



JD Edwards World

CASE Guide

Release A9.2

Revised – May 1, 2009

Copyright Notice

Copyright © 2010, Oracle. All rights reserved.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

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.

Subject to patent protection under one or more of the following U.S. patents: 5,781,908; 5,828,376; 5,950,010; 5,960,204; 5,987,497; 5,995,972; 5,987,497; and 6,223,345. Other patents pending.

Warranty Disclaimer

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.

Restricted Rights Notice

If this software or related documentation 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 RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software 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 which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software 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.

Contains GNU libgmp library; Copyright 1991 Free Software Foundation, Inc. This library is free software which can be modified and redistributed under the terms of the GNU Library General Public License.

Includes Adobe PDF Library, Copyright 1993-2001 Adobe Systems, Inc. and DL Interface, Copyright 1999-2008 Datalogics Inc. All rights reserved. Adobe is a trademark of Adobe Systems Incorporated.

Portions of this program contain information proprietary to Microsoft Corporation. Copyright 1985-1999 Microsoft Corporation.

Portions of this program contain information proprietary to Tenberry Software, Inc. Copyright 1992-1995 Tenberry Software, Inc.

Portions of this program contain information proprietary to Premia Corporation. Copyright 1993 Premia Corporation.

This product includes code licensed from RSA Data Security. All rights reserved.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (ey@cryptsoft.com).

This product includes software written by Tim Hudson (tjh@cryptsoft.com). All rights reserved.

This product includes the Sentry Spelling-Checker Engine, Copyright 1993 Wintertree Software Inc. All rights reserved.

Open Source Disclosure

Oracle takes no responsibility for its use or distribution of any open source or shareware software or documentation and disclaims any and all liability or damages resulting from use of said software or documentation. The following open source software may be used in Oracle's JD Edwards World products and the following disclaimers are provided.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). Copyright (c) 1999-2000 The Apache Software Foundation. All rights reserved. THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Contents

1 Overview

Overview to Computer Aided Software Engineering (CASE)	1-1
System Integration.....	1-1
Features	1-3
Terms and Concepts	1-4
Detailed Information	1-5
Menu Overview	1-8

2 Foundation

Foundation	2-1
About Foundation Information	2-1
Work with Prerequisites JD Edwards World Provides.....	2-3
Working with Prerequisites JD Edwards World Provides.....	2-3
Work with User-Provided Prerequisites	2-9
Development Libraries.....	2-9
Multi-member Source File (JDESRC)	2-9
Job Queues	2-11
Project Management	2-11
CASE Profiles	2-12
Object Authorities	2-17

3 Program Generator

Overview to Program Generator	3-1
About Program Generator Steps.....	3-1
Access Program Generator Specifications	3-3
Function Exits	3-6

Define Program Purpose and Type	3-9
Work with File Specifications	3-17
Define General Instructions	3-27
Define Option and Function Exits.....	3-33
Work with the Detailed Programming Facility.....	3-39
About the Detailed Programming Facility.....	3-39
About Full Data Field Parameters.....	3-42
Loading VC0 Description Fields	3-47
Enabling the Database Update Function for Subfiles.....	3-51
Creating *ENTRY PLIST Entries	3-52
Protecting Fields from Being Cleared.....	3-54
Disabling Data Dictionary Edits.....	3-55
Creating a Partial KLIST for a File	3-56
Define Processing Options	3-57
Defining Processing Options	3-60
4 Program Design Language	
Overview to Program Design Language	4-1
About PDL.....	4-1
About PDL Statements and Syntax	4-3
About PDL Statements	4-3
About Blocks of Statements	4-5
About Comments	4-6
About Assignments.....	4-7
About Database Operations.....	4-8
About Program Calls	4-9
About Loops.....	4-10
About Conditions	4-11
About Miscellaneous Keywords and Syntax.....	4-13
Understand Additional PDL Operations	4-17
Editing.....	4-17
Parsing	4-17
Source Code Generation.....	4-17
Data Item Formula Examples	4-18

Work with Data Item Formula Revisions	4-21
5 Source Modifications	
Overview to Source Modifications	5-1
About Source Modifications.....	5-1
Change Generated Source Code	5-3
Regenerate Source Code.....	5-7
When to Regenerate Source Code	5-7
Changing CAP Status.....	5-8
Resolving CAP Status Invalid Error.....	5-9
Work with Model Control Language Programs.....	5-11
JD Edwards World Model CL Programs	5-12
6 CASE Programs	
Overview to CASE Programs	6-1
About CASE Programs	6-1
Overview to Subfile Inquiry Programs	6-3
Overview to Subfile Maintenance Programs.....	6-5
Create Report Programs	6-7
Understanding RDA Special Use Fields	6-7
Creating a Total Format	6-9
Defining a Subheading.....	6-12
Understanding DREAM Writer Considerations	6-14
7 Additional Tools	
Overview to Additional Tools	7-1
About Additional Tools	7-1
Work with Quick Start CL Generator	7-3
Work with the Quick Start Application Tool	7-7
Defining the Application	7-8
Selecting Data Fields	7-11
Browsing or Updating the Screens or Reports (Optional)	7-11

Compiling the Screens or Report (Optional)	7-12
Modifying Specifications (Optional).....	7-13
Submitting the Program to Compile (Optional)	7-14
Updating the Data Dictionary and Glossary	7-15

Work with Action Diagramming.....7-19

Building an Action Diagram.....	7-19
Viewing an Action Diagram	7-20
Accessing the Logic Translation Feature.....	7-24

8 Source Code Inventory and Database

Overview to Source Code Inventory and Database8-1

About the Source Code Inventory and Database.....	8-1
---	-----

Understand Source Sequence8-3

Working with Program Types8-7

Reviewing Abbreviations for Program Types.....	8-7
Reviewing Program Types Index.....	8-8
Reviewing Program Types Cross Reference.....	8-9
Creating or Modifying Program Types.....	8-10

Work with Logic Modules8-13

Viewing the Logic Module Index.....	8-14
Viewing Logic Module Cross Reference	8-15
Viewing Logic Module Op Codes.....	8-16
Maintaining the Logic Module File.....	8-17
Creating or Modifying Logic Modules.....	8-18
Creating or Modifying Formula Library Entry	8-19
Copying or Moving Program Specifications	8-20
Printing Program Generator Specifications	8-21
Reviewing Source Modifications.....	8-22
Using Program Generator Updates	8-23
Using CASE Specifications Inquiry.....	8-23

Understand Directives8-25

Work with the Question and Answer System.....8-37

Reviewing Questions in a Master Dialogue	8-37
Adding New Q & A Dialogue	8-39
Working with an Existing Dialogue	8-42

Create User Defined PDL.....	8-51
9 Appendices	
Appendix A - Program Generator Checklist.....	9-1
Appendix B - Programming Standards	9-5
Appendix C - CASE Program Types	9-9
Appendix D - Source Listings	9-39
Appendix E - JD Edwards World Subroutines and Flows.....	9-79
Subroutines.....	9-79
Flows.....	9-82
Appendix F – Sample Code.....	9-87
Appendix G - Functional Servers.....	9-99

1 Overview

Overview to Computer Aided Software Engineering (CASE)

This section contains the following:

- [System Integration](#)
- [Features](#)
- [Terms and Concepts](#)
- [Detailed Information](#)
- [Menu Overview](#)

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.

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.

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.

History of the Program Generator

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

Evolution of the Program Generator

Clone II Programs

- Dynamic Data Dictionary

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

Clone II.5 Programs

- Started in 1989
- Dynamic totaling and page skipping - creates 80 lines of code for ALL fields
- No more “?” code generated
- Cursor Sensitive Help (F1)
- F24 Window
- Code for subfile option processing generated

Current Program Generator Programs

- More utilization of file servers
- Use of the Program Design Language (PDL) allow you to make modifications instead of making changes through Source Entry Utility

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.

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.

Terms and Concepts

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.

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.

Detailed Information

CASE Profile

See the *Advanced Programming Concepts and Skills (APCS) 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.

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)

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

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

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.

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.

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.

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

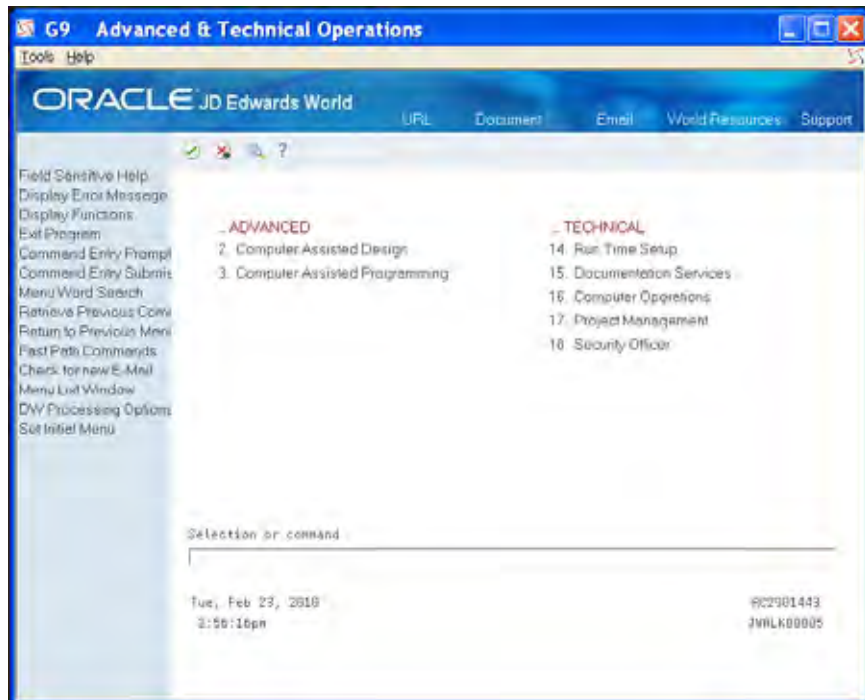
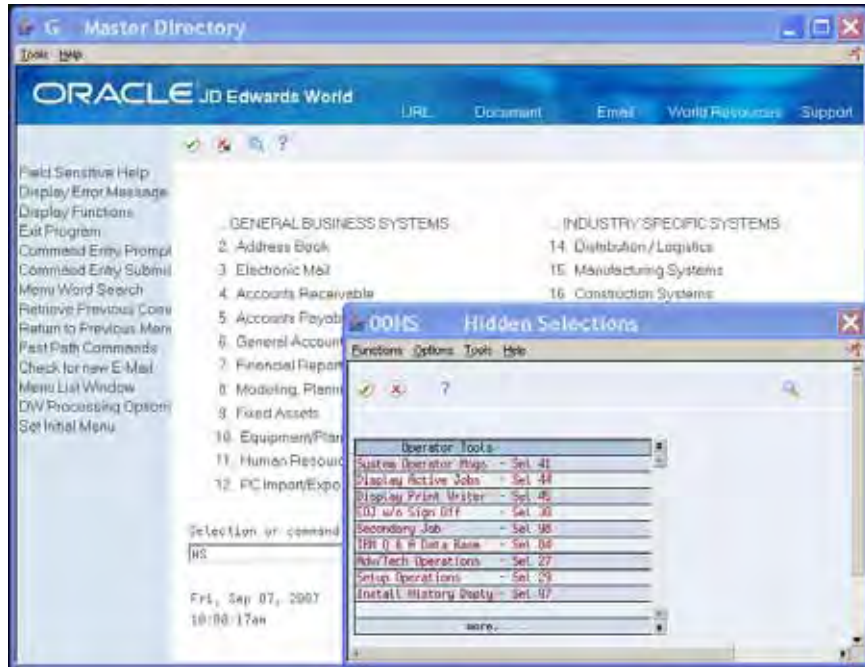
Conversion

