6.5 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, ensure that the value of 2 is in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

C.6.6 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.7 C0025 - Batch Report with Totals and Subheadings

C.7.1 Description

Use this program type to create a batch report program that you manipulate using DREAM Writer. Create a printer file prior to generating this program type. This program type processes a single master file. You choose the data for the print program using DREAM Writer Data Selection and Data Sequence parameters. The Batch Report with Totals and Subheadings program type does not use Lockout Action Codes and user defined options and function exits.

C.7.2 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.

C.7.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.7.4 File Specifications

This program type requires that you define a single master file and a printer file. The master file contains 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.

C.7.5 Special Considerations

This program type is not a Q&A response in the Program Purpose and Type step. To use this program type, enter an X in 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. When you use this program type, control the page breaks to match the subheadings.

If you are 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, ensure that the value of 2 is in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

C.7.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.8 D0010 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number

C.8.1 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 display. The secondary master file processes by relative record number and controls the database updates.

C.8.2 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 SDA with the value K. If you use the Data Base Field Selection feature in SDA, the known key field update automatically.

You must use an Action Code. Lockout Action Codes are optional.

The system stores the record number of each subfile record in a hidden relative record number field. Add the field SH#RRN to the subfile format with S in the Type field and 9.0 in the Size field, using the Display All Defined Fields in the SDA.

C.8.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.8.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed, and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the primary master file. The secondary master file contains 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.8.5 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. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database. Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After (Y/N) field. This prevents the key field for page up and page down key processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.8.6 Special Considerations

This program type uses the key information in the display file for positioning within the master file. This program type must also have a hidden relative record number field and an entry optional field.

C.8.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.9 D0020 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number

C.9.1 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 display. The secondary master file processes by relative record number and controls the database updates.

C.9.2 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 SDA with the value K. If you use the Data Base Field Selection feature in SDA, the known key field update automatically.

This program type does not use an Action Code. Enter a default cursor location.

The system stores the record number of each subfile record 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 SDA.

C.9.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.9.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the primary master file. The secondary master file contains a 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.9.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.9.6 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.

C.9.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.10 D0030 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record

C.10.1 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. The system bases updates to the subfile on read next change (READC) logic.

C.10.2 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 SDA with the value K. If you use the Data Base Field Selection feature in SDA, the known key fields update automatically.

This program type does not use an Action Code. Enter a default cursor location.

The system stores the record number of each subfile record 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 SDA.

C.10.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.10.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the

primary master file. The secondary master file contains a 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.10.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the update to the database. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.10.6 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.

C.10.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.11 D0040 - Interactive Subfile Maintenance with Action Code, with Options, by Key

C.11.1 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 a single master file by key.

C.11.2 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 SDA, the known key fields update automatically.

You must define Action Codes. Lockout Action Codes are optional.

This subfile maintenance program type allows you to delete individual subfile records using special logic. You perform this by entering C in the Action Code field, comparing the previous value with the current value and then deleting the record if the current value is blank. The system stores the previous value in a hidden field at the subfile record level using the Display All Defined Fields in the SDA.

C.11.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.11.4 File Specifications

This program type requires that you define a single master file and a display file. The master file contains 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.

C.11.5 Detailed Programming Facility

Use a selection 4 to exit to the field details for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

C.11.6 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.

C.11.7 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.12 D0050 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number

C.12.1 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 the system updates it from the fields in the control format of the display file. The secondary master file processes by relative record number and the system update it from the fields in the subfile format of the display file.

C.12.2 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 use the Data Base Field Selection feature in SDA, the known key fields update automatically.

You must define Action Codes. Lockout Action Codes are optional.

The system stores the record number of each subfile record 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 SDA.

C.12.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.12.4 File Specifications

This program type requires that you define a two keyed master files, a secondary master file which is not keyed and a display file. The first master file contains 1 in the Update column. The system updates this file from the control format of the display file. The second master file is a non-keyed file which the subfile format of the display file updates. The second master file contains 2 in the Update column and X under the Add column. Enter N in the Keyed (Y/N) field in the fold area. The third master file is the logical file that the system uses for sequencing records in the subfile. This file contains 3 in the Input column. You enter a value in the File Information Data Structure field in the fold area of this master file. 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.

C.12.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update function. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.12.6 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.

C.12.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.13 D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key

C.13.1 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 a single master file by key.

C.13.2 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 use the Data Base Field Selection feature in SDA, the known key fields update automatically.

You must define Action Codes. Lockout Action Codes are optional.

This subfile maintenance program type allows you to delete individual subfile records using special logic. You perform this logic by entering C in the Action Code, comparing the previous value with the current value and then deleting the record if the current value is blank. The system stores the previous value 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 SDA.

C.13.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.13.4 File Specifications

This program type requires that you define a single master file and a display file. The master file contains an 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.

C.13.5 Detailed Programming Facility

Use a selection 4 to exit to the field details for the SF field controlling the update to the database. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

C.13.6 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.

C.13.7 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.14 D0070 - Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number

C.14.1 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 display. The secondary master file processes by relative record number and controls the database updates.

C.14.2 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 SDA, the known key fields update automatically.

You must define the Action Codes. Lockout Action Codes are optional.

The system stores the record number of each subfile record 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 SDA.

C.14.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.14.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed, and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the primary master file. The secondary master file contains a 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display file begins with a V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.14.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page

down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.14.6 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.

C.14.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.15 D0080 - Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number

C.15.1 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 display. The secondary master file processes by relative record number and controls the database updates.

C.15.2 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 use the Data Base Field Selection feature in SDA, the known key fields update automatically.

Do not use Action Codes. Enter 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 SDA.

C.15.3 CL Program Definition

Copy and revise the model CL program J98MODEL1 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.15.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the primary master file. The secondary master file contains a 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.15.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

If you are creating an inquiry which uses a partial key list from the master file, you must change the Key Sequence field (KY PS) in the Detailed Programming Facility. The KY column is to the right of the master file field names and contains the sequence number for the key fields. Clear all sequence numbers that are not in the key search that you define in the control format of the display file. The key sequence you define in the Detailed Programming Facility should match the key fields in the control format.

C.15.6 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.

C.15.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.16 D0090 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance

C.16.1 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 display. The secondary master file processes by relative record number and controls the database updates. The system verifies all records before it performs any database updates.

C.16.2 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 SDA with the value K. If you use the Data Base Field Selection feature in SDA, the known key fields update automatically.

Do not use Action Codes. Enter 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 SDA.

C.16.3 CL Program Definition

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

C.16.4 File Specifications

This program type requires that you define a keyed master file, a secondary master file which is not keyed, and a display file. The master file contains 1 in the Input column. You enter a value in the File Information Data Structure field in the fold area of the primary master file. The secondary master file contains a 2 in the Update column and N in the Keyed (Y/N) field in the fold area. The display file begins with V and has blank selection columns. Add files to retrieve descriptions if necessary.

C.16.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

Because there are two master files for this program type, you must add special logic to control the page up and page down keys. 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 page up and page down processing for the keyed master file to work incorrectly after the first subfile page fills. To rectify the page up and page down processing, locate 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 enter N in the Clear After field. This prevents the key field for page up and page down processing from clearing.

C.16.6 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 the system to balance transactions or verify all records before it updates the database.

C.16.7 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.17 D0100 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key

C.17.1 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 the system updates it from the fields in the control format of the display file. The secondary master file processes by key and the system updates it from the fields in the subfile format of the display file.

C.17.2 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 use the Data Base Field Selection feature in SDA, the known key fields update automatically.

You must define the Action Codes. Lockout Action Codes are optional.

This program type requires that you define 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 SDA.

C.17.3 CL Program Definition

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

C.17.4 File Specifications

The first master file contains 1 in the Update column. The system updates this file from the control format of the display file. The second master file contains 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.

C.17.5 Detailed Programming Facility

Use a selection exit 4 to display the Detailed Programming Facility for the subfile field controlling the database update. Enter N in the Entry Optional field. Entering N informs the program generator that the user must complete this field before the system updates the database.

C.17.6 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.

C.17.7 Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C.18 E0010 - Interactive Window

C.18.1 Description

Use this program type to create an interactive window program. Create a display file prior to generating this program type. This program type processes a single master file by key.

C.18.2 Display File Definition

SDA 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. SDA 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 precedes the window border.

Do not use Action Codes.

C.18.3 CL Program Definition

A CL program is option for this model.

If you want to create a CL program, copy and revise the model CL program J98MODEL1 to create a CL program to use with this program type. Use the Quick Start CL Generator to automatically create your CL program. The interactive window program type assumes three parameters. Add these to the call statement for your program.

C.18.4 File Specifications

This program type requires the use of a single master file and a display file. The master file contains 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.

C.18.5 Define Option and Function Key Exits

If you use this window to retrieve values to the calling program, add #SSELC to the Function Exit definitions.

C.18.6 Detailed Programming Facility

If used, update all VC0 description fields in the Detailed Programming Facility.

The system uses a key list for record retrieval from the master file. If you are not using the complete key list, update the Key Sequence field (KY PS) in the Detailed Programming Facility to include only those data items which are necessary. This key list should match your key field 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.

C.18.7 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.

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 JD Edwards World SEU or *PROC. JD Edwards World includes two entry points into this subroutine.

The system updates the window key literal in the upper left hand corner of the display file at run time. Modify subroutine S999 through *PROC prior to compiling the RPG program. Assign the video screen name to the work field VC01 using the entry point in subroutine S999.

C.18.8 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.19 X0010 - Batch Update with Report

C.19.1 Description

Use this program type to create a batch update program that you manipulate using DREAM Writer. Create a printer file prior to generating this program type. This program type processes a single master file. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters.

C.19.2 Printer File Definition

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional exist for totals.

C.19.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.19.4 File Specifications

The master file contains M or 1 in the Update column. Remove default value of X in the Add column 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.

C.19.5 Special Considerations

If you are 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, ensure that the value of 2 is in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

This program updates the master file in subroutine S010. You might want to add special logic to control when updates occur.

C.19.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.20 X0020 - Batch Update

C.20.1 Description

Use this program type to create a batch update program that you manipulate using DREAM Writer. This program type processes two master files. The system uses the primary master file to retrieve data from the secondary master file. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. Do not create user defined options and function exits.

C.20.2 Printer File Definition

You do not use a printer file with this program type.

C.20.3 CL Program Definition

Copy and revise the model CL program J98MODEL2 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.20.4 File Specifications

This program type requires that you define a key for both the master file and secondary file. The master file contains 1 in the Input column. The secondary master file contains 2 in the Update column. Add files to retrieve descriptions if necessary.

C.20.5 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. The system does not automatically update key fields in this program type.

This program type contains subroutine S005 for all calculations. Add all special logic code between the read of the primary master file and the update or write of the secondary master file.

This program updates the master file records in subroutine S010.

C.20.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.21 X0030 - Batch Update with Subroutine S001

C.21.1 Description

Use this program type to create a batch update program that you manipulate using DREAM Writer. This program type processes two master files. The system uses the primary master file to retrieve data from the secondary master file. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. Do not create user defined options and function exits.

C.21.2 Printer File Definition

You do not use a printer file with this program type.

C.21.3 CL Program Definition

Copy and revise the model CL program J98MODEL2 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.21.4 File Specifications

This program type requires that you define a key for both the master file and secondary file. The master file contains 1 in the Input column. The secondary master file has 2 in the Update column. Add files to retrieve descriptions if necessary.

C.21.5 Special Considerations

This program type clears the non-key fields from the primary master file between each record the system processes.

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 includes subroutine S005 for all calculations. Add all special logic necessary between the read of the primary master file and the update of the secondary master file.

This program updates the master file records in subroutine S010.

C.21.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.22 X0040 - Batch Update with Report

C.22.1 Description

Use this program type to create a batch update program that you manipulate using DREAM Writer. Create a printer file prior to generating this program type. You should design the printer file to print an audit trail of each record that the system updates. This program type processes two master files. The system interprets the primary

master file and updates the second master file. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. Do not create user defined options and function exits.

C.22.2 Printer File Definition

This program type prints an audit trail for each record that the system saves in the second master file. Formats HEADING1 and DETAIL1 must exist in the printer file. Format TOTAL1 is optional, and you can use this to have the system compute totals for the level breaks that you define in the DREAM Writer Data Sequence.

C.22.3 CL Program Definition

Copy and revise the model CL program J98MODEL2 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.22.4 File Specifications

This program type requires that you define a key for both the master file and secondary file. The master file contains 1 in the Input column. The secondary master file has 2 in the Update column. Add files to retrieve descriptions, if necessary.

C.22.5 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.

This program type uses subroutine S004 to format fields for output to the report. Add any special logic necessary between the read of the primary master file and the update of the second master file.

This program type uses subroutine S005 to scrub and edit the fields in the second master file. Use the Detailed Programming Facility to associate fields in the primary master file with fields in the second master file. Add any special logic that is necessary to compute the proper value that the system loads to the output fields.

Subroutine S010 controls the printing of the report.

Subroutine S011 updates the records in the second master file.

C.22.6 Quick Start Generation

You can not generate this program type using the Quick Start CL Generator.

C.23 Y0010 - Conversion, Two Files with Error Report

C.23.1 Description

Use this program type to create a batch conversion program that you manipulate using DREAM Writer. This program type processes two master files. The system interprets the primary master file and updates the second master file. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. Do not create user defined options and function exits.

C.23.2 Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

C.23.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.23.4 File Specifications

This program type requires that you define a key for both the master file and secondary file. The master file contains 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.

C.23.5 Special Considerations

If you are 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, enter 2 in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

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 includes subroutine S005 for all calculations. Add all special logic necessary between the read of the primary master file and the update of the secondary master file.

This program updates the master file records in subroutine S010.

C.23.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.24 Y0020 - Conversion, One File Update with Error Report

C.24.1 Description

Use this program type to create a batch conversion program that you manipulate using DREAM Writer. This program type processes a single master file by key. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. You do not use Lockout Action Codes, user defined options, and function exits.

C.24.2 Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

C.24.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.24.4 File Specifications

This program type requires that you define a key for a single master file. The master file contains 1 in the Update column. This program type does not update the master file. Clear the X from the Add column, if necessary. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

C.24.5 Special Considerations

If you are 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, enter 2 in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

This program type uses subroutine S005 for all calculations. Add special logic necessary 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.

C.24.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

C.25 Y0030 - Conversion, One File Write with Error Report

C.25.1 Description

Use this program type to create a batch conversion program that you manipulate using DREAM Writer. This program type processes a single master file by key. You choose the data for the batch update program using DREAM Writer Data Selection and Data Sequence parameters. You do not use Lockout Action Codes, user defined options, and function exits.

C.25.2 Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

C.25.3 CL Program Definition

Copy and revise the model CL program J98MODEL6 to create a CL program for use with this program type. Use the Quick Start CL Generator to automatically create your CL program.

C.25.4 File Specifications

This program type requires that you define a key for a single master file. The master file contains 1 in the Update column. As this program type updates the master file, enter X in the Add column. The printer file begins with R and has blank selection columns. Add files to retrieve descriptions if necessary.

C.25.5 Special Considerations

If you are 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, enter 2 in the Type Report Totaling field on the Additional Parameters screen. This allows hierarchal totaling and page breaks in the Data Sequence.

This program type uses subroutine S005 for all calculations. Add special logic necessary 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.

C.25.6 Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

Source Listings

This appendix contains these topics:

- Section D.1, "Program Status Data Structure - I00DSPROG,"
- Section D.2, "Copy Module - Retrieve Soft Coding Data Structure - I00SC,"
- Section D.3, "Item Master Information - P928011."

D.1 Program Status Data Structure - I00DSPROG

Figure D-1 Program Status Data Structure Source Code (1 of 2)

98130	JD Edwards World	Date - 27.01.17
I00DSPROG .JDFSR061	Print Source Code	
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*	08.02.85
7.00	I* Portions of this data structure are loaded at the time the	08.02.85
8.00	I* program is loaded. Other portions of this data structure	08.02.85
9.00	I* are loaded as you perform I/O.	08.02.85
10.00	I*	08.02.85
11.00	I* PURPOSE	08.02.85
12.00	I*	08.02.85
13.00	I* This common subroutine is set up to be used with C0000	08.02.85
14.00	I* (Business Unit Security) common subroutine and C0001(Edit	08.02.85
15.00	I* Action Code) common subroutine. Those two subroutines	08.02.85
16.00	I* will retrieve ##USER for the user name.	08.02.85
17.00	I*	08.02.85
18.00	I* No program calcs are done in this subroutine.	08.02.85
19.00	I*	08.02.85
20.00	I* I##PSDS SDS	18.12.89
21.00	I*	08.02.85
22.00	I* Program Name	08.02.85
23.00	I* Status Code(09999=I/O Error)	08.02.85
24.00	I* Previous Status code	08.02.85
25.00	I* RPG Source Statement Sequence Number	08.02.85
26.00	I* RPG Routine in Which Exception/Error Occured	08.02.85
27.00	I* Number of Parameters Passed to This Program	08.02.85
28.00	I* Exception Type(MCH=Machine, CPF=CPF)	08.02.85
29.00	I* Exception Message Number	08.02.85
30.00	I* Machine Instruction/Object Definition Template Number	08.02.85
31.00	I* Work Area for Messages	08.02.85
32.00	I* Name of Library in Which Program is Located	08.02.85
33.00	I* Retrieved Exception Data. CPF Messages.	08.02.85
34.00	I* Identification of Exception That Caused RPG0001	08.02.85
35.00	I* Unused	08.02.85
36.00	I* Name of File for Last I/O(Only Updated if Error)	08.02.85
37.00	I* Status Info on Last File Used(Only on Error)	08.02.85
38.00	I* Status Code on Last File Used(Only on Error)	08.02.85
39.00	I* Job Name	08.02.85
40.00	I*	08.02.85
41.00	I*	08.02.85
42.00	I*	08.02.85
43.00	I*	08.02.85
44.00	I*	08.02.85
45.00	I*	08.02.85
46.00	I*	08.02.85
47.00	I*	08.02.85
48.00	I*	08.02.85
49.00	I*	08.02.85
50.00	I*	08.02.85
51.00	I*	08.02.85
52.00	I*	08.02.85
53.00	I*	08.02.85
54.00	I*	08.02.85
55.00	I*	08.02.85
56.00	I*	08.02.85

Figure D-2 Program Status Data Structure Source Code (2 of 2)

98330		JD Edwards World		Date - 27.01.17
I00SCPROG	JDFSR61	Print Source Code		
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(MODDTY)		08.02.85
62.00	I*		270 2750##JDT	08.02.85
63.00	I*	Date of Program Execution(MODDTY)		08.02.85
64.00	I*		276 2810##EDT	08.02.85
65.00	I*	Time of Program Execution(MODGMS)		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 ##CVL	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

D.2 Copy Module - Retrieve Soft Coding Data Structure - I00SC

Figure D-3 Copy Module - Retrieve Soft Coding Data Structure report (1 of 7)

98330		JD Edwards World		Date - 27.01.17
I00SC	JDFSR61	Print Source Code		
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 "WINFDS").		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	I00SC	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	1 1 #FEOJ	25.04.88
22.00	I			17.02.88
23.00	I*			25.04.88
24.00	I*	Function - Clear Screen	2 2 #FCLR	25.04.88
25.00	I			17.02.88
26.00	I*			25.04.88
27.00	I*	Function - Help	3 3 #FHELP	25.04.88
28.00	I			17.02.88
29.00	I*			25.04.88
30.00	I*	Function - Values List Display	4 4 #FVLST	25.04.88
31.00	I			17.02.88
32.00	I*			25.04.88
33.00	I*	Function - Roll Up	5 5 #FROLU	25.04.88
34.00	I			17.02.88
35.00	I*			25.04.88
36.00	I*	Function - Roll Down	6 6 #FROLD	25.04.88
37.00	I			17.02.88
38.00	I*			25.04.88
39.00	I*	Function - Window Screen Left	7 7 #FWLFT	25.04.88
40.00	I			17.02.88
41.00	I*			25.04.88
42.00	I*	Function - Window Screen Right	8 8 #FWRGT	25.04.88
43.00	I			17.02.88
44.00	I*			25.04.88
45.00	I*	Function - Question Mark/Cursor Sensitive Help	9 9 #FQMRK	25.04.88
46.00	I			17.02.88
47.00	I*			25.04.88
48.00	I*	Function - Display Error Message(s)	10 10 #FERRD	25.04.88
49.00	I			17.02.88
50.00	I*			25.04.88
51.00	I*	Function - Exit to Address Book	11 11 #FAB	25.04.88
52.00	I			17.02.88
53.00	I*			25.04.88

Figure D-4 Copy Module - Retrieve Soft Coding Data Structure report (2 of 7)

54.00	I*	Function - Exit to Name Search	12 12 #FMS	25.04.88
55.00	I			17.02.88
56.00	I*			25.04.88
98330		JD Edwards World		
I00SC		Print Source Code		
Seq No.				Date - 27.01.17
				Mod Date
57.00	I*	Function - Return to Previous Panel/Menu		25.04.88
58.00	I		13 13 #FFV	17.02.88
59.00	I*			25.04.88
60.00	I*	Function - Display Alternate Panel	14 14 #FALT	25.04.88
61.00	I			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	16 16 #FPM	25.04.88
67.00	I			17.02.88
68.00	I*			25.04.88
69.00	I*	Function - Hard Copy Print		25.04.88
70.00	I		17 17 #FFPT	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#SELC	07.06.88
95.00	I*			25.04.88
96.00	I*	Selection - Change/Revise		25.04.88
97.00	I		35 360#SCING	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#SELT	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
98330		JD Edwards World		
I00SC		Print Source Code		
Seq No.				Date - 27.01.17
				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 JD Edwards World Variables		07.01.91
136.00	I*			22.02.88
137.00	I		81 120 I00SCS	07.01.91
138.00	I*	Future use space, room to grow		25.02.91

Figure D-5 Copy Module - Retrieve Soft Coding Data Structure report (3 of 7)

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 88IFIL	22.02.88
147.00	I*		22.02.88
148.00	I*	Open indication (1=OPEN)	22.02.88
149.00	I	9 9 88OPEN	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 88EOF	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 15088STAT	22.02.88
156.00	I*		22.02.88
157.00	I*	Operation code	22.02.88
158.00	I	16 21 88OPCD	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 88ROUT	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 88SEQN	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 42088RESN	22.02.88
168.00	I*		22.02.88
98330		JD Edwards World	
1005C	.JDFSR61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date
169.00	I*	Record 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 88FRMT	22.02.88
172.00	I*		22.02.88
173.00	I*	Machine OR CFF message number	22.02.88
174.00	I	46 52 88XNO	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 88MT	22.02.88
178.00	I*		22.02.88
179.00	I*	UNUSED	22.02.88
180.00	I	57 80 88FLR1	22.02.88
181.00	I*		22.02.88
182.00	I*	Open data path type (DS=Device DS=Data Base SP=Spool)	22.02.88
183.00	I	81 82 88ODP	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 88FILE	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 88LIBR	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 88SPFM	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 88SPLS	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 124088SPNO	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 126088PRCL	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 128088SRCL	22.02.88
205.00	I*		22.02.88
206.00	I*	Member Name:	22.02.88
207.00	I*	- If ODP type is DS, 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 88MR	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 142088IRIN	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 146088OBIN	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
98330		JD Edwards World	
1005C	.JDFSR61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date

Figure D-6 Copy Module - Retrieve Soft Coding Data Structure report (4 of 7)

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	B 147 14808@DVCL	22.02.88
229.00	I*		22.02.88
230.00	I*	Diskette location(value from 1 to 23 = slot location)	22.02.88
231.00	I	149 151 @CHLC	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 15308@VDSW	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 15508@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 15908@RCMT	22.02.88
241.00	I*		22.02.88
242.00	I*	Access type (only supplied if ODP type is DS)	22.02.88
243.00	I*	KU = Keyed, Unique	22.02.88
244.00	I*	KF = Keyed, FIPO W/Duplicate keys	22.02.88
245.00	I*	KI = Keyed, LIPO 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 @DOUPK	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 @SRCFI	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 @PCBP	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 18508@OVLF	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 18708@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 18908@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 24208@ODFB	22.02.88
279.00	I*		22.02.88
280.00	I*	Put operation count	22.02.88
98330		JD Edwards World	
I00SC	.JDFSR61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date
281.00	I	B 243 24608@PUTC	22.02.88
282.00	I*		22.02.88
283.00	I*	Get operation count	22.02.88
284.00	I	B 247 25008@GETC	22.02.88
285.00	I*		22.02.88
286.00	I*	PutGet operation count	22.02.88
287.00	I	B 251 25408@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 25808@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