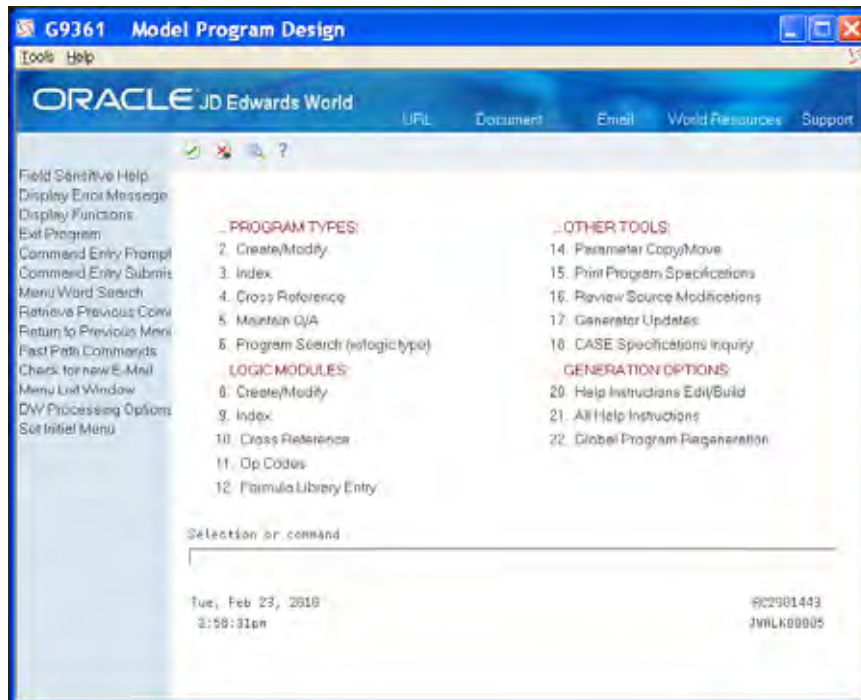
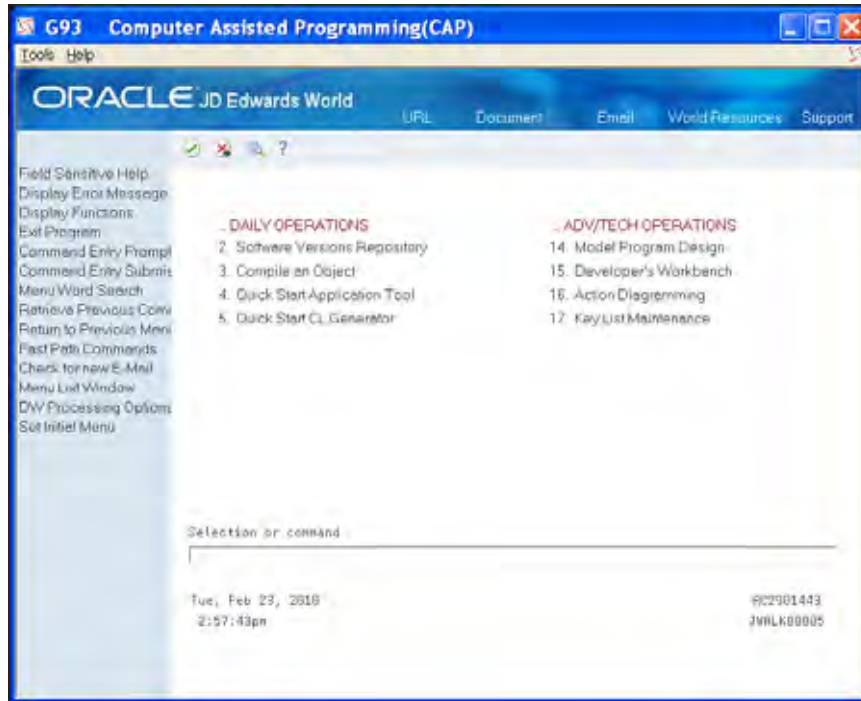
- Use this program type to convert data from one file to another
- Can contain a report

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.

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.





2 Foundation

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

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

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

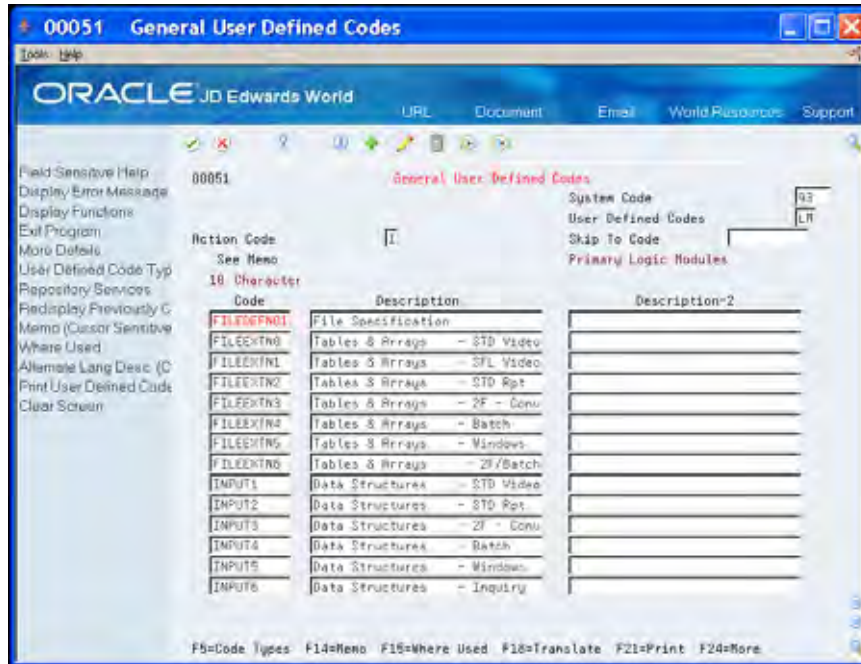
The following files are database files and include data.

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

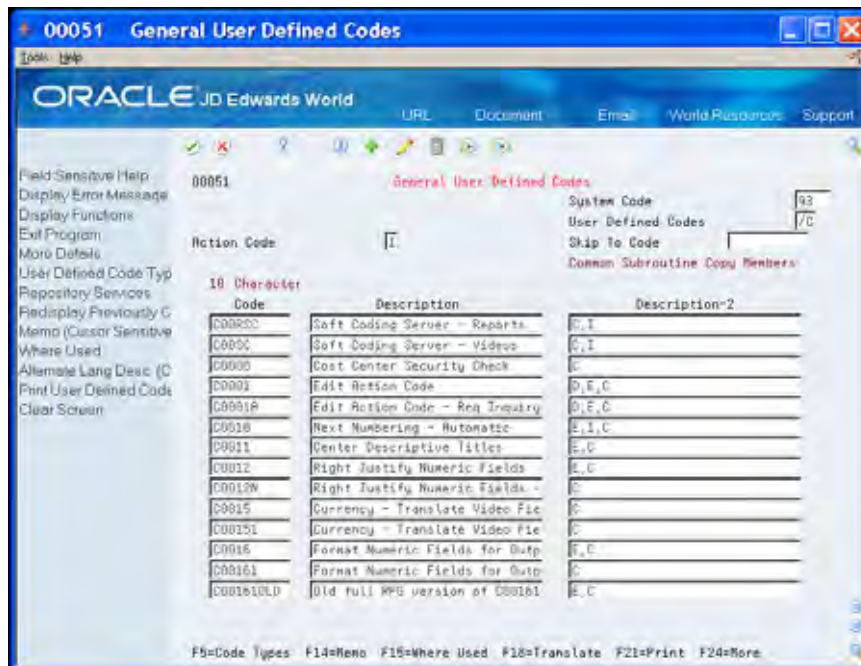
Common UDCs

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.



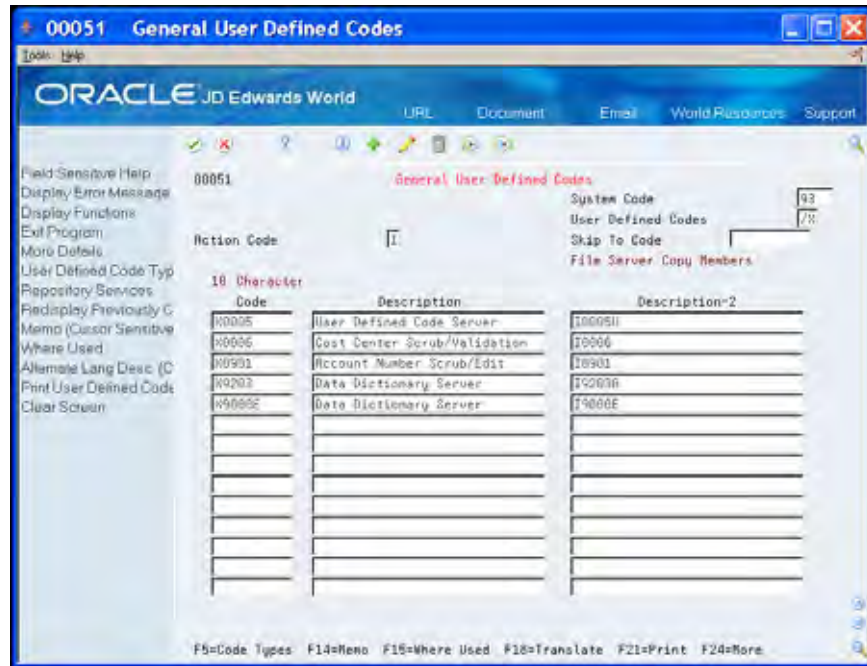
- Common Subroutine Copy Members, 93//C. Lists all of the copy modules on the system. Description-2 field contains any additional copy modules that are necessary to make the common subroutine function properly. For example, C0012 requires copy module E0012.



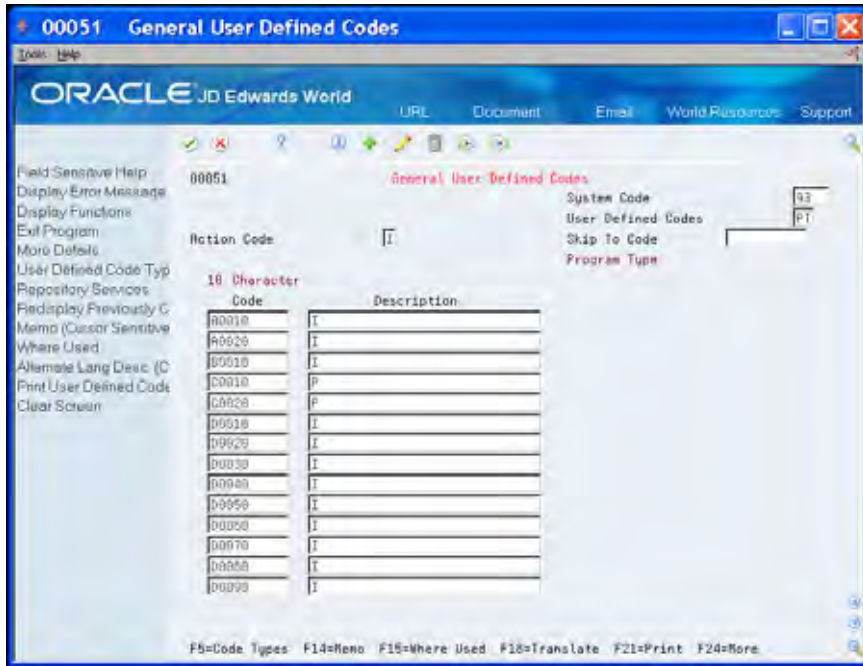
Alphanumeric Code	Type of Copy Module
D	Copy the member into the F specifications

Alphanumeric Code	Type of Copy Module
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.



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



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.

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

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 the following additional prerequisites:

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

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

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 (JDFSRC).

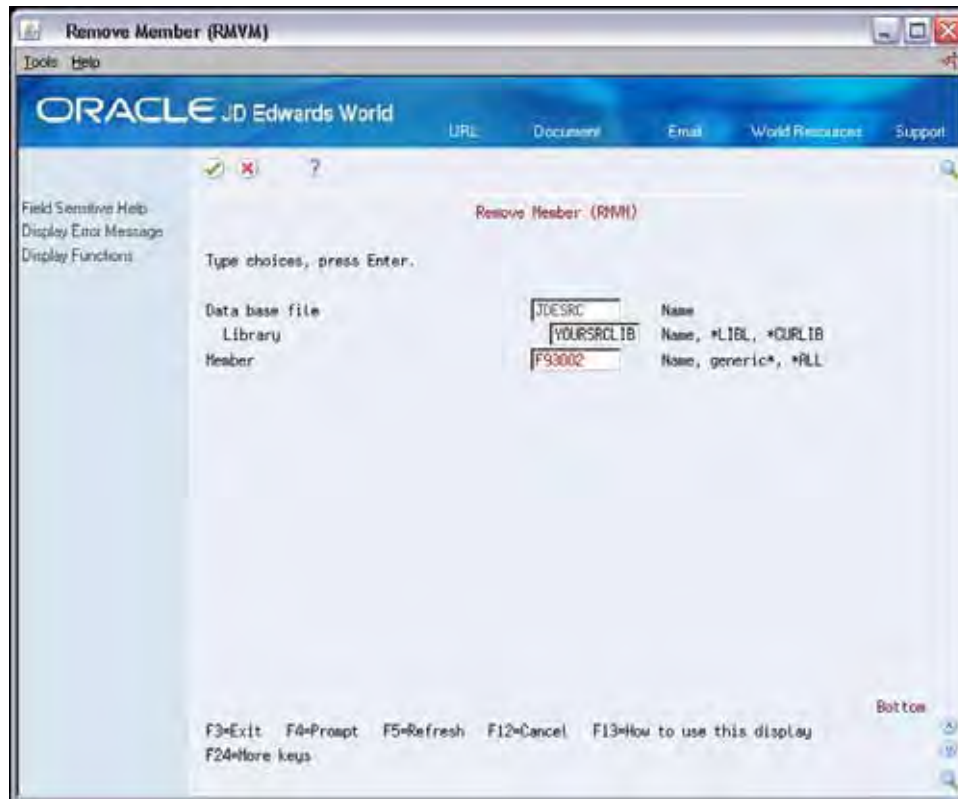
Do not use CRTSRCPF as it has only three fields in it, Date, Time, and Data, and the Program Generator requires extra fields.