Figure D-7 Copy Module - Retrieve Soft Coding Data Structure report (5 of 7)

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' = MFCU	22.02.88
326.00	I*	X'05' = 3411/3410 Tape	22.02.88
327.00	I*	X'06' = 12M 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 @@DNAME	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 283 286@@LLOL	22.02.88
98330		JD Edwards World	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date
337.00	I*		22.02.88
338.00	I*	Routing data information	22.02.88
339.00	I	287 366 @@RDTA	22.02.88
340.00	I*		22.02.88
341.00	I*	Current line number within a printer page	22.02.88
342.00	I	B 367 368@@CLNO	22.02.88
343.00	I*		22.02.88
344.00	I*	AID character indication:	22.02.88
345.00	I*	X'F1' = Enter/Rec Adv	22.02.88
346.00	I*	X'F3' = Roll up	22.02.88
347.00	I*	X'F4' = Roll down	22.02.88
348.00	I*	X'F6' = Print	22.02.88
349.00	I*	X'F8' = Home	22.02.88
350.00	I*	X'BD' = Clear	22.02.88
351.00	I*	X'F3' = Help	22.02.88
352.00	I*	X'3F' = Auto Enter	22.02.88
353.00	I*	X'31' = Command Key 01	09.08.91
354.00	I*	X'32' = Command Key 02	22.02.88
355.00	I*	X'33' = Command Key 03	22.02.88
356.00	I*	X'34' = Command Key 04	22.02.88
357.00	I*	X'35' = Command Key 05	22.02.88
358.00	I*	X'36' = Command Key 06	22.02.88
359.00	I*	X'37' = Command Key 07	22.02.88
360.00	I*	X'38' = Command Key 08	22.02.88
361.00	I*	X'39' = Command Key 09	22.02.88
362.00	I*	X'3A' = Command Key 10	22.02.88
363.00	I*	X'3B' = Command Key 11	22.02.88
364.00	I*	X'3C' = Command Key 12	22.02.88
365.00	I*	X'3D' = Command Key 13	22.02.88
366.00	I*	X'3E' = Command Key 14	22.02.88
367.00	I*	X'3F' = Command Key 15	22.02.88
368.00	I*	X'3A' = Command Key 16	22.02.88
369.00	I*	X'3B' = Command Key 17	22.02.88
370.00	I*	X'3C' = Command Key 18	22.02.88
371.00	I*	X'3D' = Command Key 19	22.02.88
372.00	I*	X'3E' = Command Key 20	22.02.88
373.00	I*	X'3F' = Command Key 21	22.02.88
374.00	I*	X'3A' = Command Key 22	22.02.88
375.00	I*	X'3B' = Command Key 23	22.02.88
376.00	I*	X'3C' = Command Key 24	22.02.88
377.00	I	369 369 @@AID	22.02.88
378.00	I*		22.02.88
379.00	I*	Cursor line in hex (display files only)	22.02.88
380.00	I	370 370 @@CURL	22.02.88
381.00	I*		22.02.88
382.00	I*	Cursor position in hex (display files only)	22.02.88
383.00	I	371 371 @@CURP	22.02.88
384.00	I*		22.02.88
385.00	I*	Note: By simply defining a 2 byte binary field	22.02.88
386.00	I*	and moving the cursor line/position field	22.02.88
387.00	I*	into it right justified you will have the	22.02.88
388.00	I*	numerical value of the line/position field.	22.02.88
389.00	I*	Remember the binary field must be set to	22.02.88
390.00	I*	zero prior to the move.	22.02.88
391.00	I*		22.02.88
392.00	I*	Number of records transmitted	22.02.88
98330		JD Edwards World	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date
393.00	I	B 371 372@@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

Figure D-8 Copy Module - Retrieve Soft Coding Data Structure report (6 of 7)

399.00	I*		B 376 37708@SRN	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		N 378 37908@SRN	22.02.88
403.00	I*			22.02.88
404.00	I*	UNUSED		22.02.88
405.00	I		380 396 @@FLRS	02.10.89
406.00	I*			22.02.88
407.00	I*	RRN of data base record		22.02.88
408.00	I		N 397 40008@SRN	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	I*	IDOCGR DG		30.08.89
417.00	I*			30.08.89
418.00	I*	Returned field name.		30.08.89
419.00	I		1 10 @@FLIN	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 @@RDGC	29.09.89
424.00	I*	Returned location: Row.		31.08.89
425.00	I		71 730@@RCOW	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 @@RDMT	29.09.89
432.00	I*	RPG Indicator Array		29.09.89
433.00	I		97 195 @@TIM	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	I	ISHEN DG		27.11.89
441.00	I		1 1 SHIM01	27.11.89
442.00	I		2 2 SHIM02	27.11.89
443.00	I		3 3 SHIM03	27.11.89
444.00	I		4 4 SHIM04	27.11.89
445.00	I		5 5 SHIM05	27.11.89
446.00	I		6 6 SHIM06	27.11.89
447.00	I		7 7 SHIM07	27.11.89
448.00	I		8 8 SHIM08	27.11.89
98330		J.D Edwards World		
100SC	.JDPSRC61	Print Source Code	Date - 27.01.17	
Seq No.				Mod Date
449.00	I		9 9 SHIM09	27.11.89
450.00	I		10 10 SHIM10	27.11.89
451.00	I		11 11 SHIM11	27.11.89
452.00	I		12 12 SHIM12	27.11.89
453.00	I		13 13 SHIM13	27.11.89
454.00	I		14 14 SHIM14	27.11.89
455.00	I		15 15 SHIM15	27.11.89
456.00	I		16 16 SHIM16	27.11.89
457.00	I		17 17 SHIM17	27.11.89
458.00	I		18 18 SHIM18	27.11.89
459.00	I		19 19 SHIM19	27.11.89
460.00	I		20 20 SHIM20	27.11.89
461.00	I		21 21 SHIM21	27.11.89
462.00	I		22 22 SHIM22	27.11.89
463.00	I		23 23 SHIM23	27.11.89
464.00	I		24 24 SHIM24	27.11.89
465.00	I		25 25 SHIM25	27.11.89
466.00	I		26 26 SHIM26	27.11.89
467.00	I		27 27 SHIM27	27.11.89
468.00	I		28 28 SHIM28	27.11.89
469.00	I		29 29 SHIM29	27.11.89
470.00	I		30 30 SHIM30	27.11.89
471.00	I		31 31 SHIM31	27.11.89
472.00	I		32 32 SHIM32	27.11.89
473.00	I		33 33 SHIM33	27.11.89
474.00	I		34 34 SHIM34	27.11.89
475.00	I		35 35 SHIM35	27.11.89
476.00	I		36 36 SHIM36	27.11.89
477.00	I		37 37 SHIM37	27.11.89
478.00	I		38 38 SHIM38	27.11.89
479.00	I		39 39 SHIM39	27.11.89
480.00	I		40 40 SHIM40	30.11.89
481.00	I		41 41 SHIM41	27.11.89
482.00	I		42 42 SHIM42	27.11.89
483.00	I		43 43 SHIM43	27.11.89
484.00	I		44 44 SHIM44	27.11.89
485.00	I		45 45 SHIM45	27.11.89
486.00	I		46 46 SHIM46	27.11.89
487.00	I		47 47 SHIM47	27.11.89
488.00	I		48 48 SHIM48	27.11.89

Figure D-9 Copy Module - Retrieve Soft Coding Data Structure report (7 of 7)

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			JD Edwards World	
I00SC			Print Source Code	
Seq No.				Date - 27.01.17
				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	I		I00SCDE DS	09.06.93
545.00	I*		Subfile Mode	09.06.93
546.00	I		1 1 #####NO	09.06.93
547.00	I*		Subfile Relative Record Number	09.06.93
548.00	I		2 60####NO	09.06.93
549.00	I*		Cursor Location - Record Format	09.06.93
550.00	I		7 16 ####CFL	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

D.3 Item Master Information - P928011

Figure D-10 Item Master Information report (1 of 32)

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*	This unpublished material is proprietary to
8.00	H*	J. D. Edwards & Company. All rights reserved.
9.00	H*	The methods and techniques described herein are
10.00	H*	considered trade secrets and/or confidential.
11.00	H*	Reproduction or distribution, in whole or in part,
12.00	H*	is forbidden except by express written permission
13.00	H*	of J. D. Edwards & Company.
14.00	H*	
15.00	H*	
16.00	P*	
17.00	P*	PROGRAM REVISION LOG
18.00	P*	
19.00	P*	
20.00	P*	
21.00	P*	
22.00	AUTHRF*	
23.00	P*	
24.00	P*	
25.00	P*	
26.00	P*	
27.00	P*	
28.00	P*	
29.00	P*	
30.00	P*	
31.00	FP001	IF E K DISK
32.00	FP92801	UP E K DISK
33.00	FP928011	CP E K WORKSTN KINPDS SEVERDS
34.00	P*	
35.00	P*	
36.00	P*	Copy Member for Composite Common Subroutine - C0001
37.00	P*	
38.00	P/COPY JDECPY,D0001	
39.00	P*	
40.00	E*	
41.00	E*	PROGRAM TABLES AND ARRAYS
42.00	E*	
43.00	E*	
44.00	E	BMK 64 4 Error Msg
45.00	E	@MK 64 1 Error Msg
46.00	E	@RR 64 4 Error Msg
47.00	E	@DV 40 1 Dflt Wrk
48.00	E	@C 256 1 Literal Work
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 STRUCTURES
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

Figure D-11 Item Master Information report (2 of 32)

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 JDBCFY, IOOCSINK	— Data structure for commonly used indexes	
99.00	I/COPY JDBCFY, IOOPS00	— Data structure used with file servers	
100.00	I/COPY JDBCFY, IOOCSPROG	— Program status data structure	
101.00	I*		
102.00	I*		
103.00	I*		
104.00	I*	Copy Member for Composite Common Subroutine - COOCS	
105.00	I*		
106.00	I/COPY JDBCFY, IOOCS	— Data structure for vocabulary overrides and function keys	
107.00	I*		
108.00	I*		
109.00	I*	Copy Member For Server - X0005	
110.00	I*		
111.00	I/COPY JDBCFY, IOOCSU	— Data structure for file server X0005	
112.00	I*		
113.00	I*		
114.00	I*	Copy Member For Server - X0006	
115.00	I*		
116.00	I/COPY JDBCFY, IOOCS661		
117.00	I*		
118.00	I*		
119.00	I*	Copy Member For Server - X9800E	
120.00	I*		
121.00	I/COPY JDBCFY, 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	— One time only functions
130.00	C*	-----	
131.00	C*		
132.00	C*	If LR on, end program.	
133.00	C*		
134.00	C	*INLR CASEQ'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	24
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 DOWDQ'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.

If information is passed to this program, it will automatically inquire on the record

Figure D-12 Item Master Information report (3 of 32)

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	S998 CASEQ' ' S998		One time only. Pulls in Data Dictionary editing information functions
158.00	C*	-----		
159.00	C	END		
160.00	C*			
161.00	C*	Begin video screen read processing.		
162.00	C			
163.00	C	SETOF		999201
164.00	C	READ V928011		9998
165.00	C	Z-ADDO	###ROW	
166.00	C	Z-ADDO	###COL	Used for cursor sensitive help. Tells where the cursor is.
167.00	C*			
168.00	C*	If video read timed out, end program.		
169.00	C*			
170.00	C	*IN99 CASEQ'1'	ROJ	LR
171.00	C*	-----		
172.00	C	@@AID CASEQ\$PROJ	ROJ	LR
173.00	C*	-----		
174.00	C*			
175.00	C*	If valid function key pressed, process and return.		
176.00	C*			
177.00	C	*IN15 IPRQ '1'		All function keys are assigned indicator 15 so if 15 is on, a function key has been pressed
178.00	C	EXSR S00EX		
179.00	C*	-----		
180.00	C	INLR CASEQ'1'	ROJ	
181.00	C*	-----		
182.00	C	*IN15 CASEQ'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 CASEQ\$PROJ	ROJ	
194.00	C*	---		
195.00	C*			
196.00	C*	If clear screen requested, process and return.		
197.00	C*			
198.00	C	@@AID IPRQ \$PCLR		
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 to load the video/report fields
208.00	C*	-----		
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, validates and edits data
213.00	C*	-----		
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 IPRQ '0'		
219.00	C	*IN24 CASEQ'0'		
220.00	C*	---	S010	Updates files
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*			

Figure D-13 Item Master Information report (4 of 32)

```

229.00 C*      Set correct message in line 24.
230.00 C*
231.00 C
232.00 C      *IN93      IFBQ '1'      VOL24
233.00 C      MOVLSVL24E
234.00 C      ELSE      VDL24
235.00 C      MOVLSVL24M
236.00 C      END
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* 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      @SAID      CARN#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      @SAID      IFBQ #FKEYS
271.00 CSR      CALL 'P9601M'
272.00 C*
273.00 CSR      PARM      IOGSC
274.00 CSR      PARM      SRVPDS
275.00 CSR      PARM      IOOCSR
276.00 C*
277.00 CSR      @SAID      CARN#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      @SAID      IFBQ #PCMRK
287.00 CSR      MOVEA*IN      ##IN
288.00 CSR      CALL 'X96CCK'
289.00 C*
290.00 CSR      PARM      IGOSC
291.00 CSR      PARM      SRVPDS
292.00 CSR      PARM      IOOCSR
293.00 CSR      PARM      ##CCFP 2
294.00 CSR      PARM      IOGMDE
295.00 C*
296.00 CSR      ##PLDN      IFME *BLANKS
297.00 CSR      ELSE SRVVL
298.00 C*
299.00 C*
300.00 CSR      MOVEA##IN      *IN,1
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

Figure D-14 Item Master Information report (5 of 32)

306.00	C*	If Display errors pressed, exit to error messages			
307.00	C*	-----			
308.00	C*				
309.00	CSR	@@AID	IPRQ \$PERRD		
310.00	CSR		Z-ADD1	#G	
311.00	CSR		Z-ADD1	#H	
312.00	CSR	#G	DOWNLE64		
313.00	CSR	@MK,#G	IPRQ '1'		
314.00	CSR		MOVE ENK,#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		FARM	@ER	
322.00	CSR		GOTO ENDEKE		
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	IPRQ \$PHLEP		
330.00	C*		CALL 'POORHELP'		98
331.00	C*		-----		
332.00	CSR		FARM	HS@@	
333.00	CSR		FARM	HE@@	
334.00	CSR		FARM	IOOSC	
335.00	CSR		FARM	SRVPDS	
336.00	CSR		GOTO ENDEKE		
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	IPRQ \$PCLR		
344.00	CSR		EXGR S001		
345.00	C*		-----		
346.00	CSR		GOTO ENDEKE		
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	IPRQ \$PROLU		
354.00	CSR	@@AID	OPRQ \$PROLD		
355.00	C*	\$SECUR	DOUQ' '		
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	IPRQ \$PROLU		
362.00	C*				
363.00	C*	Reset error indicators if roll			
364.00	C*				
365.00	CSR		MOVE\$RESET	*IN,41	
366.00	CSR		MOVE '0'	*IN,40	
367.00	CSR		SETOP		818299
368.00	CSR		READ I92801		9981
369.00	CSR	*IN81	IPRQ '1'		
370.00	CSR	\$RUKEY	SETLLI92801		
371.00	CSR		SETOP		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	IPRQ '1'		
377.00	CSR		SETON		9341
378.00	CSR		MOVE '1'	@MK,2	
379.00	CSR		GOTO ENDEKE		
380.00	C*		-----		
381.00	CSR		END		
382.00	CSR		END		

Figure D-15 Item Master Information report (6 of 32)

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 IPFQ #FROLD		
390.00	C*			
391.00	C*	Reset error indicators if roll		
392.00	C*			
393.00	CSR	MOVBA\$RSET *IN, 41		
394.00	CSR	MOVE '0' *IN, 40		
395.00	CSR	SETOP		818299
396.00	CSR	READPI92801		9981
397.00	CSR	*IN81 IPFQ '1'		
398.00	CSR	\$RDKEY SETLLI92801		8299
399.00	CSR	SETOP		9982
400.00	CSR	READPI92801		
401.00	C*			
402.00	C*	If error on read, set error.		
403.00	C*			
404.00	CSR	*IN82 IPFQ '1'		
405.00	CSR	SETON		9341
406.00	CSR	MOVE '1'	@MX, 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 IPFQ #FROLD		
417.00	CSR	@@AID ORFQ #FROLD		
418.00	C*			
419.00	C*	Release record lock or report record in use.		
420.00	C*			
421.00	CSR	*IN99 IPFQ '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'	@MX, 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	MOVE\$QXKOC	#MCU	
438.00	CSR	#AUT IPNE '1'		
439.00	CSR	#FAUT ANDNE'1'		
440.00	CSR	EXSR C0000		
441.00	C*	-----		
442.00	CSR	END		
443.00	CSR	#AUT IPNE '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 CASSQ' '	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*	-----		
457.00	CSR	END		
458.00	C*			
459.00	CSR	\$SAID IPNE '1'		0193
460.00	CSR	SETON		
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.

Figure D-16 Item Master Information report (7 of 32)

```

466.00 C******
467.00 C*
468.00 C*   Copy Common Subroutine - Coat Center Security Check
469.00 C*
470.00 C/COPY JDECFY,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          SOOVL      EROGR
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          ##RPMT    IFEQ 'V9280111'
490.00 C*
491.00 CSR          ##PLDN    IFEQ 'ACTION'
492.00 CS          MOVE##RVAL      ACTION
493.00 CSR          GOTO ENDOVL
494.00 C*          -----
495.00 CSR          END
496.00 C*
497.00 CSR          ##PLDN    IFEQ 'VDXIT'
498.00 CS          MOVE##RVAL      VDXIT
499.00 CSR          GOTO ENDOVL
500.00 C*          -----
501.00 CSR          END
502.00 C*
503.00 CSR          ##PLDN    IFEQ 'VDXDS'
504.00 CS          MOVE##RVAL      VDXDS
505.00 CSR          GOTO ENDOVL
506.00 C*          -----
507.00 CSR          END
508.00 C*
509.00 CSR          ##PLDN    IFEQ 'VDXCC'
510.00 CS          MOVE##RVAL      VDXCC
511.00 CSR          GOTO ENDOVL
512.00 C*          -----
513.00 CSR          END
514.00 C*
515.00 CSR          ##PLDN    IFEQ 'VDXTY'
516.00 CS          MOVE##RVAL      VDXTY
517.00 CSR          GOTO ENDOVL
518.00 C*          -----
519.00 CSR          END
520.00 C*
521.00 CSR          ##PLDN    IFEQ 'VDXDT'
522.00 CS          MOVE##RVAL      VDXDT
523.00 CSR          GOTO ENDOVL
524.00 C*          -----
525.00 CSR          END
526.00 C*
527.00 CSR          ##PLDN    IFEQ 'VDXQT'
528.00 CS          MOVE##RVAL      VDXQT
529.00 CSR          GOTO ENDOVL
530.00 C*          -----
531.00 CSR          END
532.00 C*
533.00 CSR          ##PLDN    IFEQ 'VDXUM'
534.00 CS          MOVE##RVAL      VDXUM
535.00 CSR          GOTO ENDOVL
536.00 C*          -----
537.00 CSR          END
538.00 C*
539.00 CSR          ##PLDN    IFEQ 'VDX001'
540.00 CS          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.

Figure D-17 Item Master Information report (8 of 32)

543.00	CSR	END		
544.00	C*			
545.00	CSR	##FLDN	IFBQ 'VDX002	'
546.00	CSR		MOVBL##RVAL	VDX002
547.00	CSR		GOTO ENDOVL	
548.00	C*		-----	
549.00	CSR	END		
550.00	C*			
551.00	CSR	#FLDN	IFBQ 'VDX003	'
552.00	CSR		MOVBL##RVAL	VDX003
553.00	CSR		GOTO ENDOVL	
554.00	C*		-----	
555.00	CSR	END		
556.00	C*			
557.00	CSR	##FLDN	IFBQ 'VDX004	'
558.00	CSR		MOVBL##RVAL	VDX004
559.00	CSR		GOTO ENDOVL	
561.00	CSR	END		
562.00	C*			
563.00	CSR	##FLDN	IFBQ 'VDX005	'
564.00	CSR		MOVBL##RVALL	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	BEGR	
581.00	C*	----		
582.00	C*			
583.00	C*	Reset fields for next transaction.		
584.00	C*			
585.00	CSR	NOKEY	CLEARI92801	Clears all the fields in the record format for P92801
586.00	CSR		MOVE *BLANK	###CLP
587.00	CSR		MOVE *BLANK	###CRC
588.00	CSR		Z-ADD*ZERO	###RCOL
589.00	CSR		Z-ADD*ZERO	###RCOM
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		MOVBLSVL24M	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	IFBQ #PCLR	
608.00	CSR		MOVE *ALL' 0'	\$RESET
609.00	CSR		MOVBL\$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	C*		END	
622.00	C*			
623.00	CSR	END001	ENDSR	

Clears all the fields in the record
format for P92801

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.

Figure D-18 Item Master Information report (9 of 32)

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	BRGR		
636.00	C*	----	----		
637.00	C*				
638.00	C*	Load data field dictionary parameters (one cycle only).			
639.00	C*				
640.00	CSR	\$999	CASEQ' ' \$999		
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'	\$RSET 39		
647.00	CSR	MOVE *BLANK	\$RST1 63		
648.00	CSR	MOVE\$RSET	*IN, 41		
649.00	CSR	MOVE\$RST1	AMK, 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	\$NBR09 90		
659.00	CSR	MOVE \$NBR09	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	91	
666.00	CSR	*IN81	DOWEQ'1'		
667.00	CSR		MOVE \$NXXIT	PSIDX 2	
668.00	CSR		CALL 'X0010'	62	
669.00	C*	-----			
670.00	CSR	PARM \$NXXIT	NNSY 4		
671.00	CSR	PARM PSIDX			
672.00	CSR	PARM *ZERO	\$NXTNO 90		
673.00	CSR	MOVE \$NXTNO	QXXIT		
674.00	CSR	MOVE \$NXTNO	VDXIT		
675.00	CSR	QXXIT	SETLLF92801	9281	
676.00	CSR	END			
677.00	CSR	END			
678.00	C*	-----			
679.00	CSR	QXXY01	CHAIN192801	9999	
680.00	C*				
681.00	C*	Cost Center security edit.			
682.00	C*				
683.00	CSR	MOVE/P92801	'\$FILE		
684.00	CSR	MOVE/QXXCC	\$MCO		
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'	\$SSECR 1	
694.00	CSR	END			
695.00	C*				
696.00	C*	If security violation, set error condition.			
697.00	C*				
698.00	CSR	\$SSECR	IFEQ '1'		
699.00	CSR		MOVE '1'	AMK,9	
700.00	CSR		SETON	9341	

Figure D-19 Item Master Information report (10 of 32)

701.00	CSR	MOVE ' '	\$\$\$EPCF 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	93
718.00	CSR	SETON		
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'		91
725.00	CSR	-----		
726.00	CSR	PARM	##PSDS	
727.00	CSR	MOVE '1'	@MK, 6	9341
728.00	CSR	SETON		
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 CABBQ'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'		JDE uses this or SETLL
741.00	CSR	(IN99 ANDBQ'0'		to release record locks
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 CABBQ'1'	END003	
748.00	C*	-----		
749.00	C*	-----		
750.00	C*			
751.00	C*	Move data base information to video screen.		
752.00	C*			Moves information to
753.00	CSR	EXSR S004		the video/report fields
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*	X0020.		