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)
```

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

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.

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 the *Advanced Programming Concepts and Skills (APCS) 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.

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.

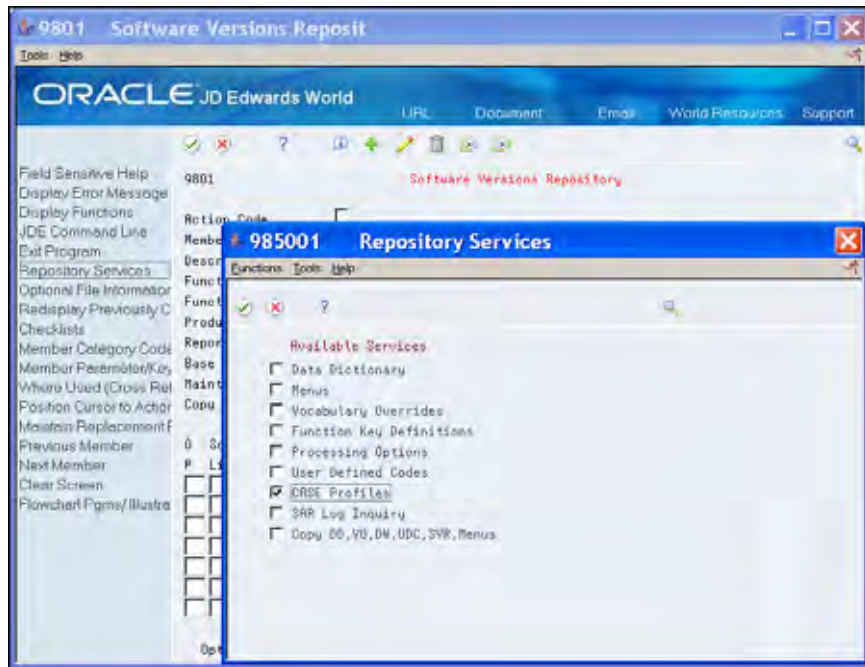
The system uses the values in the SAR Number and SAR Delivery fields to determine what type of SAR logging should occur.

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

	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.



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.



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.
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.

Field	Explanation
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.

Function Exits

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

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.

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.

Source Library

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

Source File

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

Job Queues

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

3 Program Generator

Overview to Program Generator

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

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 chapter includes the following tasks:

- [To access the Program Generator](#)
- [To access Program Generator options](#)

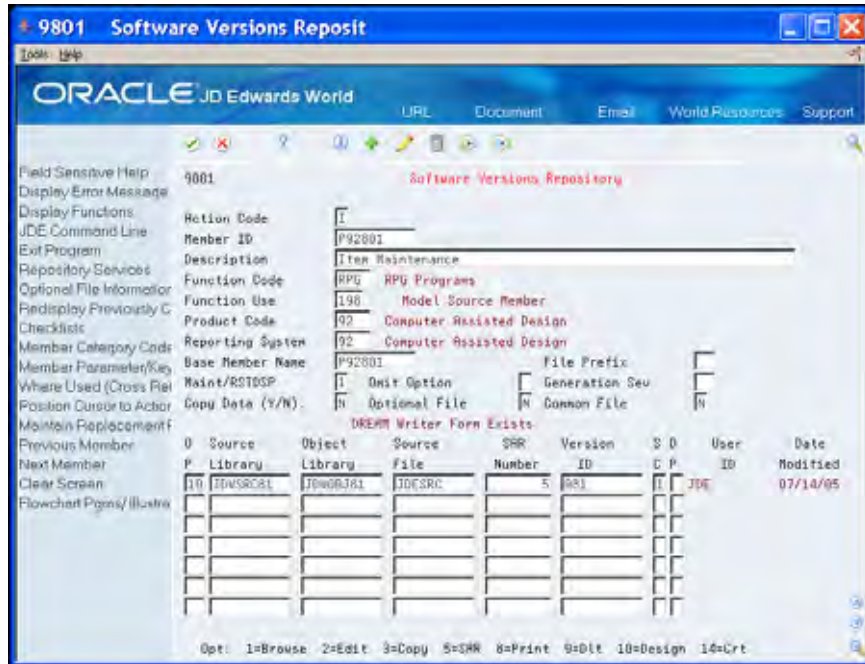
To access the Program Generator

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



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

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



2. Enter 10 (Design) next to the environment in the following field:
 - Option

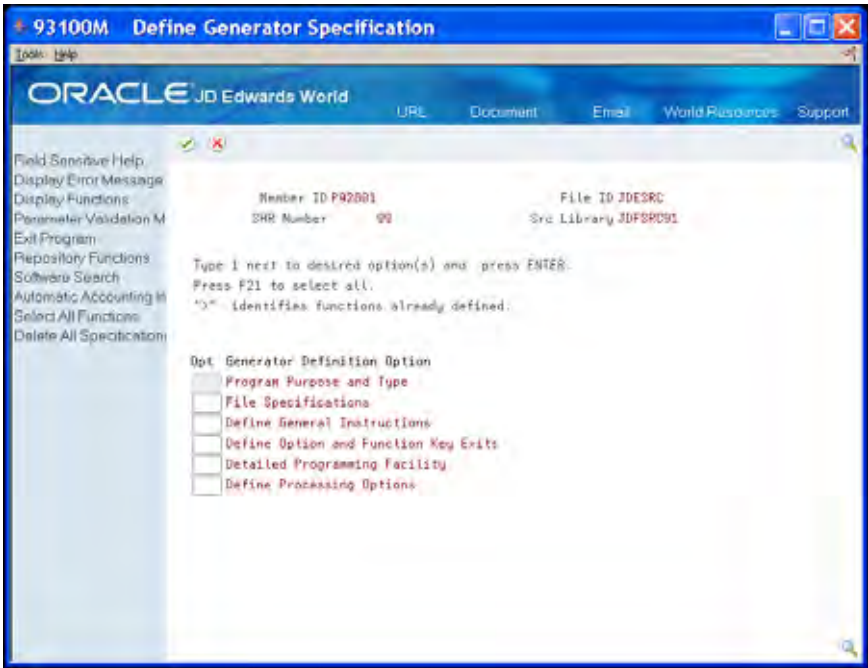
The Program Generator Specification screen displays.



To access Program Generator options

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

- Option



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.

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

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 Pxxxx 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

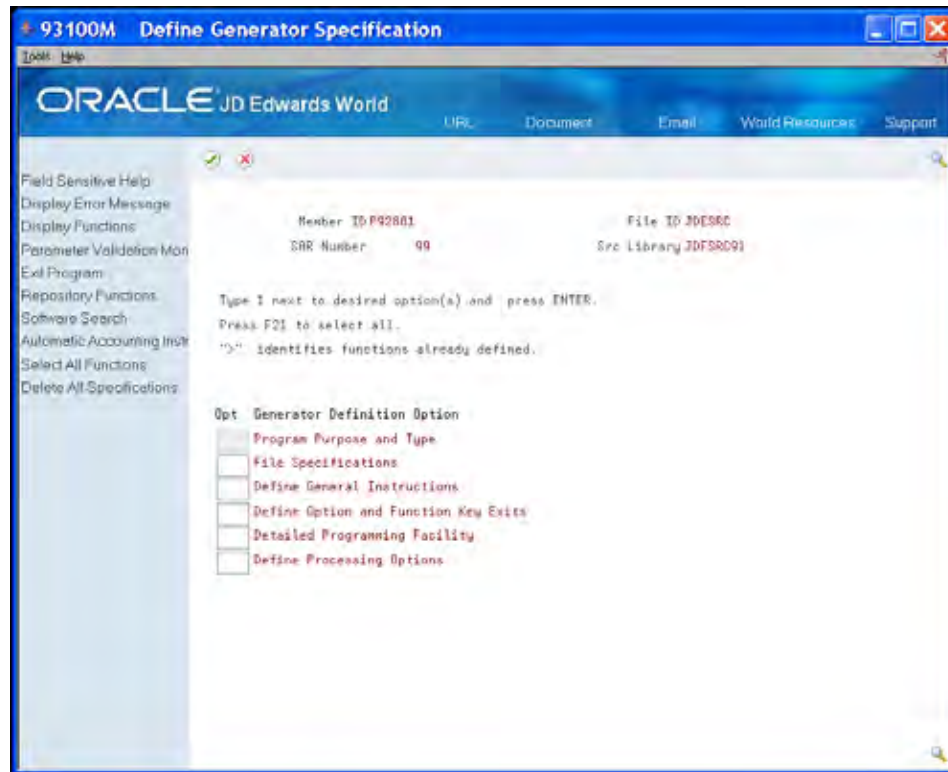
The Dialogue Selection screen is a series of questions you use to determine the Program Type.

This chapter includes the following tasks:

- [To define program purpose and type](#)
- [To identify 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



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

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	The title defaults to the description in the Software Versions Repository and should not be changed. When help instructions are generated, this title appears as the Help program title. Serves as the alpha description for the data item previously mentioned.
Purpose	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 style="text-align: center;"><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>

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.

To identify program type

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



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

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

Interactive



Interactive form

E0010

Print a report

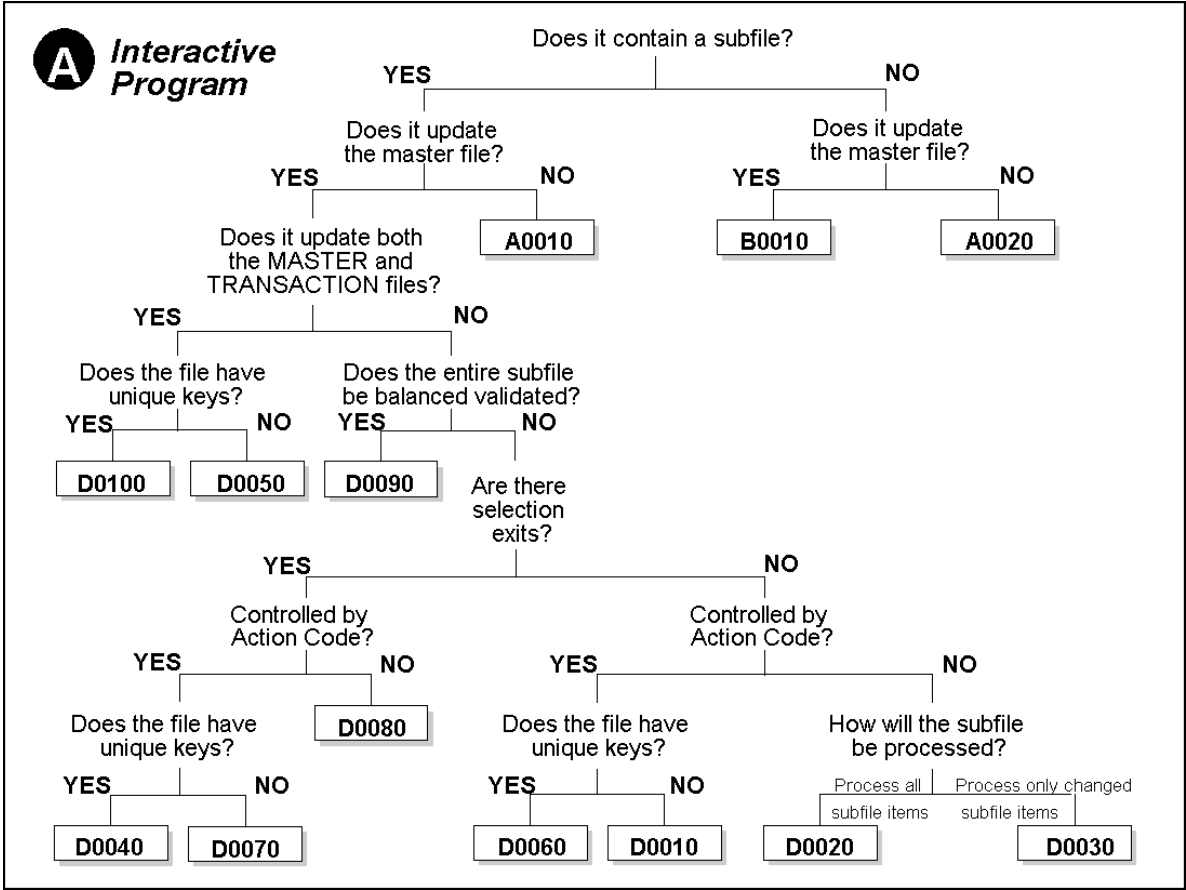


Conversion program

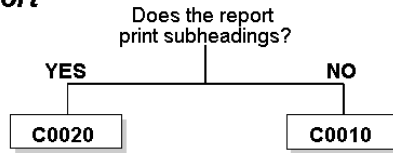


Batch update program

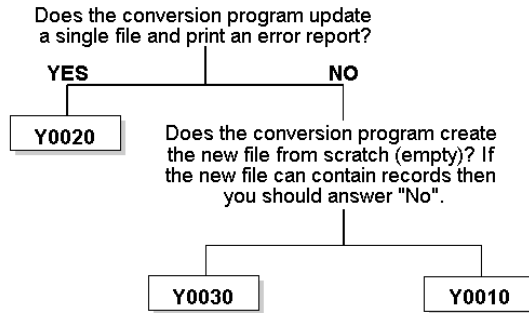




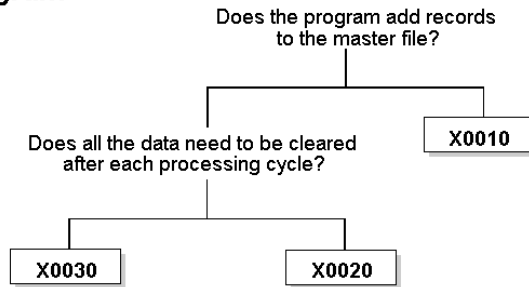
B *Print a Report*



C *Conversion Program*



D *Batch Update Program*



Work with File Specifications

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

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

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 Maintenance	Specify the master file with an M or 1 in the Update field
D0040	SFL Maintenance – KEY	
D0045	SFL Maintenance – KEY, No Action Code	
D0060	SFL Maintenance – KEY	
X0010	Batch Update – 1 File	
Y0020	File Conversion – 1 File	
Y0030	File Conversion – 1 File	