Figure D-20 Item Master Information report (11 of 32)

778.00	C*					
779.00	CSR	S004	REGSR			
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'		91	
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	PSERM 4		
791.00	CSR		PARM	I0006		
792.00	C*					
793.00	CSR		MOVE *BLANK	VC0001		
794.00	CSR	EDATRM	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'		91	
806.00	C*		----			
807.00	CSR		PARM	*0005U		
808.00	CSR		MOVE *BLANK	VC0002		
809.00	CSR	#UERR	IFEQ '0'			
810.00	CSR		MOVELM@UDL01	VC0002		
811.00	CSR		END			
812.00	C*					
813.00	C*	Description display for	- Item Unit of Measure			
814.00	C*					
815.00	CSR		CLEARI0005U			
816.00	CSR		MOVELS@XUM	#USY		
817.00	CSR		MOVE R@XUM	#URT		
818.00	CSR		MOVE QXXUM	#UKY		
819.00	CSR		CALL 'X0005'		91	
820.00	C*		----			
821.00	CSR		PARM	I0005U		
822.00	CSR		MOVE *BLANK	VC0003		
823.00	CSR	#UERR	IFEQ '0'			
824.00	CSR		MOVELM@UDL01	VC0003		
825.00	CSR		END			
826.00	C*					
827.00	C*					
828.00	C*	Description display for	- Item Category Code 001			
829.00	C*					
830.00	CSR		CLEARI0005U			
831.00	CSR		MOVELS@X001	#USY		
832.00	CSR		MOVE R@X001	#URT		
833.00	CSR		MOVE QXX001	#UKY		
834.00	CSR		CALL 'X0005'		91	
835.00	C*		----			
836.00	CSR		PARM	I0005U		
837.00	CSR		MOVE *BLANK	VC0004		
838.00	CSR	#UERR	IFEQ '0'			
839.00	CSR		MOVELM@UDL01	VC0004		
840.00	CSR		END			
841.00	C*					
842.00	C*					
843.00	C*	Description display for	- Item Category Code 002			
844.00	C*					
845.00	CSR		CLEARI0005U			
846.00	CSR		MOVELS@X002	#USY		
847.00	CSR		MOVE R@X002	#URT		
848.00	CSR		MOVE QXX002	#UKY		
849.00	CSR		CALL 'X0005'		91	
850.00	C*		----			
851.00	CSR		PARM	I0005U		
852.00	CSR		MOVE *BLANK	VC0005		
853.00	CSR	#UERR	IFEQ '0'			
854.00	CSR					

File server for user
defined codes

Figure D-21 Item Master Information report (12 of 32)

955.00	CSR	MOVE#UDL01	VC0005	
956.00	CSR	END		
957.00	C*			
958.00	C*			
959.00	C*	Description display for - Item Category Code 003		
960.00	CSR			
961.00	CSR	CLEARI0005U		
962.00	CSR	MOVE#RAX003	#USY	
963.00	CSR	MOVE RAX003	#URT	
964.00	CSR	MOVE QXX003	#UXY	
965.00	C*	CALL 'X0005'		91
966.00	CSR			
967.00	CSR	PARM	I0005U	
968.00	CSR	MOVE *BLANK	VC0006	
969.00	CSR	IFDQ '0'		
970.00	CSR	MOVE#UDL01	VC0005	
971.00	CSR	END		
972.00	C*			
973.00	C*			
974.00	C*	Description display for - Item Category Code 004		
975.00	C*			
976.00	CSR	CLEARI0005U		
977.00	CSR	MOVE#RAX004	#USY	
978.00	CSR	MOVE RAX004	#URT	
979.00	CSR	MOVE QXX004	#UXY	
980.00	C*	CALL 'X0005'		91
981.00	CSR			
982.00	CSR	PARM	I0005U	
983.00	CSR	MOVE *BLANK	VC0007	
984.00	CSR	IFDQ '0'		
985.00	CSR	MOVE#UDL01	VC0007	
986.00	CSR	END		
987.00	C*			
988.00	C*			
989.00	C*	Description display for - Item Category Code 005		
990.00	C*			
991.00	CSR	CLEARI0005U		
992.00	CSR	MOVE#RAX005	#USY	
993.00	CSR	MOVE RAX005	#URT	
994.00	CSR	MOVE QXX005	#UXY	
995.00	C*	CALL 'X0005'		91
996.00	CSR			
997.00	CSR	PARM	I0005U	
998.00	CSR	MOVE *BLANK	VC0008	
999.00	CSR	IFDQ '0'		
1000.00	CSR	MOVE#UDL01	VC0008	
1001.00	CSR	END		
1002.00	C*			
1003.00	C*			
1004.00	C*	Move to output - Cost Center		
1005.00	C*			
1006.00	CSR	MOVE *BLANK	#SINBR	
1007.00	CSR	MOVE#QXXCC	#SINBR	
1008.00	CSR	MOVE T#XCC	#DTYP	
1009.00	CSR	MOVE W#XCC	#EWRD	
1010.00	CSR	MOVE B#XCC	#EC	
1011.00	CSR	MOVE F#XCC	#DEPD	
1012.00	CSR	MOVE G#XCC	#DATD	
1013.00	CSR	MOVE J#XCC	#ALR	
1014.00	CSR	MOVE ' '	#ECOR	
1015.00	CSR	MOVE ' '	#DCOR	
1016.00	CSR	EXSR C00161		
1017.00	CSR			
1018.00	CSR			
1019.00	CSR	#ALR	IFDQ 'L'	
1020.00	CSR		MOVE#SINBR	VDXCC
1021.00	CSR		ELSE	
1022.00	CSR		MOVE #SINBR	VDXCC
1023.00	CSR		END	
1024.00	C*			
1025.00	C*	Move to output - Description		
1026.00	C*			
1027.00	CSR	MOVE#QXXDS	VDXDS	
1028.00	C*			
1029.00	C*			
1030.00	C*	Move to Output - Date Last Ship		
1031.00	C*			

Editing information
retrieved in S998

Copy module to edit field
for use on screen/report

Figure D-22 Item Master Information report (13 of 32)

932.00	CSR	MOVE QXXDT	#SIDAT	6	
933.00	CSR	MOVE *BLANK	#EDAT	8	
934.00	CSR	MOVE *JUL	#FPMT	7	
935.00	CSR	MOVE *SYSVAL	#TFMT	7	
936.00	CSR	MOVE *SYSVAL	#SKP	7	
937.00	CSR	MOVE ' '	\$KRTST	7	
938.00	CSR	CALL 'X0028		81	External program used to edit dates.
939.00	C*	-----			
940.00	CSR	PARM	#SIDAT		
941.00	CSR	PARM	#EDAT		
942.00	CSR	PARM	#FPMT		
943.00	CSR	PARM	#TFMT		
944.00	CSR	PARM	#SKP		
945.00	CSR	PARM	\$KRTST		
946.00	C*	MOVE#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	MOVEQXXIT	#SINBR		
953.00	CSR	MOVE TXXIT	\$DTYP		
954.00	CSR	MOVE WXXIT	\$ENRD		
955.00	CSR	MOVE BXXIT	\$EC		
956.00	CSR	MOVE PXXIT	\$DGPD		
957.00	CSR	MOVE GXXIT	\$DATD		
958.00	CSR	MOVE JXXIT	\$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	MOVE#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	MOVEQXXQT	#SINBR		
974.00	CSR	MOVE TXXQT	\$DTYP		
975.00	CSR	MOVE WXXQT	\$ENRD		
976.00	CSR	MOVE BXXQT	\$EC		
977.00	CSR	MOVE PXXQT	\$DGPD		
978.00	CSR	MOVE GXXQT	\$DATD		
979.00	CSR	MOVE JXXQT	\$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	MOVE#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	MOVEQXXTY	VDXTY		
994.00	C*	-----			
995.00	C*				
996.00	C*	Move to output - Item Unit of Measure			
997.00	C*				
999.00	CSR	MOVEQXXUM	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	MOVEQXX001	#SINBR		
1005.00	CSR	MOVE TXX001	\$DTYP		
1006.00	CSR	MOVE WXX001	\$ENRD		
1007.00	CSR	MOVE BXX001	\$EC		
1009.00	CSR	MOVE GXX001	\$DATD		

Figure D-23 Item Master Information report (14 of 32)

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 LQX002	#SINBR
1026.00	CSR		MOVE T&X002	#DTYP
1027.00	CSR		MOVE W&X002	#KWRD
1028.00	CSR		MOVE B&X002	#EC
1029.00	CSR		MOVE P&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 LQX003	#SINBR
1047.00	CSR		MOVE T&X003	#DTYP
1048.00	CSR		MOVE W&X003	#KWRD
1049.00	CSR		MOVE B&X003	#EC
1050.00	CSR		MOVE P&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 LQX004	#SINBR
1068.00	CSR		MOVE T&X004	#DTYP
1069.00	CSR		MOVE W&X004	#KWRD
1070.00	CSR		MOVE B&X004	#EC
1071.00	CSR		MOVE P&X004	#DSPD
1072.00	CSR		MOVE G&X004	#DATD
1073.00	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*			

Figure D-24 Item Master Information report (15 of 32)

1087.00	CSR	MOVE *BLANK	SSINER	
1088.00	CSR	MOVE LQX005	SSINER	
1089.00	CSR	MOVE TXX005	SDTYP	
1090.00	CSR	MOVE WXX005	SEWRD	
1091.00	CSR	MOVE BXX005	SEC	
1092.00	CSR	MOVE FXX005	SDSPD	
1093.00	CSR	MOVE GXX005	SDATD	
1094.00	CSR	MOVE JXX005	SALR	
1095.00	CSR	MOVE ' '	SECOR	
1096.00	CSR	MOVE ' '	SECOR	
1097.00	CSR	EXSR C00161		
1098.00	C*	-----		
1099.00	CSR	#ALR IFEQ 'L'		
1100.00	CSR	MOVE SSINER	VXX005	
1101.00	CSR	ELSE		
1102.00	CSR	MOVE SSINER	VXX005	
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		Validates and edits data
1114.00	C*	-----		entered by the user
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 X0026.		
1125.00	C*	2. Update data record fields from video.		
1126.00	C*			
1127.00	CSR	S005	SEGR	
1128.00	C*	----		
1129.00	C*			
1130.00	C*			
1131.00	C*			
1132.00	CSR	*IN21 IFEQ '0'		
1133.00	CSR	*IN22 ANDEQ '0'		
1134.00	CSR	GOTO END005		Only performs this
1135.00	C*	-----		subroutine if a record is
1136.00	CSR	END		added or changed
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'	PSMCD 1	
1145.00	CSR	PARM '1'	PSIMCD 1	
1146.00	CSR	PARM VXXCC	PSMCU 12	
1147.00	CSR	PARM *BLANKS	PSERRM 4	
1148.00	CSR	PARM	I0006	
1149.00	C*			
1150.00	CSR	PSERRM		
1151.00	CSR	IFNE *BLANK		4393
1152.00	CSR	SETON		
1153.00	CSR	MOVE PSERRM	BMK,10	
1154.00	CSR	MOVE '1'	BMK,10	
1155.00	CSR	END		
1156.00	C*	MOVE PSMCU	QXXCC	
1157.00	C*	-----		
1158.00	C*			
1159.00	C*	Scrub and edit - Description		
1160.00	CSR	MOVE VXXDS	QXXDS	
1161.00	C*			
1162.00	C*	Set default value - Description		
1163.00	C*			

```

1165.00 CSR QXXDS UFEQ *BLANK
1166.00 CSR DAXDS IFNE *BLANK
1167.00 CSR MOVEADAXDS @DV
1168.00 CSR MOVEADV QXXDS
1169.00 CSR @DV,1 IFEQ ' '
1170.00 CSR MOVE ' ' @DV,1
1171.00 CSR Z-ADD2 $M
1172.00 CSR DOWLE40 $M
1173.00 CSR @DV,$M IFEQ ' '
1174.00 CSR MOVE ' ' @DV,$M
1175.00 CSR RND
1176.00 CSR ADD 1 $M
1177.00 CSR RND
1178.00 CSR MOVEA@DV,2 QXXDS
1179.00 CSR RND
1180.00 CSR RND
1181.00 CSR RND
1182.00 C*
1183.00 C* Edit allowed values - Description
1184.00 C*
1185.00 CSR AAXDS IFEQ *NS
1186.00 CSR QXXDS ANEQ*BLANK
1187.00 CSR MOVE '1' @MK,03
1188.00 CSR SETON
1189.00 CSR RND
1190.00 C*
1191.00 C*
1192.00 C* Scrub and edit - Date Last Ship
1193.00 C*
1194.00 CSR MOVEAVDXDT @NM
1195.00 CSR EXSR C0012
1196.00 C*
1197.00 CSR Z-ADDNUM6
1198.00 CSR MOVE $NUM6
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 $SPFMT 7
1206.00 CSR MOVE$JUL $STPMT 7
1207.00 CSR MOVE$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 $SPFMT
1214.00 CSR PARM $STPMT
1215.00 CSR PARM $SKP
1216.00 CSR PARM $ERTST
1217.00 CSR MOVE$SIDAT QXXDT
1218.00 CSR $ERTST IFNE '1'
1219.00 CSR MOVE '1' @MK,04
1220.00 CSR SETON
1221.00 CSR RND
1222.00 CSR RND
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 PAXIT $DSPD
1231.00 CSR MOVE GAXIT $DATD
1232.00 CSR EXSR C00151
1233.00 C*
1234.00 CSR MOVE $NUMBER QXXIT
1235.00 C*
1236.00 C* Set default value - Item ID
1237.00 C*
1238.00 CSR VDXIT IFEQ *BLANK
1239.00 CSR DAXIT ANEQ*BLANK
1240.00 CSR MOVEADAXIT @NM
1241.00 CSR EXSR C0012

```

Figure D-26 Item Master Information report (17 of 32)

1242.00	C*	-----			
1243.00	CSR	MOVE PAXIT	\$DGPD		
1244.00	CSR	MOVE GAXIT	\$DATD		
1245.00	CSR	EXSR C00151			
1246.00	C*	-----			
1247.00	CSR	MOVE #NUMBER	QXXIT		
1248.00	CSR	END			
1249.00	C*				
1250.00	C*	Edit upper and lower range - Item ID			
1251.00	C*				
1252.00	CSR	LAXIT IFNE *BLANK			
1253.00	CSR	MOVE *BLANK	XAXIT 15		
1254.00	CSR	MOVE '1'	\$ERTST 1		
1255.00	CSR	MOVELQXXIT	XAXIT		
1256.00	CSR	XAXIT IFEG LAXIT			
1257.00	CSR	XAXIT ANDLEUAXIT			
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	SETCN		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	@HM		
1270.00	CSR	EXSR C0012			
1271.00	C*	-----			
1272.00	CSR	MOVE PAXQT	\$DGPD		
1273.00	CSR	MOVE GAXQT	\$DATD		
1274.00	CSR	EXSR C00151			
1275.00	C*	-----			
1276.00	CSR	MOVE #NUMBER	QXXQT		
1277.00	C*				
1278.00	C*	Set default value - Quantity - On Hand		Default value from Data Dictionary	
1279.00	C*				
1280.00	CSR	VDXQT IFEQ *BLANK			
1281.00	CSR	DAXQT ANDNE *BLANK			
1282.00	CSR	MOVEADAXQT	@HM		
1283.00	CSR	EXSR C0012			
1284.00	C*	-----			
1285.00	CSR	MOVE PAXQT	\$DGPD		
1286.00	CSR	MOVE GAXQT	\$DATD		
1287.00	CSR	EXSR C00151			
1288.00	C*	-----			
1289.00	CSR	MOVE #NUMBER	QXXQT		
1290.00	CSR	END			
1291.00	C*				
1292.00	C*	Edit upper and lower range - Quantity - On Hand		Upper and lower ranges from Data Dictionary	
1293.00	C*				
1294.00	CSR	LAXQT IFNE *BLANK			
1295.00	CSR	MOVE *BLANK	XAXQT 15		
1296.00	CSR	MOVE '1'	\$ERTST 1		
1297.00	CSR	MOVELQXXQT	XAXQT		
1298.00	CSR	XAXQT IFEG LAXQT			
1299.00	CSR	XAXQT ANDLEUAXQT			
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	SETCN		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	MOVELVEXTY	QXXTY		
1312.00	C*				
1313.00	C*	Set default value - Item Type			
1314.00	C*				
1315.00	CSR	QXXTY IFEQ *BLANK			
1316.00	CSR	DAXTY IFNE *BLANK			
1317.00	CSR	MOVEADAXTY	@40		
1318.00	CSR	MOVEA@40	QXXTY		

1319.00	CSR	@40,1	IFRQ ''''		
1320.00	CSR		MOVE ' '	@40,1	
1321.00	CSR		Z-ADD2	SM	
1322.00	CSR	SM	DOWLE40		
1323.00	CSR	@40,SM	IFRQ ''''		
1324.00	CSR		MOVE ' '	@40,SM	
1325.00	CSR		END		
1326.00	CSR		ADD 1	SM	
1327.00	CSR		END		
1328.00	CSR		MOVE@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	IPNE *BLANK		
1336.00	CSR	A@XTY	IFRQ '*NB'		
1337.00	CSR	QXXTY	ANDCQ*BLANK		
1338.00	CSR		MOVE '1'	@MK,02	4493
1339.00	CSR		SETON		
1340.00	CSR		ELSE		
1341.00	CSR		MOVEA@XTY	@40	
1342.00	CSR		MOVE *HIVAL	@AV	
1343.00	CSR		EKSR C997		
1344.00	C*		----		
1345.00	CSR		MOVE ' '	\$ERTST 1	
1346.00	CSR		MOVE *BLANK	\$WRK10 10	
1347.00	CSR		MOVELQXXTY	\$WRK10	
1348.00	CSR	@AV,1	IPNE *HIVAL		
1349.00	CSR	\$WRK10	LOKEFUP@AV		81
1350.00	CSR	*IN@1	IFRQ '0'		
1351.00	CSR		MOVE '1'	\$ERTST	
1352.00	CSR		END		
1353.00	CSR	\$ERTST	IFRQ '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	IPNE *BLANK		
1364.00	CSR		MOVE '1'	\$ERTST	
1365.00	CSR	QXXTY	IPGE L@XTY		
1366.00	CSR	QXXTY	ANCL@L@XTY		
1367.00	CSR		MOVE ' '	\$ERTST	
1368.00	CSR		END		
1369.00	CSR	\$ERTST	IFRQ '1'		
1370.00	CSR		MOVE '1'	@MK,07	4493
1371.00	CSR		SETON		
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	IPNE *BLANK		
1378.00	CSR		CLEARZ00005U		
1379.00	C*		MOVELS@XTY	#USY	
1380.00	CSR		MOVE R@XTY	#URT	
1381.00	CSR		MOVE QXXTY	#UNK	
1382.00	CSR		CALL 'X0005'		81
1383.00	C*				
1384.00	CSR		PARM	I0005U	
1385.00	CSR	#UERR	IFRQ '1'		
1386.00	CSR		MOVE '1'	@MK,09	4493
1387.00	CSR		SETON		
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		MOVELVEXUN	QXXUN	
1395.00	C*				

Figure D-28 Item Master Information report (19 of 32)

1396.00	C*	Set default value - Item Unit of Measure			
1397.00	C*				
1398.00	CSR	QXXUM	IPRQ	*BLANK	
1399.00	CSR	EIXUM	IPNE	*BLANK	
1400.00	CSR		MOVEA@XUM	@40	
1401.00	CSR		MOVEA@40	QXXUM	
1402.00	CSR	@40,1	IPRQ	' '	
1403.00	CSR		MOVE	' '	@40,1
1404.00	CSR		Z-ADD2		#M
1405.00	CSR	#M	DOWLE40		
1406.00	C*	@40,#M	IPRQ	' '	
1407.00	CSR		MOVE	' '	@40,#M
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	IPNE	*BLANK	
1419.00	CSR	A@XUM	IPRQ	' *NB'	
1420.00	CSR	QXXUM	ANDRQ	*BLANK	
1421.00	CSR		MOVE	'1'	@MK,03
1422.00	CSR		SETON		4793
1423.00	CSR		ELSE		
1424.00	CSR		MOVEA@XUM	@40	
1425.00	CSR		MOVE	*NIVAL	@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		MOVELQXXUM		\$WRK10
1431.00	CSR	@AV,1	IPNE	*NIVAL	
1432.00	CSR	\$WRK10	LOKUP@AV		81
1433.00	CSR	*IN@I	IPRQ	'0'	
1434.00	CSR		MOVE	'1'	\$ERTST
1435.00	CSR		END		
1436.00	C*	\$ERTST	IPRQ	'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@XUM	IPNE	*BLANK	
1447.00	CSR		MOVE	'1'	\$ERTST
1448.00	CSR	QXXUM	IPRQ	L@XUM	
1449.00	CSR	EXTON	KNO-UOXON		
1450.00	CSR		MOVE	' '	\$ERTST
1451.00	CSR		END		
1452.00	CSR	\$ERTST	IPRQ	'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	IPNE	*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          DAX001      IFNE *BLANK
1483.00 CSR          MOVEADA001      @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          AAX001      IFNE *BLANK
1502.00 CSR          AAX001      IFEQ '*NB'
1503.00 CSR          QXX001      ANDREQ*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          EXSE C997
1510.00 C*
1511.00 CSR          MOVE ' '      $ERTST 1
1512.00 CSR          MOVE *BLANK      $WRK10 10
1513.00 CSR          MOVEQLQXX001      $WRK10
1514.00 CSR          @AV,1      IFNE *HIVAL
1515.00 CSR          $WRK10      LOOKUP@AV      61
1516.00 CSR          *IN@1      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          LAX001      IFNE *BLANK
1530.00 CSR          MOVE '1'      $ERTST
1531.00 CSR          QXX001      IPGE LAX001
1532.00 CSR          QXX001      ANDLEUAX001
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          RAX001      IFNE *BLANK
1544.00 CSR          CLEARIC005U
1545.00 CSR          MOVELSAX001      $USY
1546.00 CSR          MOVE RAX001      $URT
1547.00 CSR          MOVE QXX001      $URY
1548.00 CSR          CALL 'X0005'
1549.00 C*

```

Figure D-30 Item Master Information report (21 of 32)

1550.00	CGR		PARM	I0005U	
1551.00	CGR	SUERK	IFBQ '1'		
1552.00	CGR		MOVE '1'	@MK,09	
1553.00	CGR		SETON		4693
1554.00	CGR		END		
1555.00	CGR		END		
1556.00	C*	-----			
1557.00	C*				
1558.00	C*	Scrub and edit - Item Category Code 002			
1559.00	C*				
1560.00	CGR		MOVELVX002	QXX002	
1561.00	C*				
1562.00	C*	Set default value - Item Category Code 002			
1563.00	C*				
1564.00	CGR	QXX002	IFBQ *BLANK		
1565.00	CGR	DAX002	IFNE *BLANK		
1566.00	CGR		MOVEXD002	@40	
1567.00	CGR		MOVRA@40	QXX002	
1568.00	CGR	@40,1	IFBQ ' '		
1569.00	CGR		MOVE ' '	@40,1	
1570.00	CGR		Z-ADD2	\$M	
1571.00	CGR	\$M	DOWLE40		
1572.00	CGR	@40,\$M	IFBQ ' '		
1573.00	CGR		MOVE ' '	@40,\$M	
1574.00	CGR		END		
1575.00	CGR		ADD 1	\$M	
1576.00	CGR		END		
1577.00	CGR		MOVRA@40,2	QXX002	
1578.00	CGR		END		
1579.00	CGR		END		
1580.00	CGR		END		
1581.00	C*				
1582.00	C*	Edit allowed values - Item Category Code 002			
1583.00	C*				
1584.00	CGR	AAX002	IFNE *BLANK		
1585.00	CGR	AAX002	IFBQ '*NB'		
1586.00	CGR	QXX002	ANDRQ*BLANK		
1587.00	CGR		MOVE '1'	@MK,03	4993
1588.00	CGR		SETON		
1589.00	CGR		ELSE		
1590.00	CGR		MOVRAAX002	@40	
1591.00	CGR		MOVE *HIVAL	@AV	
1592.00	CGR		EXXR C997		
1593.00	C*	---- --			
1594.00	CGR		MOVE ' '	\$ERTST 1	
1595.00	CGR		MOVE *BLANK	\$WRK10 10	
1596.00	CGR		MOVELQXX002	\$WRK10	
1597.00	CGR	@AV,1	IFNE *HIVAL		
1598.00	CGR	\$WPK10	LCKUP@AV		61
1599.00	CGR	*IN\$1	IFBQ '0'		
1600.00	CGR		MOVE '1'	\$ERTST	
1601.00	CGR		END		
1602.00	CGR	\$ERTST	IFBQ '1'		
1603.00	CGR		MOVE '1'	@MK,07	
1604.00	CGR		SETON		4993
1605.00	CGR		END		
1606.00	CGR		END		
1607.00	CGR		END		
1608.00	CGR		END		
1609.00	C*				
1610.00	C*	Edit upper and lower range - Item Category Code 002			
1611.00	C*				
1612.00	CGR	LAX002	IFNE *BLANK		
1613.00	CGR		MOVE '1'	\$ERTST	
1614.00	CGR	QXX002	IFBQ LAX002		
1615.00	CGR	QXX002	ANDLUXAX002		
1616.00	CGR		MOVE ' '	\$ERTST	
1617.00	CGR		END		
1618.00	CGR	\$ERTST	IFBQ '1'		
1619.00	CGR		MOVE '1'	@MK,07	
1620.00	CGR		SETON		4993
1621.00	CGR		END		
1622.00	CGR		END		
1623.00	C*				
1624.00	C*	Edit from User Defined Codes - Item Category Code 002			
1625.00	C*				
1626.00	CGR	RAX002	IFNE *BLANK		