D0100	SFL Maintenance – KEY, 2 Update Files	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	
D0070	No Action code	
D0080	SFL Maintenance – RRN	
D0090	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.

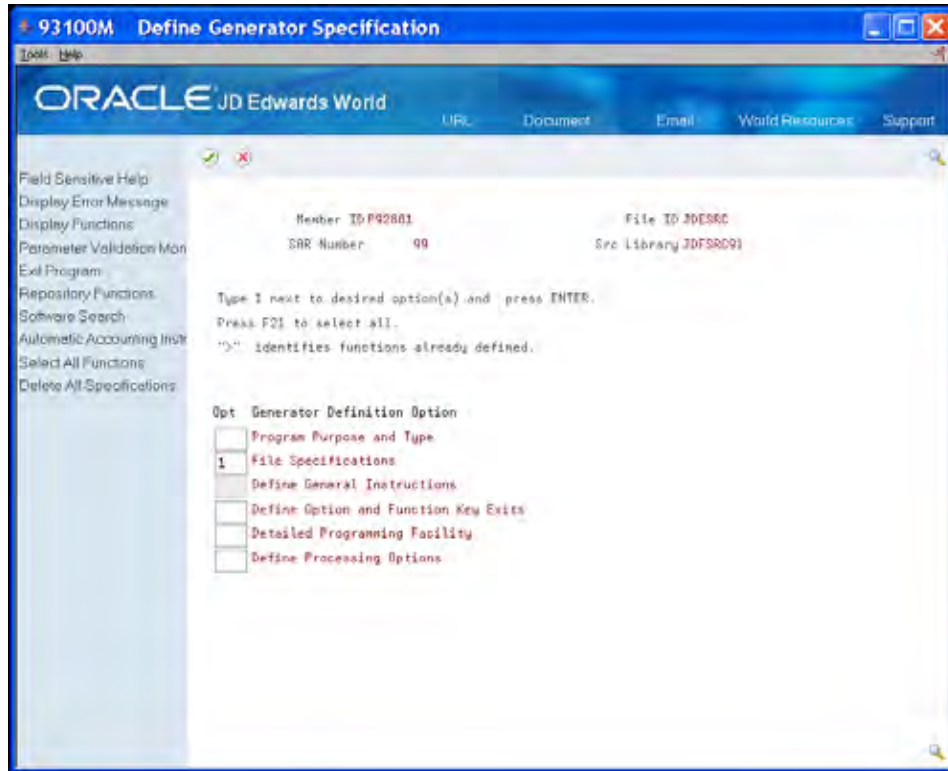
After you complete the appropriate fields on the File Specifications screen, a job runs interactively that analyzes the file specifications and creates records for three Program Generator files. As the system processes this job, messages display at the bottom of the screen.

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.

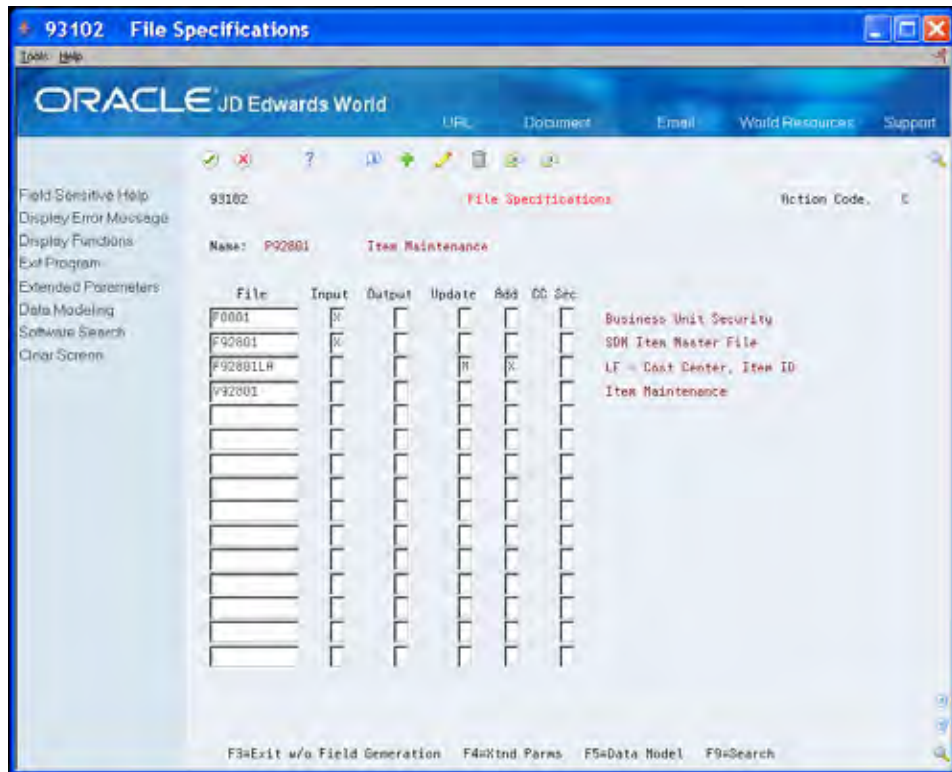
Note: If you make changes to the fields in any of the files you list in the File Specifications, you must run the File Specifications again. If you delete a field from a file, you must manually delete that field from the Detailed Programming Facility. Running the File Specifications again does not remove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

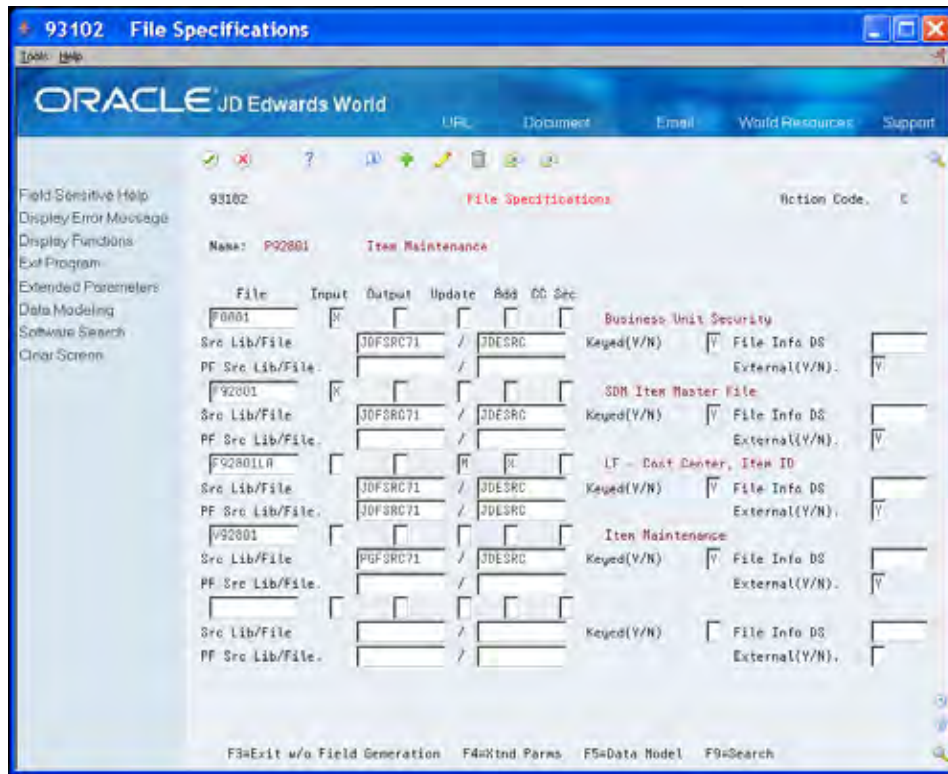


2. On File Specifications, complete the following field:
 - File
3. Complete the appropriate field:
 - Input
 - Output
 - Update
 - Add



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.



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 – Update primary file S or X – Update secondary file 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. ▪ 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.

Field	Explanation
PF Src Lib/File	Library where the source resides for the physical file linked to the logical file.
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>

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)

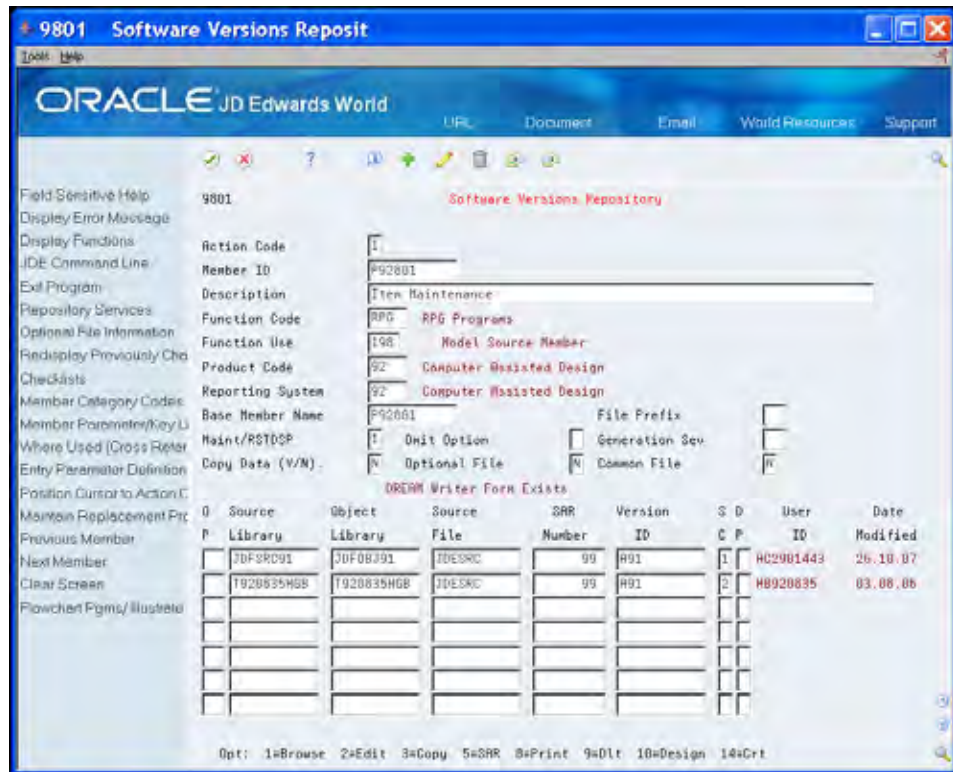
Choose Search (F9) to access the Software Search facility. On the Software Search facility, you enter a program name to view all programs that meet or are greater than the search criteria.

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.



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

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).

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.

Special Characters within Help Instructions

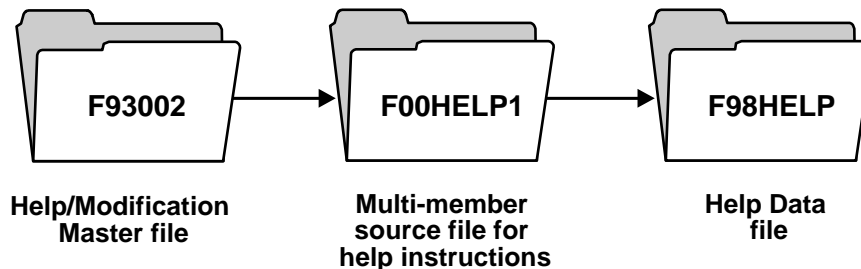
You can use the special characters to display fields with specific attributes. You begin and end the text selection as follows.

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).



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:



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

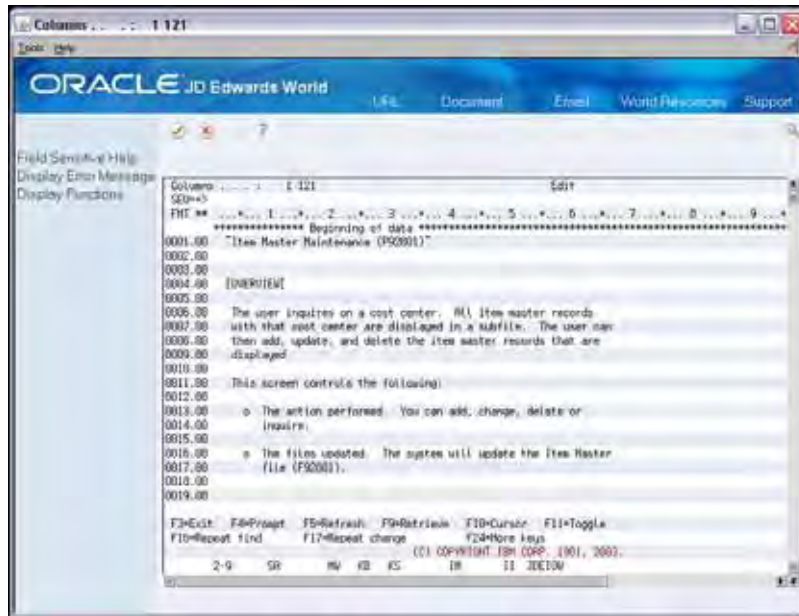
1. On Define Generator Specification, enter 1 in the following field next to Define General Instructions:

- Option



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.

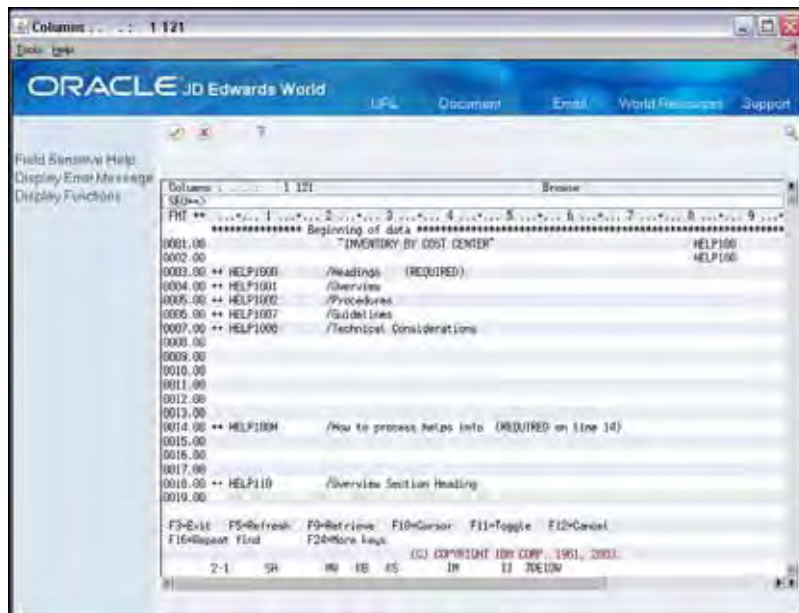


To update the help instructions

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



Define 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.

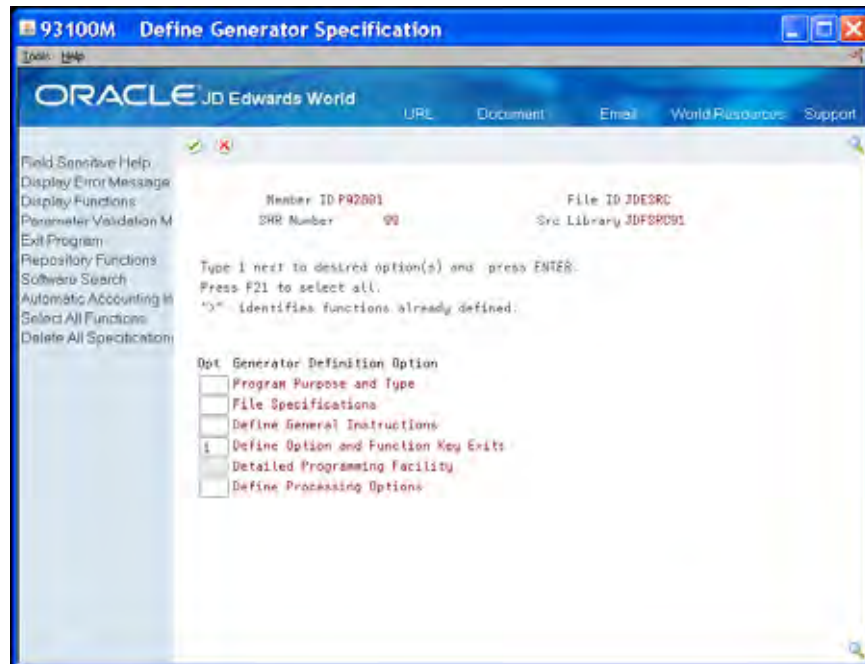
The system stores information for CASE in the Program Exit Parameters file (F93104).

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.

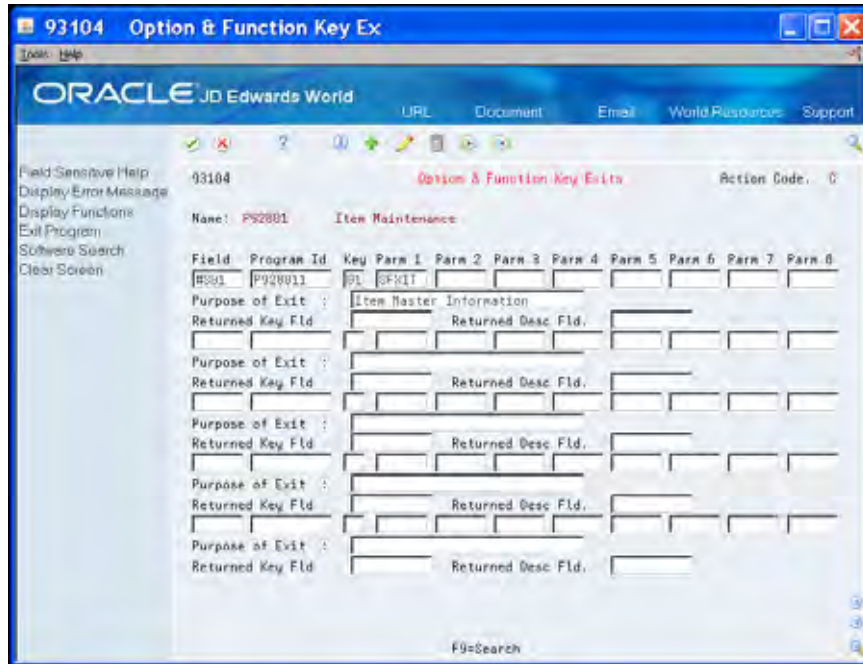
To define option and function exits

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



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



Field

Explanation

Field

The RPG field name (6 bytes) to be passed as a parameter on function key exits or subfile options.

Screen-specific information

The internal field name the system assigns to each option and function exit in the program you are generating.

Correlation exists between this field and the Function exit Definitions repository.

Maintained in the soft coding server data structure (I00SC).

- This is a required field
- Use #S01 - #S16 for options
- Use #F01 - #F15 for function exits

Field	Explanation
Program Id	<p>The identification, such as program number, table number, and report number, this is assigned to an element of software.</p> <p style="text-align: center;"><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>
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 style="text-align: center;"><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.

Field	Explanation
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>
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

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.

What You Should Know About

Values in the Parameter fields

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.

To avoid transferring screen or subfile fields values, alternative options for VDxxxx or SFxxxx include:

- 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

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).

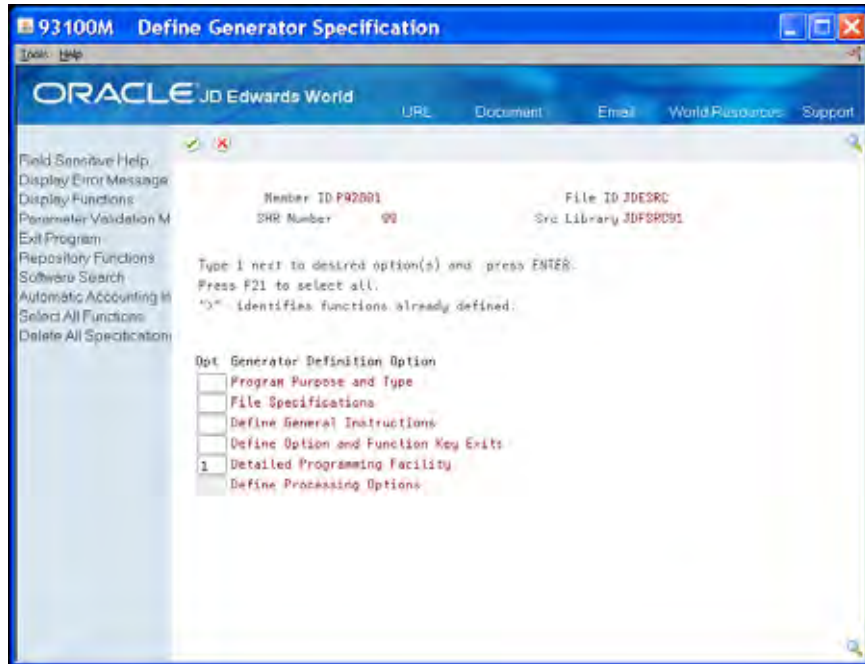
This chapter includes the following:

- [About the Detailed Programming Facility](#)
- [About Full Data Field Parameters](#)
- [Loading VC0 Description Fields](#)
- [Enabling the Database Update Function for Subfiles](#)
- [Creating *ENTRY PLIST Entries](#)
- [Protecting Fields from Being Cleared](#)
- [Disabling Data Dictionary Edits](#)
- [Creating a Partial KLIST for a File](#)

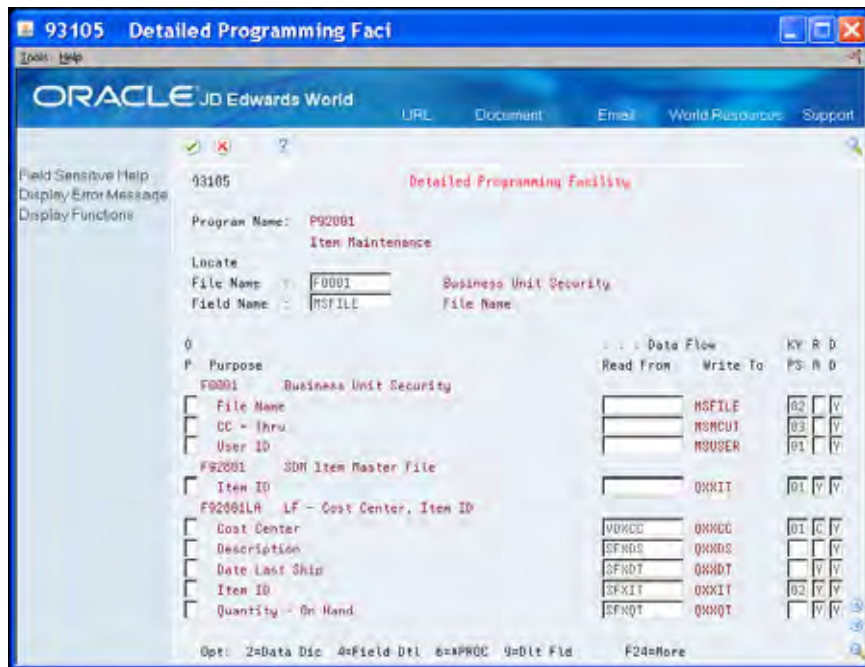
About the Detailed Programming Facility

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.



The Detailed Programming Facility screen displays.



Field	Explanation
File Name	The member ID of the file used by the program.
O P	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.

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.

Value	Description
9 - Delete Record	Allows you to delete a field from the Detailed Programming Facility.

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.

About Full Data Field Parameters

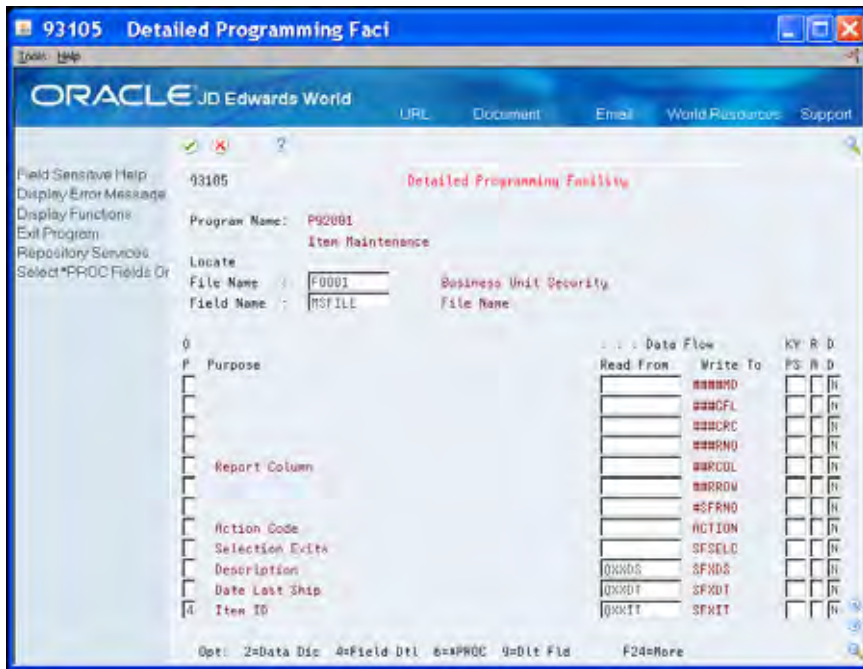
Full Data Field Parameters allows you to create additional source code.

Primary Uses of Full Data Field Parameters

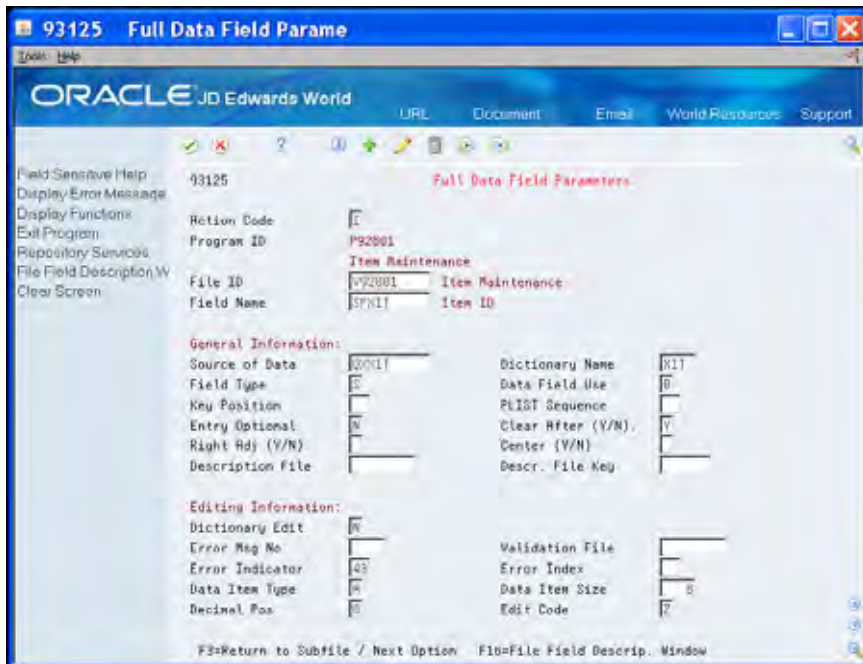
The primary uses of full data field parameters include:

- 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.



The Full Data Field Parameters screen displays.



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.

Field	Explanation
PLIST Sequence	<p>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.</p> <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	<p>Used with subfile maintenance programs to identify the field that controls database updates.</p> <ul style="list-style-type: none"> ▪ One field needs to be designated as Entry Optional: N ▪ Defaults to a blank
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)	<p>A code of Y will center the data within the field when it is displayed.</p>
Description File	<p>Used in conjunction with loading a VCO description field.</p> <ul style="list-style-type: none"> ▪ Identifies the file that contains the description

Field	Explanation
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	<p>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.</p>
Error Indicator	<p>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).</p>
Error Index	<p>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.</p>
Data Item Type	<p>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).</p>

Field	Explanation
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.

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.

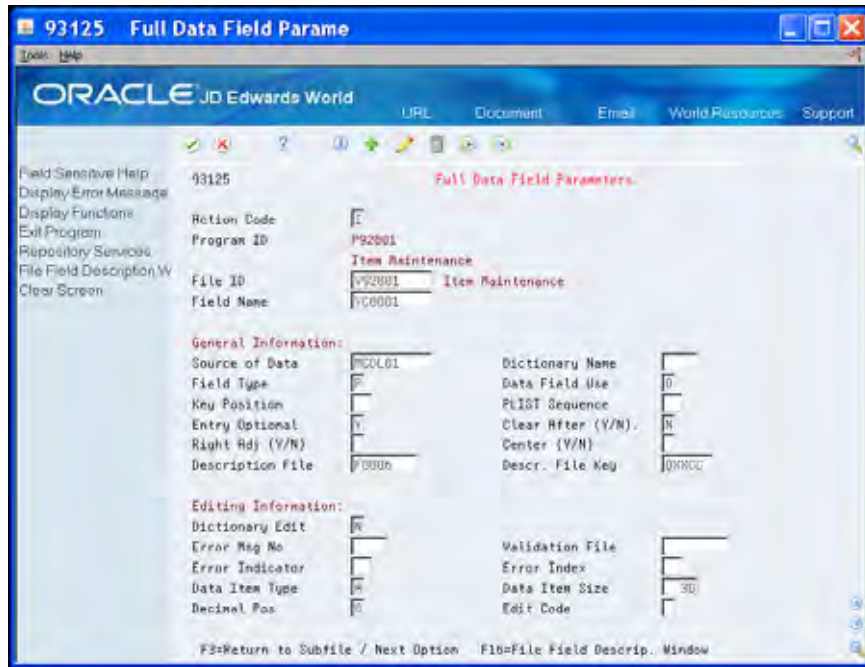
- If the cursor is in the Description File Key field, and this field:
 - Contains a file name, the system enters the fields on the screen with the fields from the description file.
 - Is blank, the system displays a blank File Field screen for you to enter a file name and then displays the fields from 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.

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).

Example

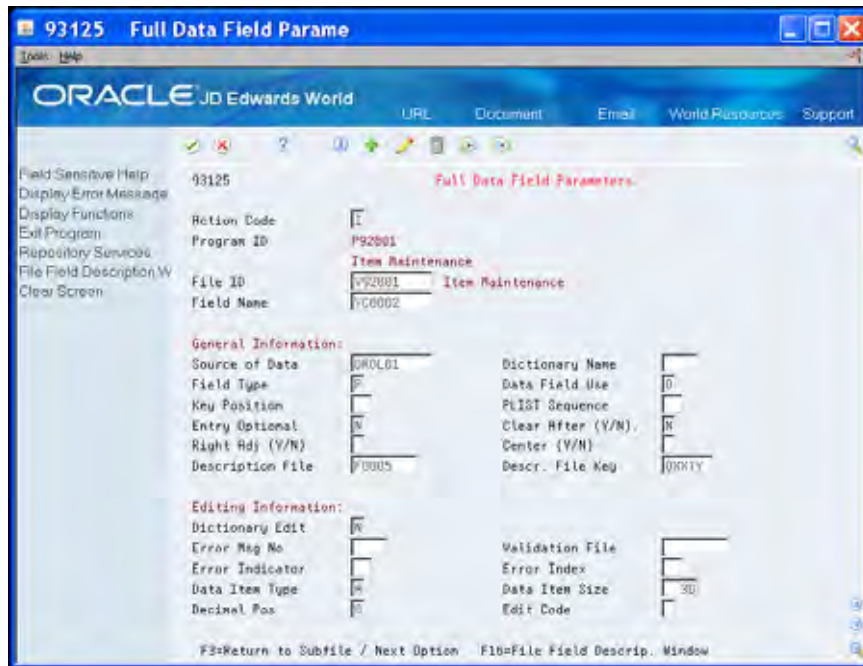
In the following example, QXXCC is a field in the Business Unit Master table (F0006) that contains the business unit value. The F0006 table also contains descriptions of the business units. MCDL01 is the field in the F0006 table that contains the business unit description and the system enters this description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



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.



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.

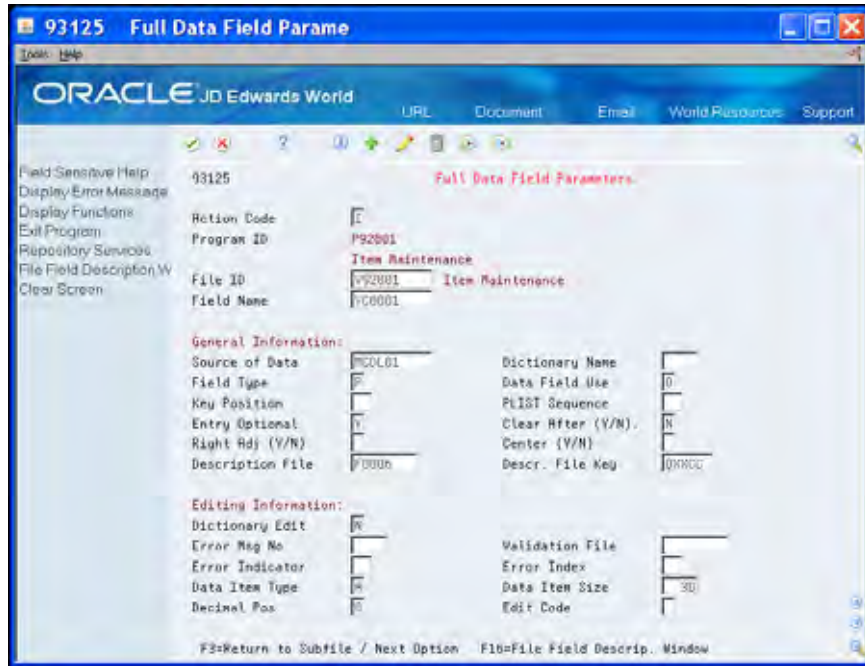
Loading field descriptions using this approach only works if the system enters the field description into a VC0 field.