```

1627.00 CSR CLEARI000SU
1628.00 CSR MOVELS#X002 BUSY
1629.00 CSR MOVE R#X002 SURT
1630.00 CSR MOVE QXX002 OUKF
1631.00 CSR CALL 'X0005' 81
1632.00 C* -----
1633.00 CSR PARM I000SU
1634.00 CSR #UERR IPBQ '1'
1635.00 CSR MOVE '1' @MK,09
1636.00 CSR SETON
1637.00 CSR END 4992
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 MOVELVEX003 QXX003
1644.00 C*
1645.00 C* Set default value - Item Category Code 003
1646.00 C*
1647.00 CSR QXX003 IPBQ *BLANK
1648.00 CSR D#X003 IPNE *BLANK
1649.00 CSR MOVEAD#X003 @40
1650.00 CSR MOVEA#40 QXX003
1651.00 CSR @40,1 IPBQ ' ' '
1652.00 CSR MOVE ' ' @40,1
1653.00 CSR Z-ADD2 EM
1654.00 CSR SM DOWLE40
1655.00 CSR @40,SM IPBQ ' ' '
1656.00 CSR MOVE ' ' @40,SM
1657.00 CSR END
1658.00 CSR ADD 1 SM
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 IPNE *BLANK
1668.00 CSR A#X003 IPBQ *NB
1669.00 CSR QXX003 ANDEQ *BLANK
1670.00 CSR MOVE '1' @MK,03
1671.00 CSR SETON
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 MOVELVQXX003 $WRK10
1680.00 CSR @AV,1 IPNE *HIVAL
1681.00 CSR $WRK10 LOOKUP#AV 81
1682.00 CSR *IN#1 IPBQ '0'
1683.00 CSR MOVE '1' $ERTST
1684.00 CSR END
1685.00 CSR $ERTST IPBQ '1'
1686.00 CSR MOVE '1' @MK,07
1687.00 CSR SETON
1688.00 CSR END 5092
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 IPNE *BLANK
1696.00 CSR MOVE '1' $ERTST
1697.00 CSR QXX003 IPBQ L#X003
1698.00 CSR QXX003 ANDLEU#X003
1699.00 CSR MOVE ' ' $ERTST
1700.00 CSR END
1701.00 CSR $ERTST IPBQ '1'
1702.00 CSR MOVE '1' @MK,07
1703.00 CSR SETON 5092

```

Figure D-32 Item Master Information Report (23 of 32)

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	RAX003	IFNE *BLANK		
1710.00	CSR		CLEAR10005U		
1711.00	CSR		MOVELSAX003	#USY	
1712.00	CSR		MOVE RAX003	#URT	
1713.00	CSR		MOVE QXX003	#UKY	
1714.00	CSR		CALL 'X0005'		81
1715.00	C*		-----		
1716.00	CSR		PARM	I0005U	
1717.00	CSR	SUERE	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	DAX004	IFNE *BLANK		
1732.00	CSR		MOVEAAX004	@40	
1733.00	CSR		MOVEA@40	QXX004	
1734.00	CSR	@40,1	IFEQ ' '		
1735.00	CSR		MOVE ' '	@40,1	
1736.00	CSR		Z-ADD2	SM	
1737.00	CSR	SM	DOBLE40		
1738.00	CSR	@40,SM	IFEQ ' '		
1739.00	CSR		MOVE ' '	@40,SM	
1740.00	CSR		END		
1741.00	CSR		ADD 1	SM	
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	AAX004	IFNE *BLANK		
1751.00	CSR	AAX004	IFEQ '*NB'		
1752.00	CSR	QXX004	ANDSQ*BLANK		
1753.00	CSR		MOVE '1'	@MK,03	
1754.00	CSR		SETON		5193
1755.00	CSR		ELSE		
1756.00	CSR		MOVEAAX004	@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	\$WREK10 10	
1762.00	CSR		MOVELQXX004	\$WREK10	
1763.00	CSR	@AV,1	IFNE *HIVAL		
1764.00	CSR	\$WREK10	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	LAX004	IFNE *BLANK		
1779.00	CSR		MOVE '1'	\$ERTST	
1780.00	CSR	QXX004	IFGE LAX004		

Figure D-33 Item Master Information report (24 of 32)

1781.00	CSR	QXX004	ANDLEQ#X004		
1782.00	CSR		MOVE '1'	\$ERTST	
1783.00	CSR		END		
1784.00	CSR	\$ERTST	IFBQ '1'		
1785.00	CSR		MOVE '1'	@MK,07	
1786.00	CSR		SETCN		5193
1787.00	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		CLEARIO005U		
1794.00	CSR		MOVE#R#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	IFBQ '1'		
1801.00	CSR		MOVE '1'	@MK,09	
1802.00	CSR		SETCN		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		MOVE#V#X005	QXX005	
1810.00	C*				
1811.00	C*	Set default value - Item Category Code 005			
1812.00	C*				
1813.00	CSR	QXX005	IFBQ *BLANK		
1814.00	CSR	D#X005	IFNE *BLANK		
1815.00	CSR		MOVE#D#X005	@40	
1816.00	CSR		MOVE#A#40	QXX005	
1817.00	CSR	@40,1	IFBQ ' '		
1818.00	CSR		MOVE ' '	@40,1	
1819.00	CSR		Z-ADD2	#M	
1820.00	CSR	#M	DOW#LE40		
1821.00	CSR	@40, #M	IFBQ ' '		
1822.00	CSR		MOVE ' '	@40,#M	
1823.00	CSR		END		
1824.00	CSR		ADD 1	#M	
1825.00	CSR		END		
1826.00	CSR		MOVE#A#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	IFBQ '*NB'		
1835.00	CSR	QXX005	AND#EQ*BLANK		
1836.00	CSR		MOVE '1'	@MK,03	
1837.00	CSR		SETCN		5293
1838.00	CSR		ELSE		
1839.00	CSR		MOVE#A#X005	@40	
1840.00	CSR		MOVE *HIVAL	@AV	
1841.00	CSR		EXSE C997		
1842.00	C*		-----		
1843.00	CSR		MOVE ' '	\$ERTST 1	
1844.00	CSR		MOVE *BLANK	\$WRK10 10	
1845.00	CSR		MOVE#L#X005	\$WRK10	
1846.00	CSR	@AV,1	IFNE *HIVAL		
1847.00	CSR	\$WRK10	LOCUP#AV		81
1848.00	CSR	*IN#1	IFBQ '0'		
1849.00	CSR		MOVE '1'	\$ERTST	
1850.00	CSR		END		
1851.00	CSR	\$ERTST	IFBQ '1'		
1852.00	CSR		MOVE '1'	@MX,07	
1853.00	CSR		SETCN		5293
1854.00	CSR		END		
1855.00	CSR		END		
1856.00	CSR		END		
1857.00	CSR		END		

Figure D-34 Item Master Information report (25 of 32)

1858.00	C*					
1859.00	C*					
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			5293
1870.00	CSR		END			
1871.00	CSR		END			
1872.00	C*					
1873.00	C*					
1874.00	C*					
1875.00	CSR	R&X005	IFNE *BLANK			
1876.00	CSR		CLEAR10005U			
1877.00	CSR		MOVELS&X005	\$USY		
1878.00	CSR		MOVE R&X005	\$URT		
1879.00	CSR		MOVE QXX005	\$UKY		
1880.00	CSR		CALL 'X0005'			81
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			5293
1886.00	CSR		END			
1887.00	CSR		END			
1888.00	C*					
1889.00	CSR	END005	ENDGE			
1890.00	C*					
1891.00	C*					
1892.00	C*					
1893.00	C*					
1894.00	C/COPY	JDECPY,C00151				
1895.00	C*					
1896.00	C*					
1897.00	C*					
1898.00	C*					
1899.00	C/COPY	JDECPY,C997				
1900.00	C*					
1901.00	C*					
1902.00	C*					
1903.00	C*					
1904.00	C*					
1905.00	C*					
1906.00	C*					
1907.00	C*					
1908.00	CSR	S010	BEGSR			
1909.00	C*					
1910.00	C*					
1911.00	C*					
1912.00	C*					
1913.00	CSR	*IN21	IFEQ '1'			
1914.00	CSR		WRITEI92801			99
1915.00	CSR		END			
1916.00	C*					
1917.00	C*					
1918.00	C*					
1919.00	CSR	*IN22	IFEQ '1'			
1920.00	CSR		UPDATI92801			99
1921.00	CSR		END			
1922.00	C*					
1923.00	C*					
1924.00	C*					
1925.00	CSR	*IN23	IFEQ '1'			
1926.00	CSR		DELETI92801			99
1927.00	CSR		END			
1928.00	C*					

```

1929.00 C*      Clear data field for next transaction
1930.00 C*
1931.00 CSR      MOVE SPCLR      @SAID
1932.00 CSR      EXSR 8001
1933.00 C*
1934.00 CSR      -----
1935.00 CSR      END010      ENDGR
1936.00 C*
1937.00 C*
1938.00 C*      SUBROUTINE S998 - Load dictionary parameters.
1939.00 C*
1940.00 CSR      S998      BDGR
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      PRDTAI
1947.00 CSR      MOVE *XCC'      PRDTAI
1948.00 CSR      CALL 'X9800R'
1949.00 C*
1950.00 CSR      PARM      I9800R
1951.00 CSR      IFREQ '0'
1952.00 CSR      MOVE PRDSCR      B&XCC      40
1953.00 CSR      MOVE PRDTAT      T&XCC      1
1954.00 CSR      MOVE PRFC      E&XCC      1
1955.00 CSR      MOVE PRDTAS      C&XCC      50
1956.00 CSR      MOVE PRDTAD      G&XCC      20
1957.00 CSR      MOVE PRDDEC      P&XCC      1
1958.00 CSR      MOVE PRSY      S&XCC      4
1959.00 CSR      MOVE PRRT      R&XCC      2
1960.00 CSR      MOVE PRDVAL      D&XCC      40
1961.00 CSR      MOVE PRVAL      A&XCC      40
1962.00 CSR      MOVE PRLVAL      L&XCC      40
1963.00 CSR      MOVE PRUVAL      U&XCC      40
1964.00 CSR      MOVE PRDWR      W&XCC      30
1965.00 CSR      MOVE PRLL      J&XCC      1
1966.00 CSR      MOVE PRNNIX      N&XCC      20
1967.00 CSR      S-ADL1      S&XCC      110
1968.00 CSR      MOVE P&XCC      SA
1969.00 CSR      DO SA
1970.00 CSR      MULT 10      S&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      PRDTAI
1978.00 CSR      MOVE *XDS'      PRDTAI
1979.00 CSR      CALL 'X9800R'
1980.00 C*
1981.00 CSR      PARM      I9800R
1982.00 CSR      IFREQ '0'
1983.00 CSR      MOVE PRDSCR      B&XDS      40
1984.00 CSR      MOVE PRDTAT      T&XDS      1
1985.00 CSR      MOVE PRFC      E&XDS      1
1986.00 CSR      MOVE PRDTAS      C&XDS      50
1987.00 CSR      MOVE PRDTAD      G&XDS      20
1988.00 CSR      MOVE PRDDEC      P&XDS      1
1989.00 CSR      MOVE PRSY      S&XDS      4
1990.00 CSR      MOVE PRRT      R&XDS      2
1991.00 CSR      MOVE PRDVAL      D&XDS      40
1992.00 CSR      MOVE PRVAL      A&XDS      40
1993.00 CSR      MOVE PRLVAL      L&XDS      40
1994.00 CSR      MOVE PRUVAL      U&XDS      40
1995.00 CSR      MOVE PRDWR      W&XDS      30
1996.00 CSR      MOVE PRLL      J&XDS      1
1997.00 CSR      MOVE PRNNIX      N&XDS      20
1998.00 CSR      S-1DD1      S&XDS      110
1999.00 CSR      MOVE P&XDS      SA
2000.00 CSR      DO SA
2001.00 CSR      MULT 10      S&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

Figure D-36 Item Master Information report (27 of 32)

2006.00	C*	Dictionary parameters for - Date Last Ship			
2007.00	C*				
2008.00	CSR	MOVE *BLANK	PRDTAI		
2009.00	CSR	MOVE *XDT'	PRDTAI		
2010.00	CSR	CALL 'X9800E'		81	
2011.00	C*	-----			
2012.00	CSR	PARM	I9800E		
2013.00	CSR	IFREQ '0'			
2014.00	CSR	MOVE PRDSCR	8AXDT	40	
2015.00	CSR	MOVE PRDTAT	TAXDT	1	
2016.00	CSR	MOVE PREC	8AXDT	1	
2017.00	CSR	MOVE PRDTAS	8AXDT	50	
2018.00	CSR	MOVE PRDTAD	8AXDT	20	
2019.00	CSR	MOVE PRCDCEC	8AXDT	1	
2020.00	CSR	MOVE PRSY	8AXDT	4	
2021.00	CSR	MOVE PRRT	8AXDT	2	
2022.00	CSR	MOVE PRDVAL	8AXDT	40	
2023.00	CSR	MOVE PRVAL	8AXDT	40	
2024.00	CSR	MOVE PRVAL	8AXDT	40	
2025.00	CSR	MOVE PRVAL	8AXDT	40	
2026.00	CSR	MOVE PRDWR	8AXDT	30	
2027.00	CSR	MOVE PRLE	8AXDT	1	
2028.00	CSR	MOVE PRNNIX	8AXDT	20	
2029.00	CSR	Z-ADD1	8AXDT	110	
2030.00	CSR	MOVE PRXDT	8A		
2031.00	CSR	DO 8A			
2032.00	CSR	MULT 10	8AXDT		
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	PRDTAI		
2040.00	CSR	MOVE *XIT'	PRDTAI		
2041.00	CSR	CALL 'X9800E'		81	
2042.00	C*	-----			
2043.00	CSR	PARM	I9800E		
2044.00	CSR	IFREQ '0'			
2045.00	CSR	MOVE PRDSCR	8AXIT	40	
2046.00	CSR	MOVE PRDTAT	TAXIT	1	
2047.00	CSR	MOVE PREC	8AXIT	1	
2048.00	CSR	MOVE PRDTAS	8AXIT	50	
2049.00	CSR	MOVE PRDTAD	8AXIT	20	
2050.00	CSR	MOVE PRCDCEC	8AXIT	1	
2051.00	CSR	MOVE PRSY	8AXIT	4	
2052.00	CSR	MOVE PRRT	8AXIT	2	
2053.00	CSR	MOVE PRDVAL	IT	40	
2054.00	CSR	MOVE PRVAL	8AXIT	40	
2055.00	CSR	MOVE PRVAL	8AXIT	40	
2056.00	CSR	MOVE PRVAL	8AXIT	40	
2057.00	CSR	MOVE PRDWR	8AXIT	30	
2058.00	CSR	MOVE PRLE	8AXIT	1	
2059.00	CSR	MOVE PRNNIX	8AXIT	20	
2060.00	CSR	Z-1DD1	8AXIT	110	
2061.00	CSR	MOVE PRXIT	8A		
2062.00	CSR	DO 8A			
2063.00	CSR	MULT 10	8AXIT		
2064.00	CSR	END			
2065.00	CSR	END			
2066.00	C*	-----			
2067.00	C*				
2068.00	C*	Dictionary parameters for - gnanity On Hand			
2069.00	C*				
2070.00	CSR	MOVE *BLANK	PRDTAI		
2071.00	CSR	MOVE *XQT'	PRDTAI		
2072.00	CSR	CALL 'X9800E'		81	
2073.00	C*	-----			
2074.00	CSR	PARM	I9800E		
2075.00	CSR	IFREQ '0'			
2076.00	CSR	MOVE PRDSCR	8AXQT	40	
2077.00	CSR	MOVE PRDTAT	TAXQT	1	
2078.00	CSR	MOVE PREC	8AXQT	1	
2079.00	CSR	MOVE PRDTAS	8AXQT	50	
2080.00	CSR	MOVE PRDTAD	8AXQT	20	
2081.00	CSR	MOVE PRCDCEC	8AXQT	1	
2082.00	CSR	MOVE PRSY	8AXQT	4	

Figure D-37 Item Master Information report (28 of 32)

2083.00	CSR	MOVE PRRT	RKQT	2
2084.00	CSR	MOVE PRDVAL	D&KQT	40
2085.00	CSR	MOVE PRVAL	A&KQT	40
2086.00	CSR	MOVE PRLVAL	L&KQT	40
2087.00	CSR	MOVE PRUVAL	U&KQT	40
2088.00	CSR	MOVE PRIDWR	W&KQT	30
2089.00	CSR	MOVE PRLR	J&KQT	1
2090.00	CSR	MOVE PRNNIX	N&KQT	20
2091.00	CSR	Z-ADD1	S&KQT	110
2092.00	CSR	MOVE P&KQT	SA	
2093.00	CSR	DO SA		
2094.00	CSR	MULT 10	S&KQT	
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	PRDTAI	
2102.00	CSR	MOVE 'XTY'	PRDTAI	
2103.00	CSR	CALL 'X9800E'		81
2104.00	C*			
2105.00	CSR	PARM	I9800E	
2106.00	CSR	IFRQ '0'		
2107.00	CSR	MOVE PRDSCR	S&XTY	40
2108.00	CSR	MOVE PRDTAT	T&XTY	1
2109.00	CSR	MOVE PREC	R&XTY	1
2110.00	CSR	MOVE PRDTAS	C&XTY	50
2111.00	CSR	MOVE PRDTAT	G&XTY	20
2112.00	CSR	MOVE PRDDEC	P&XTY	1
2113.00	CSR	MOVE PRSY	S&XTY	4
2114.00	CSR	MOVE PRRT	R&XTY	2
2115.00	CSR	MOVE PRDVAL	D&XTY	40
2116.00	CSR	MOVE PRVAL	A&XTY	40
2117.00	CSR	MOVE PRLVAL	L&XTY	40
2118.00	CSR	MOVE PRUVAL	U&XTY	40
2119.00	CSR	MOVE PRIDWR	W&XTY	30
2120.00	CSR	MOVE PRLR	J&XTY	1
2121.00	CSR	MOVE PRNNIX	N&XTY	20
2122.00	CSR	Z-ADD1	S&XTY	110
2123.00	CSR	MOVE P&XTY	SA	
2124.00	CSR	DO SA		
2125.00	CSR	MULT 10	S&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	PRDTAI	
2133.00	CSR	MOVE 'XUM'	PRDTAI	
2134.00	CSR	CALL 'X9800E'		81
2135.00	C*			
2136.00	CSR	PARM	I9800E	
2137.00	CSR	IFRQ '0'		
2138.00	CSR	MOVE PRDSCR	S&XUM	40
2139.00	CSR	MOVE PRDTAT	T&XUM	1
2140.00	CSR	MOVE PREC	R&XUM	1
2141.00	CSR	MOVE PRDTAS	C&XUM	50
2142.00	CSR	MOVE PRDTAT	G&XUM	20
2143.00	CSR	MOVE PRDDEC	P&XUM	1
2144.00	CSR	MOVE PRSY	S&XUM	4
2145.00	CSR	MOVE PRRT	R&XUM	2
2146.00	CSR	MOVE PRDVAL	D&XUM	40
2147.00	CSR	MOVE PRVAL	A&XUM	40
2148.00	CSR	MOVE PRLVAL	L&XUM	40
2149.00	CSR	MOVE PRUVAL	U&XUM	40
2150.00	CSR	MOVE PRIDWR	W&XUM	30
2151.00	CSR	MOVE PRLR	J&XUM	1
2152.00	CSR	MOVE PRNNIX	N&XUM	20
2153.00	CSR	Z-ADD1	S&XUM	110
2154.00	CSR	MOVE P&XUM	SA	
2155.00	CSR	DO SA		
2156.00	CSR	MULT 10	S&XUM	
2157.00	CSR	END		
2158.00	CSR	END		
2159.00	C*			

Figure D-38 Item Master Information report (29 of 32)

2160.00	C*				
2161.00	C*	Dictionary parameters for - Item Category Code 001			
2162.00	C*				
2163.00	CSR	MOVE *BLANK	PRDTAI		
2164.00	CSR	MOVEL'X001'	PRDTAI		
2165.00	CSR	CALL 'X9800E'		81	
2166.00	C*	-----			
2167.00	CSR	PARM	I9800E		
2168.00	CSR	IFREQ '0'			
2169.00	CSR	PRERR MOVE PRDSCR	B&X001	40	
2170.00	CSR	MOVE PRDTAT	T&X001	1	
2171.00	CSR	MOVE PREC	E&X001	1	
2172.00	CSR	MOVE PRDTAS	C&X001	50	
2173.00	CSR	MOVE PRDTAD	G&X001	20	
2174.00	CSR	MOVE PRCDCEC	F&X001	1	
2175.00	CSR	MOVEPRST	S&X001	4	
2176.00	CSR	MOVE PRRT	R&X001	2	
2177.00	CSR	MOVE PRDVAL	D&X001	40	
2178.00	CSR	MOVE PRVAL	A&X001	40	
2179.00	CSR	MOVE PRVAL	L&X001	40	
2180.00	CSR	MOVE PRDVAL	U&X001	40	
2181.00	CSR	MOVE PRDWR	W&X001	20	
2182.00	CSR	MOVE PRLE	J&X001	1	
2183.00	CSR	MOVE PRNNIX	N&X001	20	
2184.00	CSR	Z-ADDI	#&X001	110	
2185.00	CSR	MOVE P&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	PRDTAI		
2195.00	CSR	MOVEL'X002'	PRDTAI		
2196.00	CSR	CALL 'X9800E'		81	
2197.00	C*	-----			
2198.00	CSR	PARM	I9800E		
2199.00	CSR	IFREQ '0'			
2200.00	CSR	PRERR MOVE PRDSCR	B&X002	40	
2201.00	CSR	MOVE PRDTAT	T&X002	1	
2202.00	CSR	MOVE PREC	E&X002	1	
2203.00	CSR	MOVE PRDTAS	C&X002	50	
2204.00	CSR	MOVE PRDTAD	G&X002	20	
2205.00	CSR	MOVE PRCDCEC	F&X002	1	
2206.00	CSR	MOVEPRST	S&X002	4	
2207.00	CSR	MOVE PRRT	R&X002	2	
2208.00	CSR	MOVE PRDVAL	D&X002	40	
2209.00	CSR	MOVE PRVAL	A&X002	40	
2210.00	CSR	MOVE PRVAL	L&X002	40	
2211.00	CSR	MOVE PRDVAL	U&X002	40	
2212.00	CSR	MOVE PRDWR	W&X002	20	
2213.00	CSR	MOVE PRLE	J&X002	1	
2214.00	CSR	MOVE PRNNIX	N&X002	20	
2215.00	CSR	Z-ADDI	#&X002	110	
2216.00	CSR	MOVE P&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	PRDTAI		
2226.00	CSR	MOVEL'X003'	PRDTAI		
2227.00	CSR	CALL 'X9800E'		81	
2228.00	C*	-----			
2229.00	CSR	PARM	I9800E		
2230.00	CSR	IFREQ '0'			
2231.00	CSR	PRERR MOVE PRDSCR	B&X003	40	
2232.00	CSR	MOVE PRDTAT	T&X003	1	
2233.00	CSR	MOVE PREC	E&X003	1	
2234.00	CSR	MOVE PRDTAS	C&X003	50	
2235.00	CSR	MOVE PRDTAD	G&X003	20	
2236.00	CSR	MOVE PRCDCEC	F&X003	1	

Figure D-39 Item Master Information report (30 of 32)

2237.00	CSR	MOVELPRSY	SAX003	4
2238.00	CSR	MOVE FRRT	RAX003	2
2239.00	CSR	MOVE PRDVAL	DAX003	40
2240.00	CSR	MOVE PRVAL	AXX003	40
2241.00	CSR	MOVE PRVAL	LAX003	40
2242.00	CSR	MOVE PRDVAL	UAX003	40
2243.00	CSR	MOVE PRDWR	WAX003	30
2244.00	CSR	MOVE FRLR	JAX003	1
2245.00	CSR	MOVE PRNNIX	NAX003	20
2246.00	CSR	Z-ADD1	SAX003	110
2247.00	CSR	MOVE PAX003	SA	
2248.00	CSR	DO SA		
2249.00	CSR	MULT 10	SAX003	
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	PRDTAI	
2257.00	CSR	MOVEL'X004'	PRDTAI	
2258.00	CSR	CALL 'X9800E'		81
2259.00	C*			
2260.00	CSR	PARM	I9800E	
2261.00	CSR	IPRQ '0'		
2262.00	CSR	MOVE PRDSCR	SAX004	40
2263.00	CSR	MOVE PRDTAT	TAX004	1
2264.00	CSR	MOVE PREC	SAX004	1
2265.00	CSR	MOVE PRDTAS	CAX004	50
2266.00	CSR	MOVE PRDTAD	GAX004	20
2267.00	CSR	MOVE PRCDCEC	PAX004	1
2268.00	CSR	MOVELPRSY	SAX004	4
2269.00	CSR	MOVE FRRT	RAX004	2
2270.00	CSR	MOVE PRDVAL	DAX004	40
2271.00	CSR	MOVE PRVAL	AXX004	40
2272.00	CSR	MOVE PRVAL	LAX004	40
2273.00	CSR	MOVE PRDVAL	UAX004	40
2274.00	CSR	MOVE PRDWR	WAX004	30
2275.00	CSR	MOVE FRLR	JAX004	1
2276.00	CSR	MOVE PRNNIX	NAX004	20
2277.00	CSR	Z-ADD1	SAX004	110
2278.00	CSR	MOVE PAX004	SA	
2279.00	CSR	DO SA		
2280.00	CSR	MULT 10	SAX004	
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	PRDTAI	
2288.00	CSR	MOVEL'X005'	PRDTAI	
2289.00	CSR	CALL 'X9800E'		81
2290.00	C*			
2291.00	CSR	PARM	I9800E	
2292.00	CSR	IPRQ '0'		
2293.00	CSR	MOVE PRDSCR	SAX005	40
2294.00	CSR	MOVE PRDTAT	TAX005	1
2295.00	CSR	MOVE PREC	SAX005	1
2296.00	CSR	MOVE PRDTAS	CAX005	50
2297.00	CSR	MOVE PRDTAD	GAX005	20
2298.00	CSR	MOVE PRCDCEC	PAX005	1
2299.00	CSR	MOVELPRSY	SAX005	4
2300.00	CSR	MOVE FRRT	RAX005	2
2301.00	CSR	MOVE PRDVAL	DAX005	40
2302.00	CSR	MOVE PRVAL	AXX005	40
2303.00	CSR	MOVE PRVAL	LAX005	40
2304.00	CSR	MOVE PRDVAL	UAX005	40
2305.00	CSR	MOVE PRDWR	WAX005	30
2306.00	CSR	MOVE FRLR	JAX005	1
2307.00	CSR	MOVE PRNNIX	NAX005	20
2308.00	CSR	Z-ADD1	SAX005	110
2309.00	CSR	MOVE PAX005	SA	
2310.00	CSR	DO SA		
2311.00	CSR	MULT 10	SAX005	
2312.00	CSR	END		
2313.00	CSR	END		

Figure D-40 Item Master Information report (31 of 32)

2314.00	C*								
2315.00	C*								
2316.00	C*	Set subroutine execution flag.							
2317.00	C*								
2318.00	CSR		MOVE '1'	\$998	1			Assures S998 will only be executed once	
2319.00	C*								
2320.00	CSR	ENDS998	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 for unauthorized access, center video title and save to video screen.						
2328.00	C*								
2329.00	C*	3.	Initialize key list.						
2330.00	C*	4.	Load roll keys.						
2331.00	C*	5.	Passed parameters.						
2332.00	C*	6.	Load error message array.						
2333.00	C*								
2334.00	C*								
2335.00	CSR	S999	BEGR						
2336.00	C*	----							
2337.00	C*								
2338.00	C*	Required program parameters.							
2339.00	C*								
2340.00	CSR	*ENTRY	PLIST					Parameters passed to program	
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			Set auto-inquiry if information is passed	
2354.00	CSR		END						
2355.00	C*	-----							
2356.00	C*								
2357.00	C*	Load video screen text						Retrieves vocabulary overrides	
2358.00	C*								
2359.00	CSR		MOVRL\$FILE	PSKEY	10				
2360.00	CSR		Z-ADD025	PSVTX#	30			Only loads these VTX fields displayed on the video instead of all 144	
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					Composite keys are defined here	
2367.00	CSR		KPLD	MSUSER					
2368.00	CSR		KPLD	MSFILE					
2369.00	CSR		KPLD	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		KPLD	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	DEPM QXXIT	\$RUKEY				Using *LIKE more and more, especially for work fields.	
2381.00	CSR	*LIKE	DEPN \$RUKEY	\$RUKEY					
2382.00	CSR		MOVE *LOVAL	\$RUKEY					
2383.00	CSR		MOVE *ALL'9'	\$RUKEY					
2384.00	C*	-----							
2385.00	C*								

Figure D-41 Item Master Information report (32 of 32)

2386.00	C*	Load error messages array.					
2387.00	C*						
2388.00	CSR	MOVE	'0001	EMK,01	Inv Action	Error message numbers from Data Dictionary	
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*					Lockout action code function used with the Program Generator	
2402.00	CSR	MOVEA'	'	ANAC			
2403.00	C*	-----					
2404.00	C*						
2405.00	Ct	Load system date.					
2406.00	C*					Use the TIME feature to allow for all date formats	
2407.00	CSR	TIME	\$WRK12	120			
2408.00	CSR	MOVE	\$WRK12	\$SEDIT	60		
2409.00	CSR	MOVE	\$SEDIT	\$SIDAT	6		
2410.00	CSR	MOVE	'SYSVAL	'\$PFMT	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		\$PFMT			
2420.00	CSR	PARM		\$TFMT			
2421.00	CSR	PARM		\$SKP			
2422.00	CSR	PARM		\$ERTST			
2423.00	CSR	MOVE	\$SIDAT	\$SUPMJ	60		
2424.00	C*	-----					
2425.00	CSR	END999	ENDSR				
2426.00	C*	*****					
2427.00	C*	*****					
2428.00		0192801	E	UNLOCK		Method of releasing master file record locks	

JD Edwards World Subroutines and Flows

This appendix contains these topics:

- [Section E.1, "Subroutines,"](#)
- [Section E.2, "Flows."](#)

E.1 Subroutines

Using subroutines:

- Allows for standard names to make program maintenance easier.
- Launch primarily from Mainline.

The following table includes internal RPG subroutines within JD Edwards World programs:

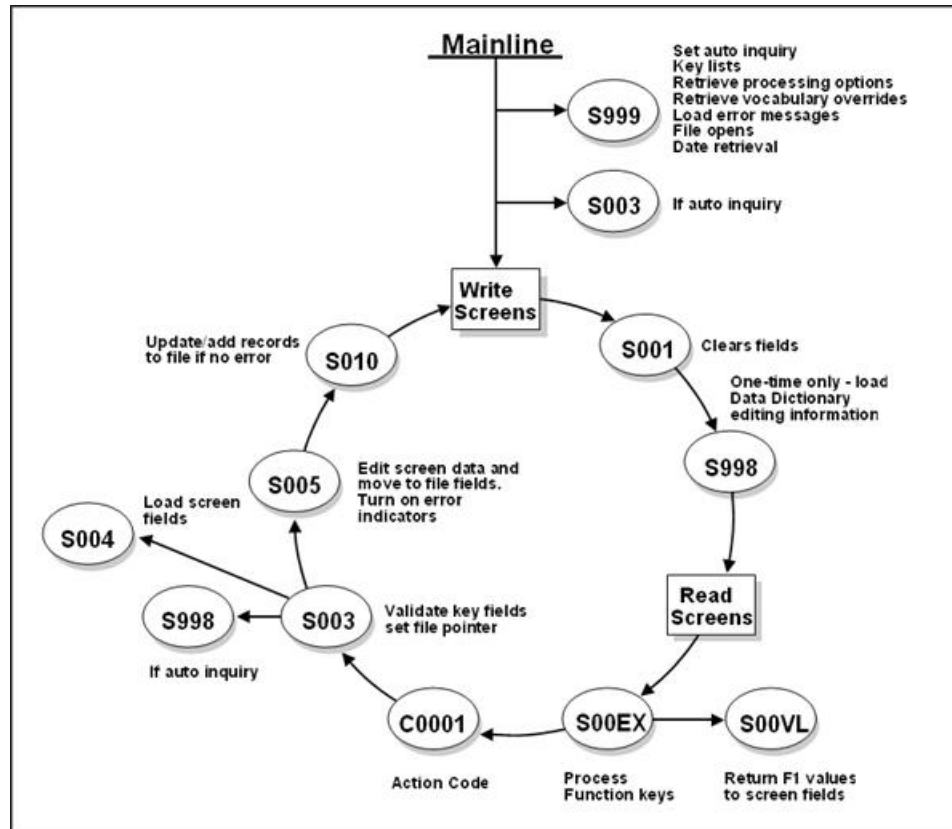
Subroutine	Description
S00EX	Processes all function exits. Choose a function exit and the system launches one of the following programs: <ul style="list-style-type: none">■ Display Functions (F24) to launch P9601H■ Field Sensitive Help (F1) launches X96CCX. After X96CCX launches, the system launches subroutine S00VL.■ Display Error Message (F7) launches P0000E■ HELP launches P00HELP■ Clear Screen (F22) launches subroutine S001■ Launches all programs to process all user defined function keys
S00VL	Retrieves values with Field Level Help. After X96CCX launches, the system launches subroutine S00VL.
S00OP	Subfile Options.
S001	Clears all database and screen fields. This usually only clears key fields and VC0 fields if you choose Clear Screen (F22).
S002	Checks for level breaks for reports. <ul style="list-style-type: none">■ Activates level break markers.■ Retrieves the total line description

Subroutine	Description
S003	<p>Validates the key fields.</p> <p>Launches subroutine S998 if the system invokes auto inquire.</p> <p>Sets the file pointer.</p> <ul style="list-style-type: none"> ■ Performs a SETLL and CHAIN if the program is a single record maintenance program ■ Performs a SETLL for subfile programs <p>Launches subroutine S004 to load screen and report fields</p> <p>Monitors that subfile records load if this is a subfile</p> <p>Loads subfile records that the system does not use with blanks</p>
S004	Display and load the screen or report fields.
S005	<p>Scrubs and edits screen and report fields.</p> <ul style="list-style-type: none"> ■ Moves screen data to database fields ■ Activates error indicators if a field value is in error ■ Updates records in the database file if this is a subfile ■ Updates the subfile
S010	<p>For reports with level breaks it:</p> <ul style="list-style-type: none"> ■ Prints the total ■ Clears the level break totals ■ Prints the grand total when it reaches the end of the file ■ Prints the detail ■ Adds to the new level break totals <p>Launches subroutine S020 if this is a report with subheadings</p> <p>If this is not a report, S010 updates, adds, or deletes records from the database file.</p> <p>Deactivates the Clear Screen (F22) function and executes S001 to clear the buffer before reading another record.</p>
S020	Print Report Subheadings.
S998	<p>Loads Data Dictionary values, one time only.</p> <p>Retrieves row description for level breaks and subheadings, if applicable.</p>
S999	<p>Housekeeping, one time only.</p> <ul style="list-style-type: none"> ■ Sets auto inquiry ■ Defines key lists ■ Retrieves processing options and level breaks, if applicable ■ Retrieves vocabulary overrides ■ Loads error messages ■ Opens file ■ Retrieves the current date ■ Defines work fields using *LIKE ■ Prints cover page and Helps in a report

E.2 Flows

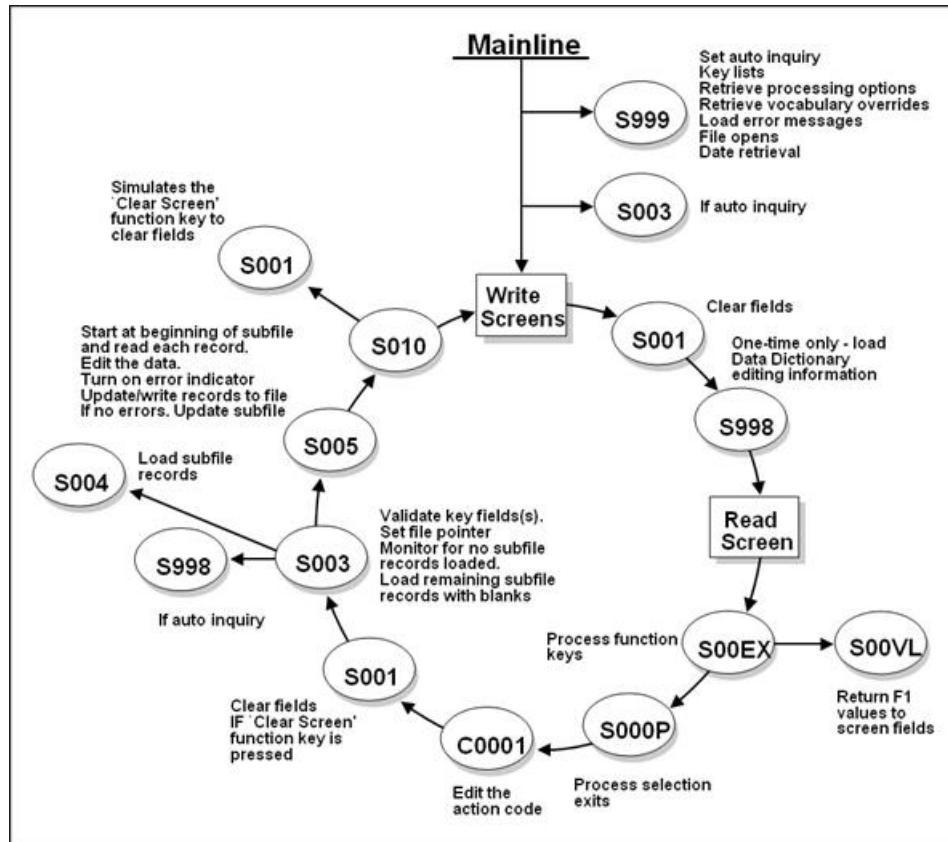
E.2.1 Interactive Non-Subfile Program

Figure E-1 Interactive Non-Subfile Program flow



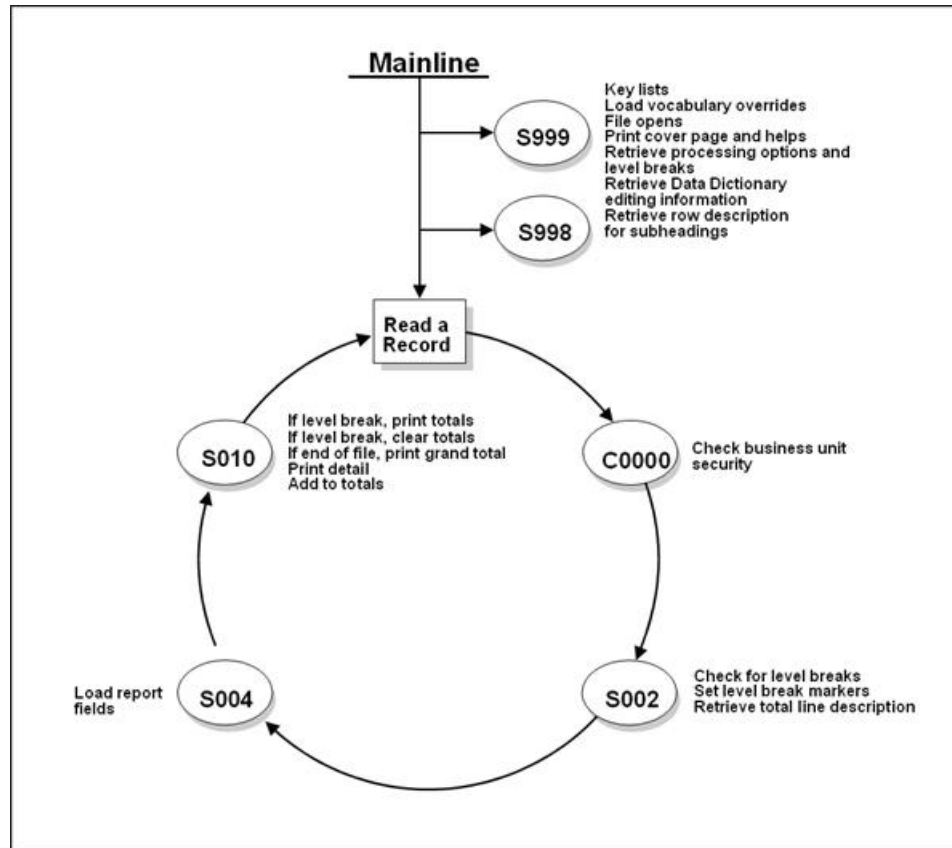
E.2.2 Subfile Program with Options

Figure E-2 Subfile Program With Options flow



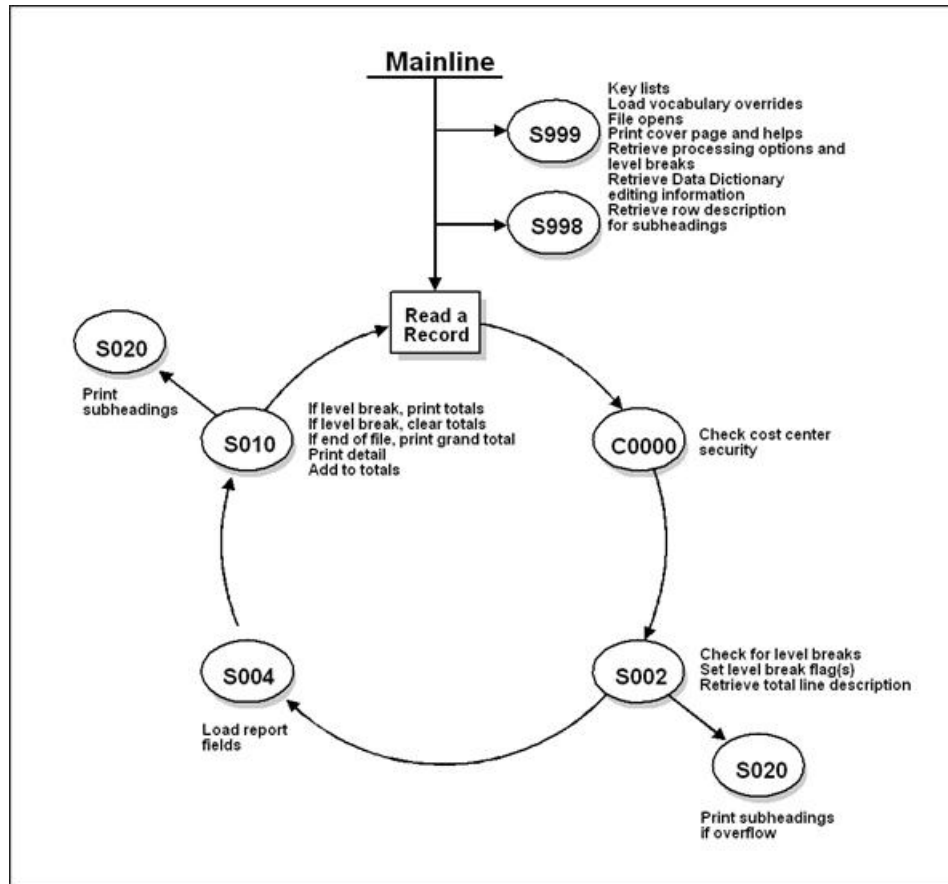
E.2.3 Report Program without Subheadings

Figure E-3 Report Program Without Subheadings flow



E.2.4 Report Program with Subheadings

Figure E-4 Report Program With Subheadings flow



Sample Code

Following is the code to create the basic shell for program type B0010.

Figure F-1 Create/Modify (Basic Shell) screen

93001 Create/Modify

ORACLE JD Edwards World

Field Sensitive Help
Display Error Message
Display Functions
Exit Program
Clear Screen

Action Code: 1

Program Type: B0010

Display	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	SUBR-1	Exits Subroutine - STD Video
6,00	SUBOP	Options Subroutine
6,50	SUBVL-1	Return Values Subr - Standard
7,00	SUB1-1	Clear Subroutine - STD Video
8,00	SUB3-1	Edit Key - STD Video
9,00	SUB4-1	Load Display Subr - STD Video
10,00	SUB5-1	Edit Subroutine - STD Video
11,00	SUB10-1	Update Subroutine - STD Video
12,00	SUB99-1	Housekeeping Subr - STD Video

F24=More

Figure F-2 Program Code for Program Type B0010 (1 of 11)

R03950 B0010 - STD/M - Action Code				DATE - 2/02/17
TITLE/TITLE				
N*				FILEDEFN01 001000000000
N*				FILEDEFN01 002000000000
N*				FILEDEFN01 003000000000
N*	Copyright (c) 2007			FILEDEFN01 004000000000
N*	JD Edwards World			FILEDEFN01 005000000000
N*				FILEDEFN01 006000000000
N*	This unpublished material is proprietary to			FILEDEFN01 007000000000
N*	JD Edwards World. All rights reserved.			FILEDEFN01 008000000000
N*	The methods and techniques described herein are			FILEDEFN01 009000000000
N*	considered trade secrets and/or confidential.			FILEDEFN01 010000000000
N*	Reproduction or distribution, in whole or in part,			FILEDEFN01 011000000000
N*	is forbidden except by express written permission			FILEDEFN01 012000000000
N*	of JD Edwards World.			FILEDEFN01 013000000000
N*				FILEDEFN01 014000000000
N*				FILEDEFN01 015000000000
F*				FILEDEFN01 016000000000
F*	PROGRAM REVISION LOG			FILEDEFN01 017000000000
F*				FILEDEFN01 018000000000
F*				FILEDEFN01 019000000000
F*	Date	Programmer	Nature of Revision	FILEDEFN01 020000000000
F*				FILEDEFN01 021000000000
AUTHOR*		SAR #	(AS/400 A/G)	FILEDEFN01 022000000000
F*				FILEDEFN01 023000000000
DESC F*				FILEDEFN01 024000000000
F*				FILEDEFN01 025000000000
F*				FILEDEFN01 026000000000
F*				FILEDEFN01 027000000000
FILES*				FILEDEFN01 028000000000
COPY F*				FILEDEFN01 029000000000
E*				FILEEXTN0 001000000000
E*	PROGRAM TABLES AND ARRAYS			FILEEXTN0 002000000000
E*				FILEEXTN0 003000000000
E*				FILEEXTN0 004000000000
E		EMC	64 4	FILEEXTN0 005000000000
E		EMC	64 1	FILEEXTN0 006000000000
			Error Msg	
			Error Msg	

Figure F-3 Program Code for Program Type B0010 (2 of 11)

E	8ER	64	4	Error Msg	FILEXTN0	007000000000
E	8DV	40	1	Deflt Wk	FILEXTN0	008000000000
E	8C	256	1	Literal Work	FILEXTN0	011000000000
COPY	E*				FILEXTN0	012000000000
I*					INPUT1	001000000000
I*	PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES				INPUT1	002000000000
I*					INPUT1	003000000000
I*					INPUT1	004000000000
I*	Data Structure to Load Video Screen Text				INPUT1	005000000000
I*					INPUT1	006000000000
VTS	IDSTXT	DS			INPUT1	007000000000
VTX	I		1	40 VTX001	INPUT1	008000000000
VTX	I		41	80 VTX002	INPUT1	009000000000
VTX	I		81	120 VTX003	INPUT1	010000000000
VTX	I		121	160 VTX004	INPUT1	011000000000
VTX	I		161	200 VTX005	INPUT1	012000000000
VTX	I		201	240 VTX006	INPUT1	013000000000
VTX	I		241	280 VTX007	INPUT1	014000000000
VTX	I		281	320 VTX008	INPUT1	015000000000
VTX	I		321	360 VTX009	INPUT1	016000000000
VTX	I		361	400 VTX010	INPUT1	017000000000
VTX	I		401	440 VTX011	INPUT1	018000000000
VTX	I		441	480 VTX012	INPUT1	019000000000
VTX	I		481	520 VTX013	INPUT1	020000000000
R03950		B0010	- STD/M	- Action Code	DATE	2/02/17
VTX	I		521	560 VTX014	INPUT1	021000000000
VTX	I		561	600 VTX015	INPUT1	022000000000
VTX	I		601	640 VTX016	INPUT1	023000000000
VTX	I		641	680 VTX017	INPUT1	024000000000
VTX	I		681	720 VTX018	INPUT1	025000000000
VTX	I		721	760 VTX019	INPUT1	026000000000
VTX	I		761	800 VTX020	INPUT1	027000000000
VTX	I		801	840 VTX021	INPUT1	028000000000
VTX	I		841	880 VTX022	INPUT1	029000000000
VTX	I		881	920 VTX023	INPUT1	030000000000
VTX	I		921	960 VTX024	INPUT1	031000000000
VTX	I		961	1000 VTX025	INPUT1	032000000000
VTX	I		1001	1040 VTX026	INPUT1	033000000000
VTX	I		1041	1080 VTX027	INPUT1	034000000000
VTX	I		1081	1120 VTX028	INPUT1	035000000000
VTX	I		1121	1160 VTX029	INPUT1	036000000000
VTX	I		1161	1200 VTX030	INPUT1	037000000000
VTX	I		1201	1240 VTX031	INPUT1	038000000000
VTX	I		1241	1280 VTX032	INPUT1	039000000000
VTX	I		1281	1320 VTX033	INPUT1	040000000000
VTX	I		1321	1360 VTX034	INPUT1	041000000000
VTX	I		1361	1400 VTX035	INPUT1	042000000000
VTX	I		1401	1440 VTX036	INPUT1	043000000000
VTX	I		1441	1480 VTX037	INPUT1	044000000000
VTX	I		1481	1520 VTX038	INPUT1	045000000000
VTX	I		1521	1560 VTX039	INPUT1	046000000000
VTX	I		1561	1600 VTX040	INPUT1	047000000000
VTX	I		1601	1640 VTX041	INPUT1	048000000000
VTX	I		1641	1680 VTX042	INPUT1	049000000000
VTX	I		1681	1720 VTX043	INPUT1	050000000000
VTX	I		1721	1760 VTX044	INPUT1	051000000000
VTX	I		1761	1800 VTX045	INPUT1	052000000000
VTX	I		1801	1840 VTX046	INPUT1	053000000000
VTX	I		1841	1880 VTX047	INPUT1	054000000000
VTX	I		1881	1920 VTX048	INPUT1	055000000000
VTX	I		1921	1960 VTX049	INPUT1	056000000000
VTX	I		1961	2000 VTX050	INPUT1	057000000000
VTX	I		2001	2040 VTX051	INPUT1	058000000000
VTX	I		2041	2080 VTX052	INPUT1	059000000000
VTX	I		2081	2120 VTX053	INPUT1	060000000000
VTX	I		2121	2160 VTX054	INPUT1	061000000000
VTX	I		2161	2200 VTX055	INPUT1	062000000000
VTX	I		2201	2240 VTX056	INPUT1	063000000000
VTX	I		2241	2280 VTX057	INPUT1	064000000000
VTX	I		2281	2320 VTX058	INPUT1	065000000000
VTX	I		2321	2360 VTX059	INPUT1	066000000000
VTX	I		2361	2400 VTX060	INPUT1	067000000000
VTX	I		2401	2440 VTX061	INPUT1	068000000000
VTX	I		2441	2480 VTX062	INPUT1	069000000000
VTX	I		2481	2520 VTX063	INPUT1	070000000000
VTX	I		2521	2560 VTX064	INPUT1	071000000000
VTX	I		2561	2600 VTX065	INPUT1	072000000000
VTX	I		2601	2640 VTX066	INPUT1	073000000000
VTX	I		2641	2680 VTX067	INPUT1	074000000000
VTX	I		2681	2720 VTX068	INPUT1	075000000000
VTX	I		2721	2760 VTX069	INPUT1	076000000000
VTX	I		2761	2800 VTX070	INPUT1	077000000000
VTX	I		2801	2840 VTX071	INPUT1	078000000000
VTX	I		2841	2880 VTX072	INPUT1	079000000000
R03950		B0010	- STD/M	- Action Code	DATE	2/02/17
VTX	I		2881	2920 VTX073	INPUT1	080000000000
VTX	I		2921	2960 VTX074	INPUT1	081000000000
VTX	I		2961	3000 VTX075	INPUT1	082000000000
VTX	I		3001	3040 VTX076	INPUT1	083000000000
VTX	I		3041	3080 VTX077	INPUT1	084000000000
VTX	I		3081	3120 VTX078	INPUT1	085000000000
VTX	I		3121	3160 VTX079	INPUT1	086000000000
VTX	I		3161	3200 VTX080	INPUT1	087000000000
VTX	I		3201	3240 VTX081	INPUT1	088000000000
VTX	I		3241	3280 VTX082	INPUT1	089000000000

Figure F-4 Program Code for Program Type B0010 (3 of 11)

VTX I	32013320	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 - 2/02/17	
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,IOODSINK			INPUT1	153000000000
I/COPY JDECPY,IOOPS00			INPUT1	153100000000
INFDGI/COPY JDECPY,IOODSPROG			INPUT1	154000000000
DATESI*			INPUT1	155000000000
COPY I*			INPUT1	156000000000
C*	*****		MAINLINE	001000000000
C*	MAINLINE PROGRAM		MAINLINE	002000000000
C*	-----		MAINLINE	003000000000
C*	Process housekeeping.		MAINLINE	004000000000
C*			MAINLINE	005000000000
C	EXSR S999		MAINLINE	006000000000
C*	-----		MAINLINE	007000000000
C*			MAINLINE	008000000000
C*	If LR on, end program.		MAINLINE	009000000000
C*			MAINLINE	010000000000
C	*INLR CASEQ'1' EOJ		MAINLINE	011000000000
C*	-----		MAINLINE	012000000000
C*			MAINLINE	013000000000
C*			MAINLINE	014000000000
C*	If automatic inquiry set, process inquiry.		MAINLINE	015000000000
C*			MAINLINE	016000000000
C	SAUTO CASEQ'1' S003 24		MAINLINE	017000000000
C*	-----		MAINLINE	018000000000
C	END		MAINLINE	019000000000
C*			MAINLINE	020000000000
C*	Begin normal program processing.		MAINLINE	021000000000
C*	-----		MAINLINE	022000000000
C*			MAINLINE	023000000000
C	*INLR DOWEQ'0'		MAINLINE	024000000000
/*			MAINLINE	025000000000
/*	If #SPRNO field, do subfile record number validation		MAINLINE	026000000000

Figure F-5 Program Code for Program Type B0010 (4 of 11)

```

/*                                     MAINLINE 027000000000
*FLDNC*      #SFEND      ESFLCEND      MAINLINE 028000000000
/*                                     MAINLINE 029000000000
/* If SFCLR is used, process *in38 accordingly      MAINLINE 030000000000
/*                                     MAINLINE 031000000000
*FLDNC*      ?SFCLR      ESFLCLR      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      ENWRITE      MAINLINE 039000000000
/*                                     MAINLINE 040000000000
R03950      B0010      - STD/M      - Action Code      DATE - 2/02/17
/* If a subfile display, write format1 and formatc      MAINLINE 041000000000
/*                                     MAINLINE 042000000000
*FLDNC*      ?SFL      ENWRITE$FL      MAINLINE 043000000000
C      MOVE '1'      @@AID      MAINLINE 044000000000
C      EXCR S001      MAINLINE 045000000000
C      -----      MAINLINE 046000000000
C* Load data field dictionary parameters (one cycle only).      MAINLINE 047000000000
C*                                     MAINLINE 048000000000
C      $998      CAREQ' '      $998      MAINLINE 049000000000
C      -----      MAINLINE 050000000000
C      END      MAINLINE 051000000000
C*                                     MAINLINE 052000000000
C* Begin video screen read processing.      MAINLINE 053000000000
C*                                     MAINLINE 054000000000
C      SETOF      999301      MAINLINE 055000000000
C      READ &OLFILE      9998      MAINLINE 056000000000
C      Z-ADD0      ##RROW      MAINLINE 057000000000
C      Z-ADD0      ##RCOL      MAINLINE 058000000000
C* If video read timed out, end program.      MAINLINE 059000000000
C*                                     MAINLINE 060000000000
C      *IN99      CAREQ'1'      EOJ      LR      MAINLINE 061000000000
C      -----      MAINLINE 062000000000
C      @@AID      CAREQ#FEQJ      EOJ      LR      MAINLINE 063000000000
C      -----      MAINLINE 064000000000
C* If valid function key pressed, process and return.      MAINLINE 065000000000
C*                                     MAINLINE 066000000000
C      *IN15      IFEQ '1'      MAINLINE 067000000000
C      EXCR S00EX      MAINLINE 068000000000
C      -----      MAINLINE 069000000000
C      *INLR      CAREQ'1'      EOJ      MAINLINE 070000000000
C      -----      MAINLINE 071000000000
C      *IN15      CAREQ'1'      END      MAINLINE 072000000000
C      -----      MAINLINE 073000000000
C*                                     MAINLINE 074000000000
/*                                     MAINLINE 075000000000
/* If any selection exits, exar S00OP      MAINLINE 076000000000
/*                                     MAINLINE 077000000000
*DTAIC*      SELC      ES00OP      MAINLINE 078000000000
/*                                     MAINLINE 079000000000
/* If action code then exar C0001      MAINLINE 080000000000
/*                                     MAINLINE 081000000000
*FLDNC*      ACTION      ACTION      MAINLINE 082000000000
C* Load subfile records.      MAINLINE 083000000000
C*                                     MAINLINE 084000000000
C      EXCR S003      MAINLINE 085000000000
C      -----      MAINLINE 086000000000
/*                                     MAINLINE 087000000000
/* If any update files then exar S005      MAINLINE 088000000000
/*                                     MAINLINE 089000000000
*FILEC*      *ANY      DB      ES005      @      MAINLINE 090000000000
/*                                     MAINLINE 091000000000
/* If any update files and action code then do S010      MAINLINE 092000000000
/*                                     MAINLINE 093000000000
*FILEC*      *ANY      DB      *AND      @      MAINLINE 094000000000
/*                                     MAINLINE 095000000000
R03950      B0010      - STD/M      - Action Code      DATE - 2/02/17
*FLDNC*      ACTION      ES010A      MAINLINE 096000000000
/*                                     MAINLINE 097000000000
/* If a Master File 2 exists, then do S011.      MAINLINE 098000000000
/*                                     MAINLINE 099000000000
*FILEC*      *ANY      DB      *AND      @      MAINLINE 100000000000
/*                                     MAINLINE 101000000000
*FILEC*      *ANY      DB      *AND      2      MAINLINE 102000000000
/*                                     MAINLINE 103000000000
*FLDNC*      ACTION      ES011      MAINLINE 104000000000
C* Return for next input.      MAINLINE 105000000000
C*                                     MAINLINE 106000000000
C      END      TAG      MAINLINE 107000000000
C*                                     MAINLINE 108000000000
C*                                     MAINLINE 109000000000
C* Set correct message in line 24.      MAINLINE 110000000000
C*                                     MAINLINE 111000000000
C      *IN93      IFEQ '1'      MAINLINE 112000000000
C      MOVESVL24E      VDL24      MAINLINE 113000000000
C      ELSE      MAINLINE 114000000000
C      MOVESVL24M      VDL24      MAINLINE 115000000000
C      END      MAINLINE 116000000000
C      -----      MAINLINE 117000000000

```

Figure F-6 Program Code for Program Type B0010 (5 of 11)

```

C*                                     END                                MAINLINE 118000000000
C*                                     END                                MAINLINE 119000000000
C*                                     END                                MAINLINE 120000000000
C*                                     END                                MAINLINE 121000000000
C*                                     END                                MAINLINE 122000000000
C*                                     END                                MAINLINE 123000000000
C*                                     END                                MAINLINE 124000000000
C*                                     END                                MAINLINE 125000000000
C*                                     END                                MAINLINE 126000000000
COPY *****
C* SUBROUTINE S00EX - Process Function Keys                          S00EX-1 001000000000
C* -----                                                        S00EX-1 002000000000
C* Processing: 1. Determine function key pressed.                  S00EX-1 003000000000
C*              2. Process function key request.                   S00EX-1 004000000000
C* -----                                                        S00EX-1 005000000000
CSR      S00EX      REGGR                                           S00EX-1 006000000000
C* -----                                                        S00EX-1 007000000000
*FLDGR*  S00EX      REGGR                                           S00EX-1 008000000000
CSR      S00EX      REGGR                                           S00EX-1 009000000000
CSR      S00EX      REGGR                                           S00EX-1 010000000000
C* -----                                                        S00EX-1 011000000000
C* If NOJ requested, exit subroutine.                               S00EX-1 012000000000
C* -----                                                        S00EX-1 013000000000
CSR      S00EX      REGGR                                           S00EX-1 014000000000
C* -----                                                        S00EX-1 015000000000
C* If Display Keys pressed, exit to help facility and return.      S00EX-1 016000000000
C* -----                                                        S00EX-1 017000000000
CSR      S00EX      REGGR                                           S00EX-1 018000000000
C* -----                                                        S00EX-1 019000000000
C* -----                                                        S00EX-1 020000000000
CSR      S00EX      REGGR                                           S00EX-1 021000000000
CSR      S00EX      REGGR                                           S00EX-1 022000000000
C* -----                                                        S00EX-1 023000000000
CSR      S00EX      REGGR                                           S00EX-1 024000000000
CSR      S00EX      REGGR                                           S00EX-1 025000000000
C* -----                                                        S00EX-1 026000000000
CSR      S00EX      REGGR                                           S00EX-1 027000000000
CSR      S00EX      REGGR                                           S00EX-1 028000000000
C* -----                                                        S00EX-1 029000000000
R93950  S00EX      REGGR                                           S00EX-1 030000000000
C* -----                                                        S00EX-1 031000000000
CSR      S00EX      REGGR                                           S00EX-1 032000000000
C* -----                                                        S00EX-1 033000000000
C* If Cursor Sensitive Help Pressed, exit to CS Help.              S00EX-1 034000000000
C* -----                                                        S00EX-1 035000000000
CSR      S00EX      REGGR                                           S00EX-1 036000000000
C* -----                                                        S00EX-1 037000000000
CSR      S00EX      REGGR                                           S00EX-1 038000000000
C* -----                                                        S00EX-1 039000000000
CSR      S00EX      REGGR                                           S00EX-1 040000000000
CSR      S00EX      REGGR                                           S00EX-1 041000000000
C* -----                                                        S00EX-1 042000000000
CSR      S00EX      REGGR                                           S00EX-1 043000000000
CSR      S00EX      REGGR                                           S00EX-1 044000000000
C* -----                                                        S00EX-1 045000000000
CSR      S00EX      REGGR                                           S00EX-1 046000000000
C* -----                                                        S00EX-1 047000000000
CSR      S00EX      REGGR                                           S00EX-1 048000000000
C* -----                                                        S00EX-1 049000000000
CSR      S00EX      REGGR                                           S00EX-1 050000000000
C* -----                                                        S00EX-1 051000000000
CSR      S00EX      REGGR                                           S00EX-1 052000000000
C* -----                                                        S00EX-1 053000000000
CSR      S00EX      REGGR                                           S00EX-1 054000000000
C* -----                                                        S00EX-1 055000000000
C* If Display errors pressed, exit to error messages.              S00EX-1 056000000000
C* -----                                                        S00EX-1 057000000000
CSR      S00EX      REGGR                                           S00EX-1 058000000000
C* -----                                                        S00EX-1 059000000000
CSR      S00EX      REGGR                                           S00EX-1 060000000000
C* -----                                                        S00EX-1 061000000000
CSR      S00EX      REGGR                                           S00EX-1 062000000000
C* -----                                                        S00EX-1 063000000000
CSR      S00EX      REGGR                                           S00EX-1 064000000000
C* -----                                                        S00EX-1 065000000000
CSR      S00EX      REGGR                                           S00EX-1 066000000000
C* -----                                                        S00EX-1 067000000000
CSR      S00EX      REGGR                                           S00EX-1 068000000000
C* -----                                                        S00EX-1 069000000000
CSR      S00EX      REGGR                                           S00EX-1 070000000000
C* -----                                                        S00EX-1 071000000000
CSR      S00EX      REGGR                                           S00EX-1 072000000000
C* -----                                                        S00EX-1 073000000000
CSR      S00EX      REGGR                                           S00EX-1 074000000000
C* -----                                                        S00EX-1 075000000000
C* If HELP key pressed, exit to help facility and return.          S00EX-1 076000000000
C* -----                                                        S00EX-1 077000000000
CSR      S00EX      REGGR                                           S00EX-1 078000000000
C* -----                                                        S00EX-1 079000000000
CSR      S00EX      REGGR                                           S00EX-1 080000000000
C* -----                                                        S00EX-1 081000000000
CSR      S00EX      REGGR                                           S00EX-1 082000000000
C* -----                                                        S00EX-1 083000000000

```

Figure F-7 Program Code for Program Type B0010 (6 of 11)

```

CSR          PARM          I00SC          S00EX-1  084000000000
CSR          PARM          SRVFS          S00EX-1  085000000000
CSR          PARM          I00CSR          S00EX-1  086000000000
CSR          GOTO ENDEXE          S00EX-1  087000000000
R03950          B0010          - STD/M          - Action Code          DATE - 2/02/17
C*          -----
CSR          END          S00EX-1  088000000000
C*          S00EX-1  089000000000
C*          If Clear screen pressed, clear screen and return.          S00EX-1  090000000000
C*          -----
C*          S00EX-1  091000000000
C*          S00EX-1  092000000000
C*          S00EX-1  093000000000
CSR          @@AID          IFBQ #FCLR          S00EX-1  094000000000
CSR          EXSR S001          S00EX-1  095000000000
C*          -----
CSR          GOTO ENDEXE          S00EX-1  096000000000
C*          -----
CSR          S00EX-1  097000000000
EXITOCSS          END          S00EX-1  098000000000
C*          S00EX-1  099000000000
C*          Process roll up and down keys.          S00EX-1  100000000000
C*          -----
C*          S00EX-1  101000000000
C*          S00EX-1  102000000000
C*          S00EX-1  103000000000
CSR          @@AID          IFBQ #FROLU          S00EX-1  104000000000
CSR          @@AID          ORBQ #FROLD          S00EX-1  105000000000
CSR          $SECUR          DOUEQ' '          S00EX-1  106000000000
CSR          MOVE ' '          $SECUR 1          S00EX-1  107000000000
C*          S00EX-1  108000000000
C*          If ROLL UP key pressed, process read next.          S00EX-1  109000000000
C*          -----
C*          S00EX-1  110000000000
CSR          @@AID          IFBQ #FROLU          S00EX-1  111000000000
C*          Reset error indicators if roll          S00EX-1  112000000000
C*          S00EX-1  113000000000
CSR          MOVEASRESET          *IN,41          S00EX-1  114000000000
CSR          MOVE '0'          *IN,40          S00EX-1  115000000000
CSR          SETOF          S18299          S00EX-1  116000000000
MF          %          READP401FORMAT          9981          S00EX-1  117000000000
CSR          *INB1          IFBQ '1'          S00EX-1  118000000000
MF          $RKEY          SETLL401FORMAT          S00EX-1  119000000000
CSR          SETOF          S299          S00EX-1  120000000000
MF          %          READP401FORMAT          9982          S00EX-1  121000000000
C*          S00EX-1  122000000000
C*          If error on read, set error.          S00EX-1  123000000000
C*          S00EX-1  124000000000
CSR          *INB2          IFBQ '1'          S00EX-1  125000000000
CSR          SETON          S00EX-1  126000000000
CSR          MOVE '1'          @NK,2          9341          S00EX-1  127000000000
CSR          GOTO ENDEXE          S00EX-1  128000000000
C*          -----
CSR          S00EX-1  129000000000
CSR          END          S00EX-1  130000000000
CSR          END          S00EX-1  131000000000
CSR          END          S00EX-1  132000000000
C*          S00EX-1  133000000000
C*          If ROLL DOWN key pressed, process read prior.          S00EX-1  134000000000
C*          -----
C*          S00EX-1  135000000000
CSR          @@AID          IFBQ #FROLD          S00EX-1  136000000000
C*          Reset error indicators if roll          S00EX-1  137000000000
C*          S00EX-1  138000000000
CSR          MOVEASRESET          *IN,41          S00EX-1  139000000000
CSR          MOVE '0'          *IN,40          S00EX-1  140000000000
CSR          SETOF          S18299          S00EX-1  141000000000
MF          %          READP401FORMAT          9981          S00EX-1  142000000000
R03950          B0010          - STD/M          - Action Code          DATE - 2/02/17
C*          S00EX-1  143000000000
MF          *INB1          IFBQ '1'          S00EX-1  144000000000
MF          $RKEY          SETLL401FORMAT          S00EX-1  145000000000
CSR          SETOF          S299          S00EX-1  146000000000
MF          %          READP401FORMAT          9982          S00EX-1  147000000000
C*          S00EX-1  148000000000
C*          If error on read, set error.          S00EX-1  149000000000
C*          S00EX-1  150000000000
CSR          *INB2          IFBQ '1'          S00EX-1  151000000000
CSR          SETON          S00EX-1  152000000000
CSR          MOVE '1'          @NK,2          9341          S00EX-1  153000000000
CSR          GOTO ENDEXE          S00EX-1  154000000000
C*          -----
CSR          S00EX-1  155000000000
CSR          END          S00EX-1  156000000000
CSR          END          S00EX-1  157000000000
CSR          END          S00EX-1  158000000000
C*          S00EX-1  159000000000
C*          Load video screen data on roll keys.          S00EX-1  160000000000
C*          -----
C*          S00EX-1  161000000000
CSR          @@AID          IFBQ #FROLU          S00EX-1  162000000000
CSR          @@AID          ORBQ #FROLD          S00EX-1  163000000000
/*          S00EX-1  164000000000
/*          Include record lock logic if update files exist.          S00EX-1  165000000000
/*          S00EX-1  166000000000
*FILE*          *ANY          DB          UNLOCK          @          S00EX-1  167000000000
C*          S00EX-1  168000000000
MCU01C          S00EX-1  169000000000
MCU01C          Cost Center security edit.          S00EX-1  170000000000
MCU01C          S00EX-1  171000000000
MCU01CSR          MOVELL401(FILE )#FILE          S00EX-1  172000000000

```

Figure F-8 Program Code for Program Type B0010 (7 of 11)

MCU01CSR		MCVTL601KEY	#MCU	S00EX-1	173000000000
MCU01CSR	#FAUT	IFNE '1'		S00EX-1	174000000000
MCU01CSR	#FAUT	ANDNE '1'		S00EX-1	175000000000
MCU01CSR		EXSR C0000		S00EX-1	176000000000
MCU01C*		-----		S00EX-1	177000000000
MCU01CSR		END		S00EX-1	178000000000
MCU01CSR	#AUT	IFNE '1'		S00EX-1	179000000000
MCU01CSR	#FAUT	ANDNE '1'		S00EX-1	180000000000
MCU01CSR	#FAUT	ANDNE '1'		S00EX-1	181000000000
MCU01CSR		MOVE '1'	\$SECU	S00EX-1	182000000000
MCU01CSR		END		S00EX-1	183000000000
CSR	\$SECU	CASEQ' '	S004	S00EX-1	184000000000
C*		-----		S00EX-1	185000000000
CSR		END		S00EX-1	186000000000
C*				S00EX-1	187000000000
CSR		END		S00EX-1	188000000000
C*				S00EX-1	189000000000
CSR		END		S00EX-1	190000000000
CSR		GOTO ENDEXE		S00EX-1	191000000000
C*		-----		S00EX-1	192000000000
CSR		END		S00EX-1	193000000000
C*				S00EX-1	194000000000
CSR	\$BAID	IFNE '1'		S00EX-1	195000000000
CSR		SETON	0193	S00EX-1	196000000000
CSR		GOTO ENDEXE		S00EX-1	197000000000
C*		-----		S00EX-1	198000000000
CSR		END		S00EX-1	199000000000
C*				S00EX-1	200000000000
CSR	ENDEXE	ENDSR		S00EX-1	201000000000
R93950		B0010	- STD/M - Action Code		DATE - 2/02/17
COPY C*		-----		S00EX-1	202000000000
/*				S00OP	000100000000
/*		If the display file has the selection option field,		S00OP	000200000000
/*		include the S00OP subroutine to process selection options.		S00OP	000300000000
/*				S00OP	000400000000
#FLDNC**	VDSLC	*AND		S00OP	001000000000
-FLDNC**	SFSLC	S00OP-1		S00OP	001100000000
/*				S00OP	001200000000
#FLDNC**	SFSLC	S00OP-2		S00OP	001300000000
C*				S00VL-1	001000000000
C*		SUBROUTINE S00VL - Cursor Control Return Values		S00VL-1	002000000000
C*		-----		S00VL-1	003000000000
C*				S00VL-1	004000000000
C*		By format, find the field to update and move in the		S00VL-1	005000000000
C*		returned value. If the format is a subfile, the record		S00VL-1	006000000000
C*		to change is found in \$SRRN.		S00VL-1	007000000000
C*				S00VL-1	008000000000
CSR	S00VL	REGSR		S00VL-1	009000000000
C*		-----		S00VL-1	010000000000
C*				S00VL-1	011000000000
CSR	\$SRVAL	IFEQ '*BLANK'		S00VL-1	012000000000
CSR		MOVE *BLANK	\$SRVAL	S00VL-1	013000000000
CSR		END		S00VL-1	014000000000
S00VL*				S00VL-1	015000000000
C*				S00VL-1	016000000000
CSR	END0VL	ENDSR		S00VL-1	017000000000
COPY C*		-----		S00VL-1	018000000000
C*		SUBROUTINE S001 - Clear Fields		S001-1	001000000000
C*		-----		S001-1	002000000000
C*		Processing: 1. Reset all video screen and data file fields		S001-1	003000000000
C*		for next transaction.		S001-1	004000000000
C*		2. Clear action code only if requested.		S001-1	005000000000
C*				S001-1	006000000000
CSR	S001	REGSR		S001-1	007000000000
C*		-----		S001-1	008000000000
C*				S001-1	009000000000
C*				S001-1	010000000000
C*		Reset fields for next transaction.		S001-1	011000000000
C*				S001-1	012000000000
C*				S001-1	013000000000
MF CSR	*NOKEY	CLEAR&01FORMAT		S001-1	013100000000
CLRY C*				S001-1	014000000000
CSR		MOVLSVL24M	VDL24	S001-1	015000000000
CSR		MOVE ' '	\$IN3? 1	S001-1	016000000000
C*				S001-1	017000000000
C*		Clear action code only if clear screen action.		S001-1	018000000000
C*				S001-1	019000000000
CSR	\$BAID	IFEQ #FCLR		S001-1	020000000000
CSR		MOVE *ALL'0'	\$RESET	S001-1	021000000000
CSR		MOVESRESET	*IN,41	S001-1	022000000000
CSR		MOVE ' '	ACTION 1	S001-1	023000000000
CLRN C*				S001-1	024000000000
CSR		END		S001-1	025000000000
C*				S001-1	026000000000
CSR	END001	ENDSR		S001-1	027000000000
COPY C*		-----		S001-1	028000000000
C*				S003-1	001000000000
C*		SUBROUTINE S003 - Edit Key		S003-1	002000000000
C*		-----		S003-1	003000000000
R93950		B0010	- STD/M - Action Code		DATE - 2/02/17
C*				S003-1	004000000000
C*		Processing: 1. Clear error indicators and arrays.		S003-1	005000000000
C*		2. Load input keys.		S003-1	006000000000
C*		3. Validate master file key.		S003-1	007000000000
C*		4. Release master file record lock.		S003-1	008000000000

Figure F-9 Program Code for Program Type B0010 (8 of 11)

```

C*          5. Load video screen output on inquiry.
C*
CSR      S003      BEGR
C*          -----
C*          Load data field dictionary parameters (one cycle only).
C*
CSR      S998      CASEQ/ ' '      S998
C*          -----
CSR      END
C*
C*          Reset error indicators and arrays.
C*
CSR      MOVE 'ALL/0' $RESET 39
CSR      MOVE 'BLANK $RESET1 63
CSR      MOVE$RESET 'IN,41
CSR      MOVE$RESET1 @M,2
CSR      CLEAR$ERR
C*-----
KEYS
C*-----
MF      CSR      CHAIN$01FORMAT      9899
MOU$IC*
MOU$IC*      Cost Center security edit.
MOU$IC*
MOU$ICSR      MOVE$401(FILE )#FILE
MOU$ICSR      MOVE$401KEY #MCU
MOU$ICSR      #AUT      IFNE '1'
MOU$ICSR      #FAUT      ANDNE '1'
MOU$ICSR      EXSR C0000
MOU$IC*      -----
MOU$ICSR      END
MOU$ICSR      #AUT      IFNE '1'
MOU$ICSR      #FAUT      ANDNE '1'
MOU$ICSR      #MAUT      ANDNE '1'
MOU$ICSR      MOVE '1' $SECR 1
MOU$ICSR      END
C*
C*          If security violation, set error condition.
C*
CSR      $SECR      IFEQ '1'
CSR      MOVE '1' @M,8
CSR      SETCN
CSR      MOVE ' ' $SECR 1
CSR      GOTO END003
C*          -----
C*          Edit result of read and action code.
C*
CSR      *IN98      IFEQ '1'
CSR      *IN21      COMP '0'      41 *error*
CSR      ELSE
CSR      *IN21      COMP '1'      41 *error*
R93950      CSR      END      D0010 - STD/M - Action Code
C*
C*          If indicator 41 on, invalid key for action code.
C*
CSR      *IN41      IFEQ '1'
CSR      MOVE '1' @M,2
CSR      SETCN
CSR      END
C*
C*          If indicator 99 on, record in use.
C*
CSR      *IN99      IFEQ '1'
CSR      CALL 'P9$RLCK'      81
C*          -----
CSR      FARM      ##PSDG
CSR      MOVE '1' @M,6
CSR      SETCN
CSR      END
C*-----
C*          If not inquiry, skip remainder of subroutine.
C*
CSR      *IN24      CASEQ/0'      END003
C*          -----
C*          Release record lock on master file.
C*
CSR      *IN98      IFEQ '0'
CSR      *IN99      ANDQ/0'
CSR      EXCPTUNLOCK
CSR      END
C*
C*          If errors, skip remainder of subroutine.
C*
CSR      *IN93      CASEQ/1'      END003
C*          -----
C*          Move data base information to video screen.

```

Figure F-10 Program Code for Program Type B0010 (9 of 11)

```

C*                                     S003-1  090000000000
CSR                                     S003-1  091000000000
C*                                     S003-1  092000000000
C*                                     S003-1  093000000000
CSR                                     S003-1  094000000000
COPY C*                               S003-1  095000000000
C*                                     S004-1  001000000000
C* SUBROUTINE S004 - Load Video Screen Data S004-1  002000000000
C*                                     S004-1  003000000000
C*                                     S004-1  004000000000
C* Processing: 1. Move data base information to video screen. S004-1  005000000000
C* All video screen fields are alpha and S004-1  006000000000
C* therefore numeric information must be S004-1  007000000000
C* processed through subroutine C0014 to set S004-1  008000000000
C* proper decimals and provide editing for S004-1  009000000000
C* display on screen. S004-1  010000000000
C*                                     S004-1  011000000000
C* Date fields must be converted from their S004-1  012000000000
C* internal format of month, day and year or S004-1  013000000000
R93950 C* B0010 - STD/M - Action Code S004-1  014000000000
C* julian to the system format using program S004-1  015000000000
C* X0028. S004-1  016000000000
CSR S004 REGR S004-1  017000000000
C* S004-1  018000000000
DSP1 C* S004-1  025000000000
CSR END004 ENDR S004-1  026000000000
COPY C* S004-1  027000000000
C* SUBROUTINE S005 - Scrub Input S005-1  001000000000
C* S005-1  002000000000
C* S005-1  003000000000
C* Processing: 1. Validate all video input. S005-1  004000000000
C* All numeric fields must be processed S005-1  005000000000
C* thru subroutines C0012 and C0015 in order S005-1  006000000000
C* to scrub the alpha input field and convert S005-1  007000000000
C* back to internal numeric representation of S005-1  008000000000
C* 15 digits and 0 decimals. S005-1  009000000000
C* S005-1  010000000000
C* Date fields must be converted from system S005-1  011000000000
C* format to their internal format of month, S005-1  012000000000
C* day and year or julian using program X0028. S005-1  013000000000
C* 2. Update data record fields from video. S005-1  014000000000
C* S005-1  015000000000
C* S005-1  016000000000
CSR S005 REGR S005-1  017000000000
C* S005-1  018000000000
C* S005-1  019000000000
C* If not addition or change, bypass subroutine S005-1  020000000000
C* S005-1  021000000000
CSR *IN21 IFBQ '0' S005-1  022000000000
CSR *IN22 ANBQ '0' S005-1  023000000000
CSR GOTO END005 S005-1  024000000000
C* S005-1  025000000000
CSR END S005-1  026000000000
C* S005-1  028000000000
FIELD* S005-1  029000000000
CSR END005 ENDR S005-1  030000000000
COPY C* S005-1  031000000000
C* SUBROUTINE S010 - Update Data Base S010-1  001000000000
C* S010-1  002000000000
C* S010-1  003000000000
C* Processing: 1. Update data base file based upon valid S010-1  004000000000
C* action codes. S010-1  005000000000
C* S010-1  006000000000
CSR S010 REGR S010-1  007000000000
C* S010-1  008000000000
C* S010-1  009000000000
AC* S010-1  010000000000
AC* If add action, add record. S010-1  011000000000
AC* S010-1  012000000000
MF AC* *IN21 IFBQ '1' S010-1  013000000000
AC* & WRITE&01FORMAT 99 S010-1  014000000000
AC* END S010-1  015000000000
C* S010-1  016000000000
CC* If change action, update record. S010-1  017000000000
CC* S010-1  018000000000
CC* S010-1  019000000000
MF CC* *IN22 IFBQ '1' S010-1  020000000000
CC* & UPDAT&01FORMAT 99 S010-1  021000000000
CC* END S010-1  022000000000
R93950 C* B0010 - STD/M - Action Code S010-1  023000000000
C* S010-1  024000000000
DC* If delete action, delete record. S010-1  025000000000
DC* S010-1  026000000000
MF DC* *IN23 IFBQ '1' S010-1  027000000000
DC* & DELET&01FORMAT 99 S010-1  028000000000
C* S010-1  029000000000
C* Clear data field for next transaction S010-1  030000000000
C* S010-1  031000000000
CSR MOVE #PCLR @&AID S010-1  032000000000
CSR EXSR S001 S010-1  033000000000
C* S010-1  034000000000
CSR END010 ENDR S010-1  035000000000
COPY C* S010-1  036000000000
C* S010-1  037000000000
C* S010-1  038000000000
C* S010-1  039000000000
C* S010-1  040000000000
C* S010-1  041000000000
C* S010-1  042000000000
C* S010-1  043000000000
C* S010-1  044000000000
C* S010-1  045000000000
C* S010-1  046000000000
C* S010-1  047000000000
C* S010-1  048000000000
C* S010-1  049000000000
C* S010-1  050000000000
C* S010-1  051000000000
C* S010-1  052000000000
C* S010-1  053000000000
C* S010-1  054000000000
C* S010-1  055000000000
C* S010-1  056000000000
C* S010-1  057000000000
C* S010-1  058000000000
C* S010-1  059000000000
C* S010-1  060000000000
C* S010-1  061000000000
C* S010-1  062000000000
C* S010-1  063000000000
C* S010-1  064000000000
C* S010-1  065000000000
C* S010-1  066000000000
C* S010-1  067000000000
C* S010-1  068000000000
C* S010-1  069000000000
C* S010-1  070000000000
C* S010-1  071000000000
C* S010-1  072000000000
C* S010-1  073000000000
C* S010-1  074000000000
C* S010-1  075000000000
C* S010-1  076000000000
C* S010-1  077000000000
C* S010-1  078000000000
C* S010-1  079000000000
C* S010-1  080000000000
C* S010-1  081000000000
C* S010-1  082000000000
C* S010-1  083000000000
C* S010-1  084000000000
C* S010-1  085000000000
C* S010-1  086000000000
C* S010-1  087000000000
C* S010-1  088000000000
C* S010-1  089000000000
C* S010-1  090000000000
C* S010-1  091000000000
C* S010-1  092000000000
C* S010-1  093000000000
C* S010-1  094000000000
C* S010-1  095000000000
C* S010-1  096000000000
C* S010-1  097000000000
C* S010-1  098000000000
C* S010-1  099000000000
C* S010-1  100000000000
C* S010-1  101000000000
C* S010-1  102000000000
C* S010-1  103000000000
C* S010-1  104000000000
C* S010-1  105000000000
C* S010-1  106000000000
C* S010-1  107000000000
C* S010-1  108000000000
C* S010-1  109000000000
C* S010-1  110000000000
C* S010-1  111000000000
C* S010-1  112000000000
C* S010-1  113000000000
C* S010-1  114000000000
C* S010-1  115000000000
C* S010-1  116000000000
C* S010-1  117000000000
C* S010-1  118000000000
C* S010-1  119000000000
C* S010-1  120000000000
C* S010-1  121000000000
C* S010-1  122000000000
C* S010-1  123000000000
C* S010-1  124000000000
C* S010-1  125000000000
C* S010-1  126000000000
C* S010-1  127000000000
C* S010-1  128000000000
C* S010-1  129000000000
C* S010-1  130000000000
C* S010-1  131000000000
C* S010-1  132000000000
C* S010-1  133000000000
C* S010-1  134000000000
C* S010-1  135000000000
C* S010-1  136000000000
C* S010-1  137000000000
C* S010-1  138000000000
C* S010-1  139000000000
C* S010-1  140000000000
C* S010-1  141000000000
C* S010-1  142000000000
C* S010-1  143000000000
C* S010-1  144000000000
C* S010-1  145000000000
C* S010-1  146000000000
C* S010-1  147000000000
C* S010-1  148000000000
C* S010-1  149000000000
C* S010-1  150000000000
C* S010-1  151000000000
C* S010-1  152000000000
C* S010-1  153000000000
C* S010-1  154000000000
C* S010-1  155000000000
C* S010-1  156000000000
C* S010-1  157000000000
C* S010-1  158000000000
C* S010-1  159000000000
C* S010-1  160000000000
C* S010-1  161000000000
C* S010-1  162000000000
C* S010-1  163000000000
C* S010-1  164000000000
C* S010-1  165000000000
C* S010-1  166000000000
C* S010-1  167000000000
C* S010-1  168000000000
C* S010-1  169000000000
C* S010-1  170000000000
C* S010-1  171000000000
C* S010-1  172000000000
C* S010-1  173000000000
C* S010-1  174000000000
C* S010-1  175000000000
C* S010-1  176000000000
C* S010-1  177000000000
C* S010-1  178000000000
C* S010-1  179000000000
C* S010-1  180000000000
C* S010-1  181000000000
C* S010-1  182000000000
C* S010-1  183000000000
C* S010-1  184000000000
C* S010-1  185000000000
C* S010-1  186000000000
C* S010-1  187000000000
C* S010-1  188000000000
C* S010-1  189000000000
C* S010-1  190000000000
C* S010-1  191000000000
C* S010-1  192000000000
C* S010-1  193000000000
C* S010-1  194000000000
C* S010-1  195000000000
C* S010-1  196000000000
C* S010-1  197000000000
C* S010-1  198000000000
C* S010-1  199000000000
C* S010-1  200000000000
C* S010-1  201000000000
C* S010-1  202000000000
C* S010-1  203000000000
C* S010-1  204000000000
C* S010-1  205000000000
C* S010-1  206000000000
C* S010-1  207000000000
C* S010-1  208000000000
C* S010-1  209000000000
C* S010-1  210000000000
C* S010-1  211000000000
C* S010-1  212000000000
C* S010-1  213000000000
C* S010-1  214000000000
C* S010-1  215000000000
C* S010-1  216000000000
C* S010-1  217000000000
C* S010-1  218000000000
C* S010-1  219000000000
C* S010-1  220000000000
C* S010-1  221000000000
C* S010-1  222000000000
C* S010-1  223000000000
C* S010-1  224000000000
C* S010-1  225000000000
C* S010-1  226000000000
C* S010-1  227000000000
C* S010-1  228000000000
C* S010-1  229000000000
C* S010-1  230000000000
C* S010-1  231000000000
C* S010-1  232000000000
C* S010-1  233000000000
C* S010-1  234000000000
C* S010-1  235000000000
C* S010-1  236000000000
C* S010-1  237000000000
C* S010-1  238000000000
C* S010-1  239000000000
C* S010-1  240000000000
C* S010-1  241000000000
C* S010-1  242000000000
C* S010-1  243000000000
C* S010-1  244000000000
C* S010-1  245000000000
C* S010-1  246000000000
C* S010-1  247000000000
C* S010-1  248000000000
C* S010-1  249000000000
C* S010-1  250000000000
C* S010-1  251000000000
C* S010-1  252000000000
C* S010-1  253000000000
C* S010-1  254000000000
C* S010-1  255000000000
C* S010-1  256000000000
C* S010-1  257000000000
C* S010-1  258000000000
C* S010-1  259000000000
C* S010-1  260000000000
C* S010-1  261000000000
C* S010-1  262000000000
C* S010-1  263000000000
C* S010-1  264000000000
C* S010-1  265000000000
C* S010-1  266000000000
C* S010-1  267000000000
C* S010-1  268000000000
C* S010-1  269000000000
C* S010-1  270000000000
C* S010-1  271000000000
C* S010-1  272000000000
C* S010-1  273000000000
C* S010-1  274000000000
C* S010-1  275000000000
C* S010-1  276000000000
C* S010-1  277000000000
C* S010-1  278000000000
C* S010-1  279000000000
C* S010-1  280000000000
C* S010-1  281000000000
C* S010-1  282000000000
C* S010-1  283000000000
C* S010-1  284000000000
C* S010-1  285000000000
C* S010-1  286000000000
C* S010-1  287000000000
C* S010-1  288000000000
C* S010-1  289000000000
C* S010-1  290000000000
C* S010-1  291000000000
C* S010-1  292000000000
C* S010-1  293000000000
C* S010-1  294000000000
C* S010-1  295000000000
C* S010-1  296000000000
C* S010-1  297000000000
C* S010-1  298000000000
C* S010-1  299000000000
C* S010-1  300000000000
C* S010-1  301000000000
C* S010-1  302000000000
C* S010-1  303000000000
C* S010-1  304000000000
C* S010-1  305000000000
C* S010-1  306000000000
C* S010-1  307000000000
C* S010-1  308000000000
C* S010-1  309000000000
C* S010-1  310000000000
C* S010-1  311000000000
C* S010-1  312000000000
C* S010-1  313000000000
C* S010-1  314000000000
C* S010-1  315000000000
C* S010-1  316000000000
C* S010-1  317000000000
C* S010-1  318000000000
C* S010-1  319000000000
C* S010-1  320000000000
C* S010-1  321000000000
C* S010-1  322000000000
C* S010-1  323000000000
C* S010-1  324000000000
C* S010-1  325000000000
C* S010-1  326000000000
C* S010-1  327000000000
C* S010-1  328000000000
C* S010-1  329000000000
C* S010-1  330000000000
C* S010-1  331000000000
C* S010-1  332000000000
C* S010-1  333000000000
C* S010-1  334000000000
C* S010-1  335000000000
C* S010-1  336000000000
C* S010-1  337000000000
C* S010-1  338000000000
C* S010-1  339000000000
C* S010-1  340000000000
C* S010-1  341000000000
C* S010-1  342000000000
C* S010-1  343000000000
C* S010-1  344000000000
C* S010-1  345000000000
C* S010-1  346000000000
C* S010-1  347000000000
C* S010-1  348000000000
C* S010-1  349000000000
C* S010-1  350000000000
C* S010-1  351000000000
C* S010-1  352000000000
C* S010-1  353000000000
C* S010-1  354000000000
C* S010-1  355000000000
C* S010-1  356000000000
C* S010-1  357000000000
C* S010-1  358000000000
C* S010-1  359000000000
C* S010-1  360000000000
C* S010-1  361000000000
C* S010-1  362000000000
C* S010-1  363000000000
C* S010-1  364000000000
C* S010-1  365000000000
C* S010-1  366000000000
C* S010-1  367000000000
C* S010-1  368000000000
C* S010-1  369000000000
C* S010-1  370000000000
C* S010-1  371000000000
C* S010-1  372000000000
C* S010-1  373000000000
C* S010-1  374000000000
C* S010-1  375000000000
C* S010-1  376000000000
C* S010-1  377000000000
C* S010-1  378000000000
C* S010-1  379000000000
C* S010-1  380000000000
C* S010-1  381000000000
C* S010-1  382000000000
C* S010-1  383000000000
C* S010-1  384000000000
C* S010-1  385000000000
C* S010-1  386000000000
C* S010-1  387000000000
C* S010-1  388000000000
C* S010-1  389000000000
C* S010-1  390000000000
C* S010-1  391000000000
C* S010-1  392000000000
C* S010-1  393000000000
C* S010-1  394000000000
C* S010-1  395000000000
C* S010-1  396000000000
C* S010-1  397000000000
C* S010-1  398000000000
C* S010-1  399000000000
C* S010-1  400000000000
C* S010-1  401000000000
C* S010-1  402000000000
C* S010-1  403000000000
C* S010-1  404000000000
C* S010-1  405000000000
C* S010-1  406000000000
C* S010-1  407000000000
C* S010-1  408000000000
C* S010-1  409000000000
C* S010-1  410000000000
C* S010-1  411000000000
C* S010-1  412000000000
C* S010-1  413000000000
C* S010-1  414000000000
C* S010-1  415000000000
C* S010-1  416000000000
C* S010-1  417000000000
C* S010-1  418000000000
C* S010-1  419000000000
C* S010-1  420000000000
C* S010-1  421000000000
C* S010-1  422000000000
C* S010-1  423000000000
C* S010-1  424000000000
C* S010-1  425000000000
C* S010-1  426000000000
C* S010-1  427000000000
C* S010-1  428000000000
C* S010-1  429000000000
C* S010-1  430000000000
C* S010-1  431000000000
C* S010-1  432000000000
C* S010-1  433000000000
C* S010-1  434000000000
C* S010-1  435000000000
C* S010-1  436000000000
C* S010-1  437000000000
C* S010-1  438000000000
C* S010-1  439000000000
C* S010-1  440000000000
C* S010-1  441000000000
C* S010-1  442000000000
C* S010-1  443000000000
C* S010-1  444000000000
C* S010-1  445000000000
C* S010-1  446000000000
C* S010-1  447000000000
C* S010-1  448000000000
C* S010-1  449000000000
C* S010-1  450000000000
C* S010-1  451000000000
C* S010-1  452000000000
C* S010-1  453000000000
C* S010-1  454000000000
C* S010-1  455000000000
C* S010-1  456000000000
C* S010-1  457000000000
C* S010-1  458000000000
C* S010-1  4
```

Figure F-11 Program Code for Program Type B0010 (10 of 11)

```

C* SUBROUTINE S998 - Load dictionary parameters.
C* -----
C* S998 REGGR
C* -----
DPRMNC*
C*
C* Set subroutine execution flag.
C*
CSR MOVE '1' S998 1
C*
CSR END998 ENDGR
C*****
C* SUBROUTINE S999 - Housekeeping
C* -----
C*
C* Processing: 1. Load video screen text.
C* 2. Retrieve screen title data area, text
C* for unauthorized access, center video
C* title and move to video screen.
C* 3. Initialize key list.
C* 4. Load roll keys.
C* 5. Passed parameters.
C* 6. Load error message array.
C*
CSR S999 REGGR
C* -----
C*
C* Required program parameters.
C*
ENTRYCSR *ENTRY PLIST
AUTOIC*
C* -----
C*
C* Load video screen text.
C*
CSR MOVE%$FILE PSKEY 10
VTX1
C/COPY JDECPY,C00SC
C* -----
C*
C* If processing options exist, load processing options
C*
R93950 B0010 - STD/M - Action Code DATE - 2/02/17
+FLMNC* *OPTION SOPTIONK
MLISTC*
C*
C* Load roll key upper and lower key values.
C*
MF CSR *LIKE DEPN 401KEYFLD $RKEY
CSR *LIKE DEPN $RKEY $RKEY
CSR MOVE $LOVAL $RKEY
CSR MOVE *ALL'9' $RKEY
C* -----
C*
C* Load error messages array.
C*
CSR MOVE '0001' ENK,01 Inv Action
CSR MOVE '0002' ENK,02 Inv Key
CSR MOVE '0003' ENK,03 Inv Blanks
CSR MOVE '0004' ENK,04 Inv Date
CSR MOVE '0005' ENK,05 Inv Next Nbr
CSR MOVE '0007' ENK,06 In Use
CSR MOVE '0025' ENK,07 Inv Values
CSR MOVE '0026' ENK,08 Inv MCU
ENK CSR MOVE '0027' ENK,09 Inv Desc Ttl
C* -----
C*
C* Load invalid action code array.
C*
ACTN CSR MOVE% ' ' $RAC
C* -----
C*
C* Load system date.
C*
CSR TIME $WRK12 120
CSR MOVE $WRK12 $SDAT 60
CSR MOVE $SDAT $SIDAT 6
CSR MOVE%$SYVAL ' $TFMT 7
CSR MOVE%$BLANKS $EDAT 8
CSR MOVE%$JUL ' $TFMT 7
CSR MOVE%$NONE ' $SEP 7
CSR MOVE ' ' $ERTST 1
CSR CALL 'X0028 '
C* -----
CSR PARM $SIDAT
CSR PARM $EDAT
CSR PARM $TFMT
CSR PARM $SEP
CSR PARM $ERTST
CSR MOVE $SIDAT $SUFMT 60
C* -----
CSR END999 ENDGR

```

Figure F-12 Program Code for Program Type B0010 (11 of 11)

```

C*****
C*
C* If processing options exist, include copy module
C*
+FLMNC* *OPTION SOPTIONK
COPY C*****
MF G401PMT E UNLOCK

```

Functional Servers

This appendix contains the topic:

- [Section G.1, "Example: Voucher Processing Functional Server."](#)

Several JD Edwards World 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 using a functional server include:

- Reduces maintenance of entry programs because edit rules reside in one central location.
- Allows you to standardize documents across all applications because you create them using the same business rules.
- Separates the user interface (screen appearance and interaction) from the functions of a program.

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. JD Edwards World provides DREAM Writer version ZJDE0001 as the default functional server version for your entry programs.

Caution: 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 *Understand DREAM Writer in the JD Edwards World Technical Foundation Guide*.

G.1 Example: Voucher Processing Functional Server

The following programs use the voucher processing functional server. JD Edwards World provides two demo versions of the functional server, ZJDE0001 and ZJDE0002.

- Speed Voucher Entry (P040015)
- Standard Voucher Entry (P04105)
- Void Payment Entry (P4704103)
- Credit Tied to Debit Bill (P041010)
- Multi-Voucher (P041017)
- Calculate Withholding (P04580)

Index

A

A0010 - Interactive Subfile Inquiry program, C-2
A0020 - Interactive Single Record Inquiry program, C-3
Abbreviations for the program types index, 30-1
About
 abbreviations for the program types index, 30-1
 action diagramming, 27-1
 additional tools, 24-1
 changing generated source, 17-1
 clone status all/only active toggle, 30-4
 creating or modifying program types, 30-1
 edit screen, 9-4
 foundation information, 2-1
 glossary K, 30-4
 logic modules, 31-1
 master source code, 1-6
 option and function exits, 10-1
 options for program types, 30-2
 program design language, 13-1, 16-1
 program generator, 1-5
 program specifications, 1-6
 program types, 1-6
 program types cross reference, 30-3
 program types index, 30-2
 quick start application tool, 26-1
 source modifications, 16-1
 special characters, 9-1
 the detailed programming facility screen, 11-1
 user defined PDL, 34-1
 using the source code inventory and database, 28-1
Accessing
 CASE profiles, 4-5
 data item formula revisions, 15-4
 program generator, 6-1
 program generator options, 6-2
 program generator specifications, 5-1
 quick start CL generator, 25-1
 the logic module index, 31-3
Action diagram
 building, 27-1
 overview, 27-1
 viewing, 27-2
Adding
 new Q & A dialogue, 33-3
 PDL to a field, 15-4
Additional tools, 24-1
All help instructions, 31-2
Answer Entry screen (P98552), 33-3, 33-4
Application development cycle, 1-1
Arrays for EMK, @MK and @ER, B-1
Assignments in PDL, 14-5
Authorities for objects, 4-8

B

B0010 - Interactive Single Record Maintenance program, C-4
Blocks of statements, 14-3
Browse for screens or reports using quick start, 26-5
Browse screen, 11-13, 12-3
Building an action diagram, 27-1

C

C0010 - Batch Report with Totals program, C-4
C0020 - Batch Report with Totals and Subheadings program, C-5
C0025 - Batch Report with Totals and Subheadings program, C-6
CAP
 overview, 1-1
 status
 changing, 18-2
CASE
 benefits, 1-3
 menus, 1-7
 profiles
 accessing, 4-5
 understanding, 4-4
 program types, C-2
 specifications inquiry overview, 31-11
CASE Profiles screen (P98009), 4-5
CASE Specifications Inquiry screen (P93130), 31-11
Changing
 a dialogue, 33-8
 CAP status, 18-2
 program specifications, 26-7
Checklist
 data file design aid, A-1

- program generator, A-2
- report design aid, A-2
- screen design aid, A-1
- CL
 - overview, 19-1
 - program compile using quick start, 25-3
- Comments in PDL, 14-4
- Compiling
 - CL programs using quick start, 25-3
 - using quick start, 26-6, 26-8
- Computer Assisted Design, 1-1
- Computer Assisted Programming, 1-1
- Conditional directives, 32-11
- Conditions in PDL, 14-8
- Control language, 19-1
- Copy File screen (CPYF), 4-2
- Copying
 - a dialogue, 33-8
 - model control language, 19-1
 - program specifications, 31-8
- Create/Modify Logic Modules screen (P93001SEU), 31-6
- Create/Modify Program Types screen (P93001), 30-4, 34-1
- Creating
 - *ENTRY PLIST entries, 11-12
 - a partial KLIST for a file, 11-17
 - formula library entry, 31-7
 - JDESRC file for use with Program Generator, 4-2
 - logic modules, 31-6
 - report programs, 23-1
 - subheadings, 23-6
 - total formats, 23-3
 - user defined PDL, 34-1
- Customizing model control language, 19-2

D

- D0010 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number program, C-7
- D0020 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number program, C-9
- D0030 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record program, C-10
- D0040 - Interactive Subfile Maintenance with Action Code, with Options, by Key program, C-11
- D0050 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number program, C-12
- D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key program, C-14
- D0070 - Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number program, C-15
- D0080 - Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number program, C-16
- D0090 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance program, C-17
- D0100 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key program, C-19
- Data dictionary
 - editing disabled, 11-16
 - glossary update, 26-9
- Data fields to select using quick start, 26-4
- Data file design aid checklist, A-1
- Data Item Formula Revisions screen (P93109), 14-11, 15-3, 34-5
- Data structure
 - I00DSPROG source, D-1
 - I00SC source, D-2
- Database operations in PDL, 14-5
- Database update function for subfiles, 11-11
- Date and time standards, B-4
- Define Generator Specification screen (P93100M), 7-1, 8-3, 9-3, 10-1, 11-1, 12-4, 18-2, 18-3, 34-3
- Define Program Generator Specifications screen (P93100M), 6-2
- Defining
 - applications using quick start, 26-2
 - general instructions, 9-1
 - option and function exits, 10-1
 - processing options, 12-1
 - program purpose and type, 7-1
- Deleting a dialogue, 33-12
- Detailed information
 - CAD, 1-5
 - CAP, 1-5
- Detailed Programming Facility, 11-1, 21-2, 22-2
- Detailed Programming Facility screen (P93105), 11-2, 11-17, 15-4, 34-4
- Development libraries, 4-1
- Dialogue
 - changing, 33-8
 - copying, 33-8
 - deleting, 33-12
 - existing, 33-6
 - rename, 33-9
 - reviewing the flow, 33-7
 - running, 33-10
- Dialogue Copy screen (P98536), 33-8, 33-9
- Dialogue Descriptions screen (P98541), 33-3
- Dialogue Flow Revisions screen (P98531), 33-7
- Dialogue Lists screen (P98530), 33-7, 33-8, 33-9, 33-10
- Dialogue Selection screen (P98533), 7-4, 33-11
- Dialogue Test screen (P98535), 33-10, 33-12
- Dialogue Test screen (P98537), 33-13
- Directives
 - conditional, 32-11
 - exception, 32-10
 - functional, 32-1
 - substitution, 32-7
 - understanding, 32-1

Disable data dictionary editing, 11-16
Display Action Diagram screen (P92705), 27-3, 27-4
DREAM Writer considerations for report
formats, 23-7

E

E0010 - Interactive Window program, C-20
Edit screen, 9-2, 9-4
Editing, parsing, and source generation of PDL, 15-1
Enabling database update function for
subfiles, 11-11
ENTRY PLIST entries, 11-12
Error handling using arrays, B-1
Exception directives, 32-10

F

Features, 1-3
Field Definition screen, 23-4
Field protection, 11-14
File specifications, 8-1
File Specifications screen (P93102), 8-4, 8-8
Formula library entry for creating or
modifying, 31-7
Formula Library Entry screen (P93109), 14-10, 31-7
Full data field parameters
accessing, 11-4
understanding, 11-4
Full Data Field Parameters screen (P93125), 11-4,
11-9, 11-10, 11-12, 11-14, 11-15, 11-16
Function exits
previous profile, 4-8
Function exits set up, 10-1
Functional directives, 32-1
Functional servers
example, G-2
overview, G-1

G

General instructions for help text, 9-1
Generated source code
changing, 17-2
regenerating, 18-1
resolving errors, 18-3
Generator Updates screen (P9366), 31-11
Global program regeneration, 31-2
Glossary K, 30-4
Glossary revisions screen (P92001), 26-10
Glossary updates, 26-9
Guidelines for program types, C-2

H

Help file updates, 9-4
Help instructions
rebuild for a single program, 31-2
rebuild for all programs, 31-2
Help Instructions Master file (F98HELP), 9-2

I

I00DSPROG data structure source, D-1
I00SC data structure source, D-2
Interactive non-subfile program flow, E-3

J

Job queues, 4-3

K

Keywords
in blocks of statements, 14-3
in comments, 14-4
KLIST standards, B-3

L

Line structure, 29-1
Loading VCO description fields, 11-9
Locating
a dialogue flow, 33-7
screens or reports using quick start, 26-5
Logic Module Cross Reference screen (P93952), 31-4
Logic modules, 31-1
accessing the index, 31-3
creating or modifying, 31-6
detail, 31-2
maintaining, 31-5
viewing cross reference, 31-3
viewing op codes, 31-4
Logic translation feature, 27-5
Loops in PDL, 14-7

M

Maintaining logic modules
remove logic module, 31-5
resequence logic module, 31-5
Master dialogue questions, 33-1
Miscellaneous keywords and syntax in PDL, 14-10
Model Control Language Programs
customizing, 19-2
overview, 19-1
provided by JD Edwards World, 19-2
Modifying
formula library entry, 31-7
logic modules, 31-6
program specifications, 26-7
Moving program specifications, 31-8
Multi-member source file, 4-1

N

Naming convention standards, B-3
New Q & A dialogue, 33-3

O

Object authorities, 4-8

- job control, 4-8
- job queues, 4-8
- source file, 4-8
- source library, 4-8
- Op Codes screen (P93108), 31-4
- Operators in assignments, 14-5
- Option & Function exits screen (P93104), 10-1
- Options
 - defining, 10-1
 - overview, 10-1

P

- Parameter Copy/Move screen (P93890), 31-8
- Partial KLIST created for a file, 11-17
- PDL
 - editing, parsing, and source generation
 - parsing, 15-1
 - source code generation, 15-2
 - statements, 13-1
 - constants, 14-3
 - database files, 14-2
 - keywords, 14-2
 - operations, 14-2
 - operators, 14-2
 - punctuation, 14-3
 - variables, 14-2
 - user defined
 - creating, 34-1
- Prerequisites
 - common user defined codes, 3-2
 - Program Generator files, 3-1
 - source code for copy modules, 3-5
 - source code for JD Edwards World files, 3-5
- Primary module, 30-4
- Print Program Specification screen (P98300), 31-9
- Printing program generator specifications, 31-9
- Process for quick start, 26-2
- Processing options defined, 12-1
- Processing Options Setup screen (P98304), 12-5
- Program calls in PDL, 14-7
- Program code sample, F-1
- Program Design Language (PDL), 13-1, 16-1
- Program Generator
 - accessing, 6-1
 - checklist, A-2
 - files, 3-1
 - program design language, 3-2
 - source modifications/helps, 3-1
 - specifications, 3-1
 - merging updates, 31-11
 - printing specifications, 31-9
 - reviewing options, 6-2
 - specifications
 - accessing, 5-1
- Program purpose and type definition, 7-1
- Program Purpose and Type screen (P93100), 7-2, 18-2
- Program specifications
 - modifications using quick start, 26-7

- Program specifications to copy or move, 31-8
- Program Types
 - A0010 - Interactive Subfile Inquiry, C-2
 - A0020 - Interactive Single Record Inquiry, C-3
 - B0010 - Interactive Single Record Maintenance, C-4
 - C0010 - Batch Report with Totals, C-4
 - C0020 - Batch Report with Totals and Subheadings, C-5
 - C0025 - Batch Report with Totals and Subheadings, C-6
 - D0010 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, C-7
 - D0020 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number, C-9
 - D0030 - Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record, C-10
 - D0040 - Interactive Subfile Maintenance with Action Code, with Options, by Key, C-11
 - D0050 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number, C-12
 - D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key, C-14
 - D0070 - Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number, C-15
 - D0080 - Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number, C-16
 - D0090 - Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance, C-17
 - D0100 - Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key, C-19
 - E0010 - Interactive Window, C-20
 - guidelines, C-2
 - overview, C-2
 - X0010 - Batch Update with Report, C-21
 - X0020 - Batch Update, C-22
 - X0030 - Batch Update with Subroutine S001, C-23
 - X0040 - Batch Update with Report, C-23
 - Y0010 - Conversion, Two Files with Error Report, C-24
 - Y0020 - Conversion, One File Update with Error Report, C-25
 - Y0030 - Conversion, One File Write with Error Report, C-26
- Program types
 - B0010 example, F-1
 - conversion, 1-7
 - creating or modifying, 30-1
 - cross reference, 30-3
 - index, 30-2
 - interactive, 1-6

- report, 1-6
- server, 1-7
- window, 1-6
- Program Types Index screen (P93900), 30-2
- Program Types X-Reference screen (P93953), 30-3
- Programming standards, B-1
- Programs and IDs
 - CPYF (copy file), 4-2
 - P00051 (user defined code revisions), 3-2
 - P2710 (translation table), 27-5
 - P92001 (glossary revisions), 26-10
 - P92705 (display action diagrams), 27-3, 27-4
 - P93001 (create/modify program types), 30-4, 34-1
 - P93001SEU (create/modify logic modules), 31-6
 - P93100 (program purpose and type), 7-2, 18-2
 - P93100M (define generator specification), 7-1, 8-3, 9-3, 10-1, 11-1, 12-4, 18-2, 18-3, 34-3
 - P93100M (define generator specifications), 6-2
 - P93102 (file specifications), 8-4, 8-8
 - P93104 (option & function exits), 10-1
 - P93105 (detailed programming facility), 11-2, 11-17, 15-4, 34-4
 - P93108 (logic module op codes), 31-4
 - P93109 (data item formula revisions), 14-11, 15-3, 15-4, 34-5
 - P93109 (formula library entry), 14-10, 31-7
 - P93125 (full data field parameters), 11-4, 11-9, 11-10, 11-12, 11-14, 11-15, 11-16
 - P93130 (CASE specifications inquiry), 31-11
 - P93513 (quick start application tool), 26-2
 - P93513J (quick start CL generator), 25-1
 - P93515V (quick start application tool), 26-4, 26-5, 26-6, 26-7, 26-8, 26-9, 26-10
 - P93515V (quick start CL generator), 25-3
 - P9366 (generator updates), 31-11
 - P93890 (parameter copy/move), 31-8
 - P93900 (program types index), 30-2
 - P93952 (logic module cross reference), 31-4
 - P93953 (program types x-reference), 30-3
 - P98009 (CASE profiles), 4-5
 - P9801 (software versions repository), 4-5, 8-8
 - P98300 (build action diagrams), 27-1
 - P98300 (print program specification), 31-9
 - P98304 (processing options setup), 12-1
 - P98529 (simple question & answer), 33-1
 - P98530 (dialogue lists), 33-7, 33-8, 33-9, 33-10
 - P98531 (dialogue flow revisions), 33-7
 - P98533 (dialogue selection), 7-4, 33-11
 - P98534 (quiz answer review), 33-11
 - P98535 (dialogue test), 33-10, 33-12
 - P98536 (dialogue copy), 33-8, 33-9
 - P98537 (dialogue test), 33-13
 - P98541 (dialogue descriptions), 33-3
 - P98551 (question entry), 33-3
 - P98552 (answer entry), 33-3, 33-4
 - RM/M (remove member), 4-3
- Project management, 4-3
- Protecting fields from being cleared, 11-14

Q

- Q & A dialogue, 33-3
- Question and answer system overview, 33-1
- Question Entry screen (P98551), 33-3
- Questions in a master dialogue, 33-1
- Quick reference of program types, C-2
- Quick Start
 - application tool overview, 26-1
 - CL generator for creating programs, 21-1, 22-2, 25-1
 - generating subfile inquiry programs, 21-2
 - generator to create subfile maintenance programs, 22-2
 - steps for process, 26-2
- Quick Start Application Tool screen (P93513), 26-2
- Quick Start Application Tool screen (P93515V), 26-4, 26-5, 26-6, 26-7, 26-8, 26-9, 26-10
- Quick Start C L Generator screen (P93515V), 25-3
- Quick Start CL Generator screen (P93513J), 25-1
- Quiz Answer Review screen (P98534), 33-11
- Quiz to determine program type, 33-12

R

- RDA, 23-1
- Rebuild help instructions, 31-2
- Record Formats List screen, 23-6
- Regenerating source code, 31-2
- Remove logic module, 31-5
- Remove Member (RM/M), 4-3
- Renaming a dialogue, 33-9
- Report Design Aid
 - creating reports, 23-1
- Report design aid
 - checklist, A-2
- Report format considerations for DREAM Writer, 23-7
- Report program with subheadings flow, E-6
- Report program without subheadings flow, E-5
- Report programs subheadings and totals, 23-1
- Reports or screens
 - browse or update using quick start, 26-5
 - compiling using quick start, 26-6
- Resequence logic module, 31-5
- Reviewing
 - dialogue flow, 33-7
 - questions, 33-1
 - source modifications, 31-10
- RPG subroutines, E-1
- Running a dialogue, 33-10
- Running a quiz, 33-12

S

- Sample program code, F-1
- Screen design aid checklist, A-1
- Screens
 - Answer Entry, 33-3, 33-4
 - Browse, 11-13, 12-3
 - Build Action Diagrams, 27-1

- CASE Profiles, 4-5
- CASE Specifications Inquiry, 31-11
- Copy File (CPYF), 4-2
- Create/Modify Logic Modules, 31-6
- Create/Modify Program Types, 30-4, 34-1
- Data Item Formula Revisions, 14-11, 15-3, 15-4, 34-5
- Define Generator Specification, 7-1, 8-3, 9-3, 10-1, 11-1, 12-4, 18-2, 18-3, 34-3
- Define Generator Specifications, 6-2
- Detailed Programming Facility, 11-2, 11-17, 15-4, 34-4
- Dialogue Copy, 33-8, 33-9
- Dialogue Descriptions, 33-3
- Dialogue Flow Revisions, 33-7
- Dialogue Lists, 33-7, 33-8, 33-9, 33-10
- Dialogue Selection, 7-4, 33-11
- Dialogue Test, 33-10, 33-12, 33-13
- Display Action Diagram, 27-3, 27-4
- Edit, 9-2, 9-4
- Field Definition, 23-4
- File Specifications, 8-4, 8-8
- Formula Library Entry, 14-10, 31-7
- Full Data Field Parameters, 11-4, 11-9, 11-10, 11-12, 11-14, 11-15, 11-16
- Generator Updates, 31-11
- Glossary, 26-10
- Logic Module Cross Reference, 31-4
- Op Codes, 31-4
- Option & Function exits, 10-1
- Parameter Copy/Move, 31-8
- Print Program Specification, 31-9
- Processing Options Setup, 12-5
- Program Purpose and Type, 7-2, 18-2
- Program Types Cross Reference, 30-3
- Program Types Index, 30-2
- Question Entry, 33-3
- Quick Start Application Tool, 26-2, 26-4, 26-5, 26-6, 26-7, 26-8, 26-9, 26-10
- Quick Start CL Generator, 25-1, 25-3
- Quiz Answer Review, 33-11
- Record Formats List, 23-6
- Remove Member, 4-3
- Simple Question & Answer, 33-1
- Software Versions Repository, 4-5, 8-8
- Source Entry Utility, 19-1
- Translation Table, 27-5
- User Defined Code Revisions, 3-2
- Screens or reports
 - browse or update using quick start, 26-5
 - compiling using quick start, 26-6
- Selecting data fields using quick start, 26-4
- Serial numbers, 29-1, 29-2
- Simple Question & Answer screen (P98529), 33-1
- Software Action Request (SAR), 4-3
- Software Versions Repository screen (P9801), 4-5, 8-8, 19-1
- Solving generation problems, 18-3
- Source code
 - for copy modules, 3-5

- for JD Edwards World World files, 3-5
- inventory and database, 28-1
- regenerating, 18-1
- when to regenerate, 18-1
- Source Entry Utility screen, 19-1
- Source Listings, D-1
- Source modifications
 - about, 16-1
 - code review, 31-10
- Source sequence line numbers, 29-1
- Source sequence line structure, 29-1
- Source serial numbers, 29-1
- Special characters
 - about, 9-1
 - within the help file, 9-2
- Standards for programming, B-1
- Standards using functional servers, G-1
- Statements in PDL, 13-1
- Structure of serial numbers, 29-2
- Subfile Inquiry Program
 - components, 21-1
 - overview, 21-1
- Subfile Maintenance Program
 - components, 22-1
 - overview, 22-1
 - special considerations, 22-2
- Subfile program with options flow, E-4
- Subfiles database update function, 11-11
- Subheadings
 - formats, 23-6
 - overview, 23-1
- Submit program to compile using quick start, 26-8
- Subroutines for RPG, E-1
- Substitution directives, 32-7
- Syntax
 - in assignments, 14-5
 - in blocks of statements, 14-3
 - in comments, 14-4
- System integration, 1-1
 - application development cycle, 1-1
 - fundamentals, 1-2
 - history of program generator, 1-2
 - specifications, 1-2

T

- Terms and concepts, 1-4
- Totaling
 - overview, 23-1
- Translation Table screen (P92710), 27-5

U

- Understanding
 - assignments, 14-5
 - operator and syntax, 14-5
 - rules, 14-5
- blocks of statements, 14-3
 - keywords and syntax, 14-3
 - rules, 14-3

- CASE profiles, 4-4
- comments
 - keywords and syntax, 14-4
 - rules, 14-4
- conditions
 - keywords and syntax, 14-8
 - rules, 14-9
 - symbols, 14-9
- database operations
 - keywords and syntax, 14-5
 - rules, 14-6
- directives, 32-1
- full data field parameters, 11-4
- loops
 - keywords and syntax, 14-7
 - rules, 14-8
- miscellaneous keywords and syntax
 - keywords and syntax, 14-10
 - rules for include, 14-10
 - rules for return, 14-12
- program calls
 - keywords and syntax, 14-7
 - rules, 14-7
- source sequence line numbers, 29-1
 - source sequence line structure, 29-1
 - source serial numbers, 29-1
 - structure of the serial number, 29-2
- Updating
 - data dictionary, 26-9
 - glossary, 26-9
 - help file, 9-4
 - screens or reports using quick start, 26-5
- Usage indicator standards, B-2
- User Defined Code Revisions screen (P00051), 3-2
- User Defined Codes, 3-2
- User-provided prerequisites, 4-1
- Using
 - CASE specifications inquiry, 31-11
 - program generator updates, 31-11

V

- VCO description fields for screens, 11-9
- Viewing
 - an action diagram, 27-2
 - logic module cross reference, 31-3
 - logic module op codes, 31-4

W

- Work field standards, B-4
- Working with
 - file specifications, 8-1
 - the question and answer system, 33-1
 - user provided prerequisites
 - development libraries, 4-1
 - job queues, 4-3
 - multi-member source file, 4-1
 - overview, 4-1
 - project management, 4-3

X

- X0010 - Batch Update with Report program, C-21
- X0020 - Batch Update program, C-22
- X0030 - Batch Update with Subroutine S001
 - program, C-23
- X0040 - Batch Update with Report program, C-23

Y

- Y0010 - Conversion, Two Files with Error Report
 - program, C-24
- Y0020 - Conversion, One File Update with Error
 - Report program, C-25
- Y0030 - Conversion, One File Write with Error Report
 - program, C-26