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



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>

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.

The screenshot shows the 'Full Data Field Parameters' screen in Oracle JD Edwards World. The window title is '93125 Full Data Field Parame'. The screen displays various parameters for a data field named 'SFXIT'. Key parameters include:

- Action Code: []
- Program ID: P92801
- File ID: P92801
- Field Name: SFXIT
- Dictionary Name: RIT
- Data Field Use: R
- Entry Optional: N
- Edit Code: R

Other parameters shown include Source of Data (R001), Field Type (R), Key Position (), Entry Optional (N), Right Adj (Y/N) (), Description File (), Dictionary Name (RIT), Data Field Use (R), PLIST Sequence (), Clear After (Y/N) (Y), Center (Y/N) (), Descr. File Key (), Dictionary Edit (N), Error Msg No (), Validation File (), Error Indicator (43), Error Index (), Data Item Type (R), Data Item Size (), Decimal Pos (0), and Edit Code (R).

At the bottom, there are instructions: F3=Return to Subfile / Next Option, F10=File Field Descrip. Window.

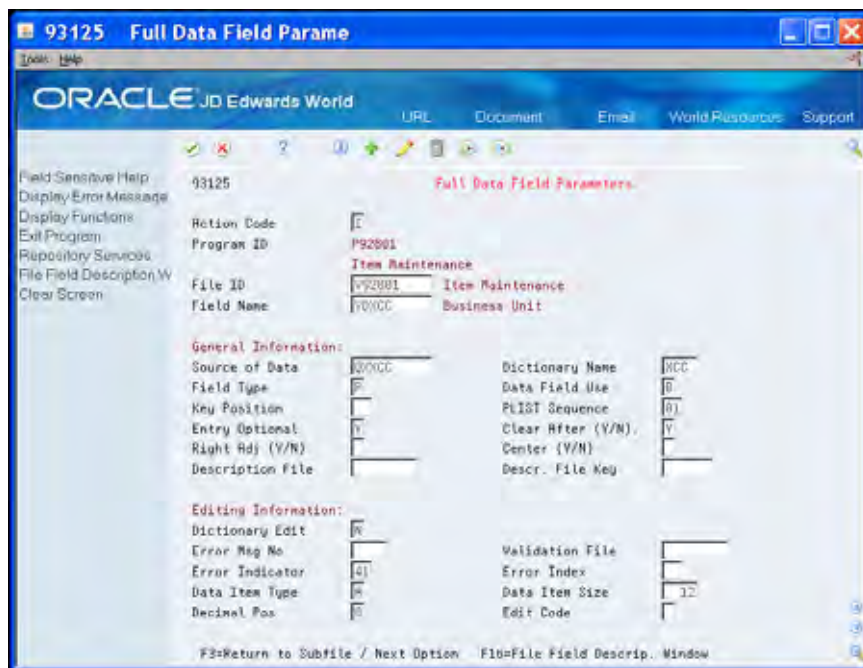
Creating *ENTRY PLIST Entries

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

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.



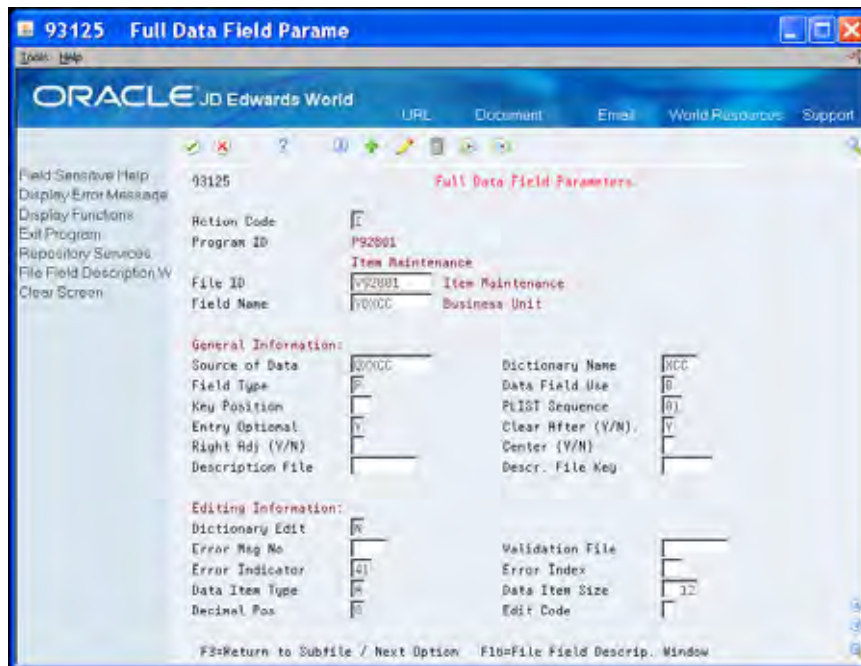
If the parameter value is not blank, the system updates the variable \$AUTO with a 1. When the system launches this program, this parameter informs the program to perform an automatic inquiry (S003).



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.



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.

The screenshot shows the 'Full Data Field Parameters' screen in Oracle JD Edwards World. The title bar indicates '93125 Full Data Field Parame'. The screen is divided into several sections:

- Field Sensitive Help:** 93125
- Action Code:** []
- Program ID:** P92001
- File ID:** V92001 Item Maintenance
- Field Name:** FC0001
- General Information:**
 - Source of Data: MC0L01
 - Field Type: F
 - Key Position: []
 - Entry Optional: Y
 - Right Adj (Y/N): []
 - Description File: P000h
 - Dictionary Name: []
 - Data Field Use: D
 - PLIST Sequence: []
 - Clear After (Y/N): N
 - Center (Y/N): []
 - Descr. File Key: D0K000
- Editing Information:**
 - Dictionary Edit: N
 - Error Msg No: []
 - Error Indicator: []
 - Data Item Type: N
 - Decimal Pos: []
 - Validation File: []
 - Error Index: []
 - Data Item Size: 30
 - Edit Code: []

At the bottom, there are instructions: F3=Return to Subfile / Next Option and F10=File Field Descrip. Window.

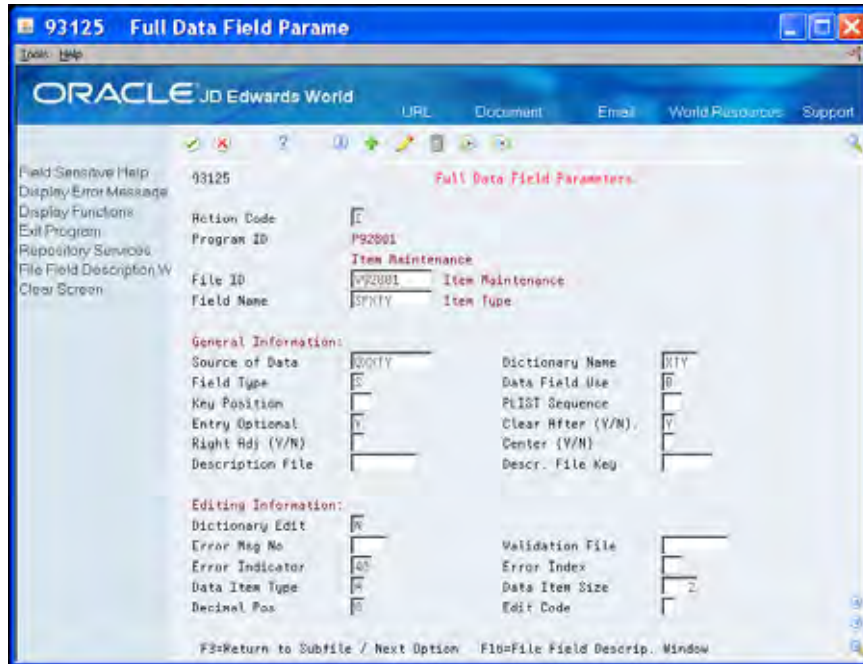
What You Should Know About

User Error Messages

Updating the Error Message Number and Error Index fields adds errors to the EMK array in Subroutine S999.

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.

You must modify your source to use this error message index.

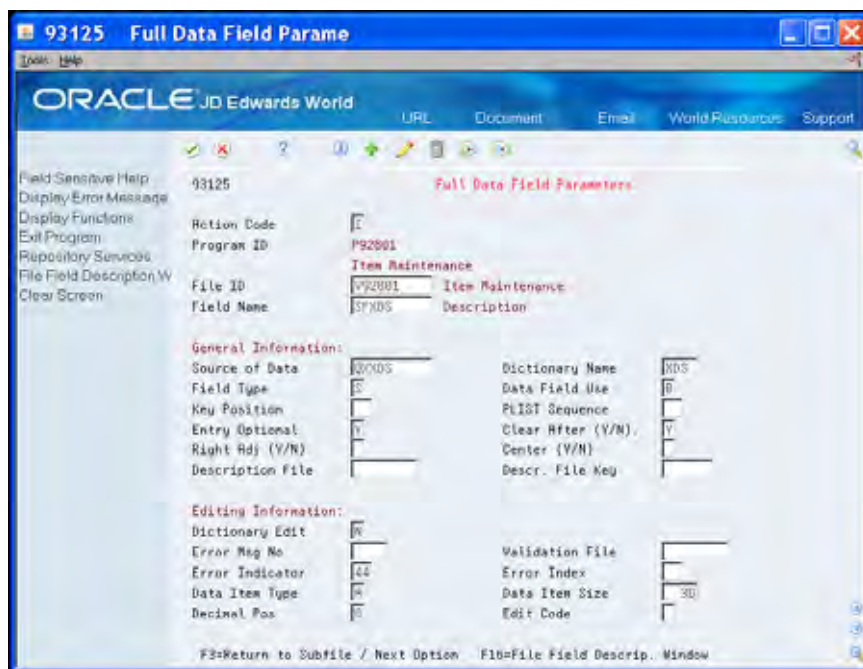


Disabling Data Dictionary Edits

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.



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.

Example

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

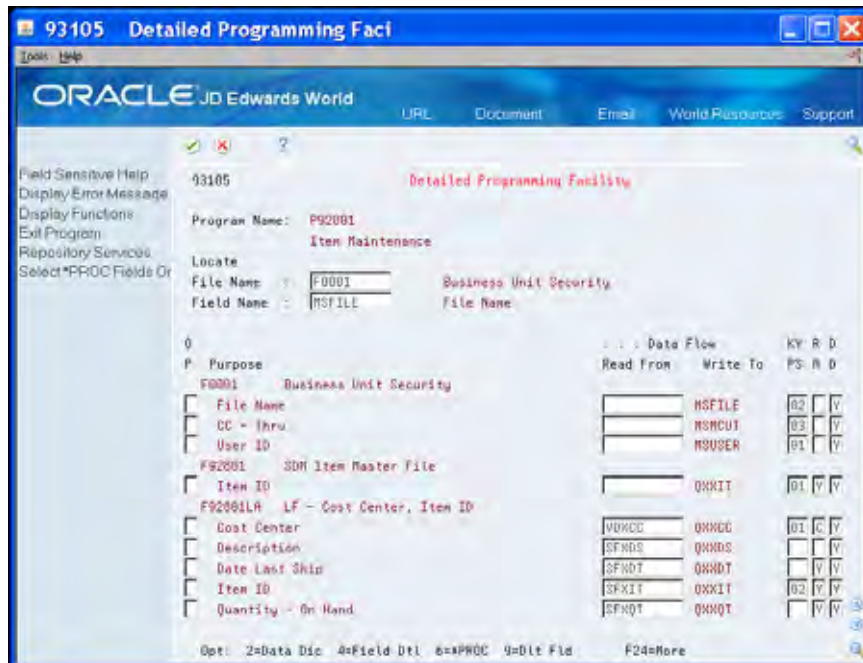
- User ID
- File Name
- Through Business Unit

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.



Define Processing Options

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.

What You Should Know About

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.

Example – Interactive Programs Using Processing Options

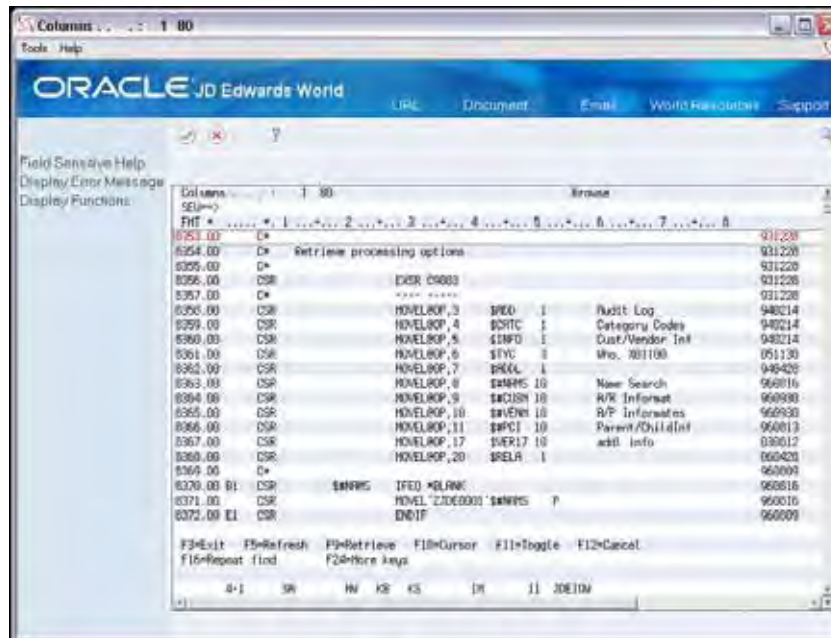
Following is an example of an interactive program containing processing options.

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



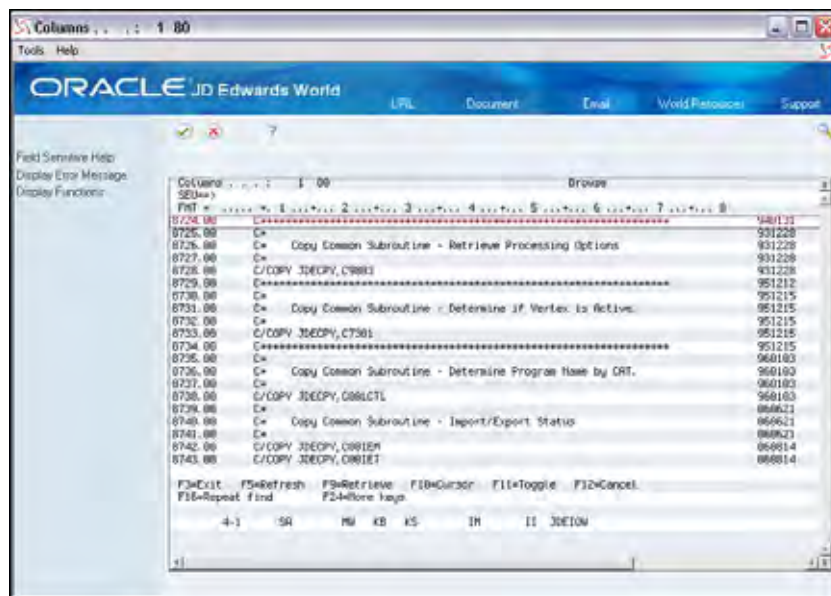
This code copies the E Specs that relate to the common subroutine. This code defines all necessary arrays and tables for the copy module C9803. Typically, you add copy statements such as this at the end of the E Specs.

The next section of code in this program which relates to processing options follows:



The system loads the @OP array for the processing options. @OP,1 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 which relates to processing options follows:



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.

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.

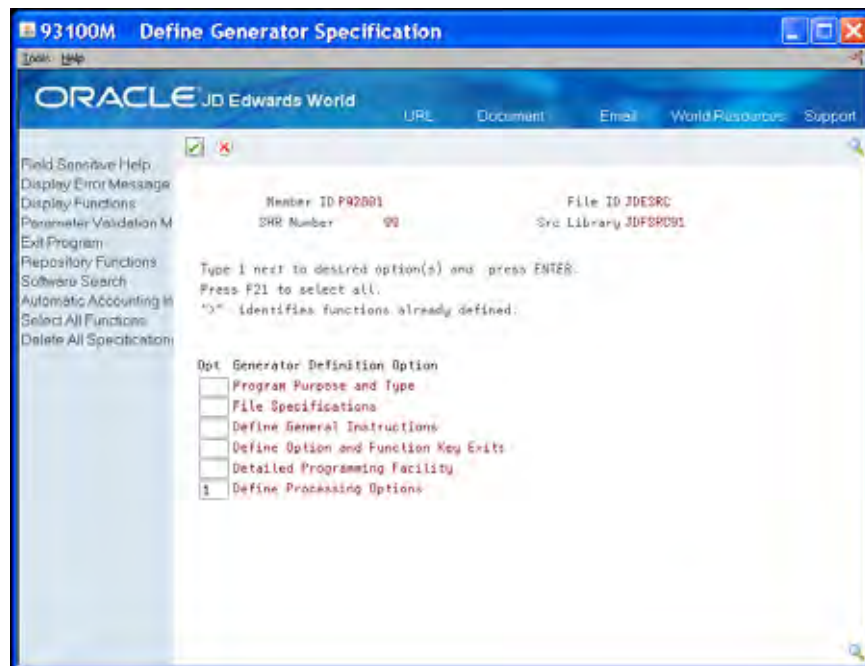
Defining Processing Options

This section includes the following tasks:

- [To enter processing options](#)
- [To view code for the processing options](#)

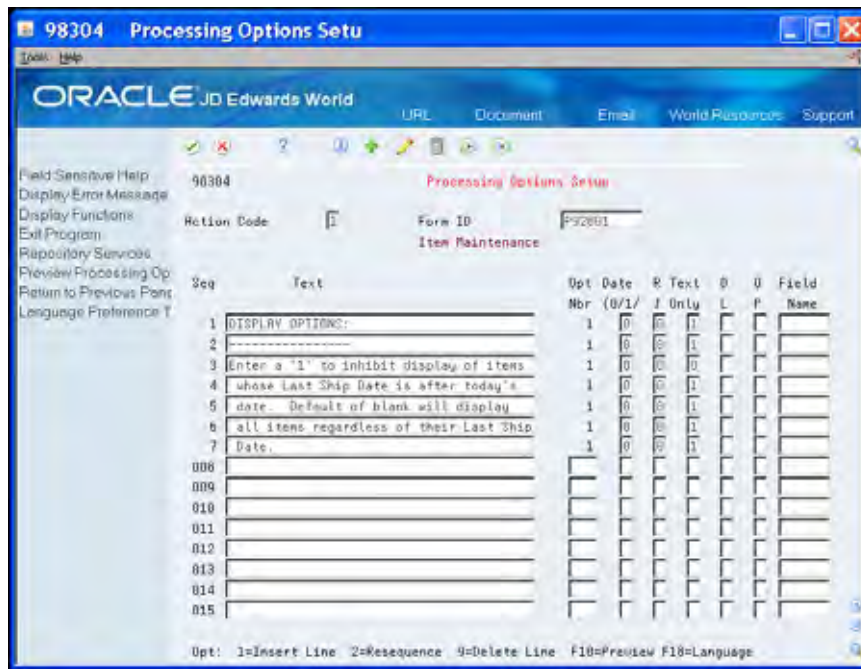
To enter processing options

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



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



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.

Field	Explanation
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 but the processing option number should never be changed. This field is not input capable for existing lines of text.</p> <p style="text-align: center;"><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>
R J(Right Justify)	<p>Determines if the entry field is right-justified. Valid values are:</p> <ul style="list-style-type: none">0 Information is not right-justified1 Information to be entered is numeric and should be right-justified2 Information to be entered is to be right-justified and left-filled with blanks
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> <ul style="list-style-type: none">1 for text only0 for a value entry line. <p>Each separate processing option can have only one input value, or "0" value.</p>

Field	Explanation
D L (Display Level)	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.
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 (I005C).</p> <ul style="list-style-type: none"> ▪ This is a required field ▪ Use #S01 - #S15 for options ▪ Use #F01 - #F15 for function keys <p style="text-align: center;"><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 style="text-align: center;"><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.

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.

4 Program Design Language

Overview to Program Design Language

Objectives

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

About 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

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.

This chapter includes the following:

- [About PDL Statements](#)
- [About Blocks of Statements](#)
- [About Comments](#)
- [About Assignments](#)
- [About Database Operations](#)
- [About Program Calls](#)
- [About Loops](#)
- [About Conditions](#)
- [About Miscellaneous Keywords and Syntax](#)

About PDL Statements

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

- Keywords
- Variables
- Database Files
- Operators
- Constants

- Punctuation

Keywords

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

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.

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.

Operators

You define the valid assignment and arithmetic operators.

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.

Punctuation

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

About Blocks of Statements

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

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 vdtrdj = '' Then
    vdtrdj := $$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
End

```

About Comments

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.

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

```

About Assignments

Operator and Syntax in as

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

Rules

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 (qxxcc,3,10)

```

About Database Operations

Keywords and Syntax

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

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
```

About Program Calls

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;

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
    
```

About Loops

Keywords and Syntax

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

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;
        End
    End
End
    
```

```

    $#am1 := $#am1 + $#xtp;
    reade f59422;
  End
End

```

About Conditions

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)

Symbols

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

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

```

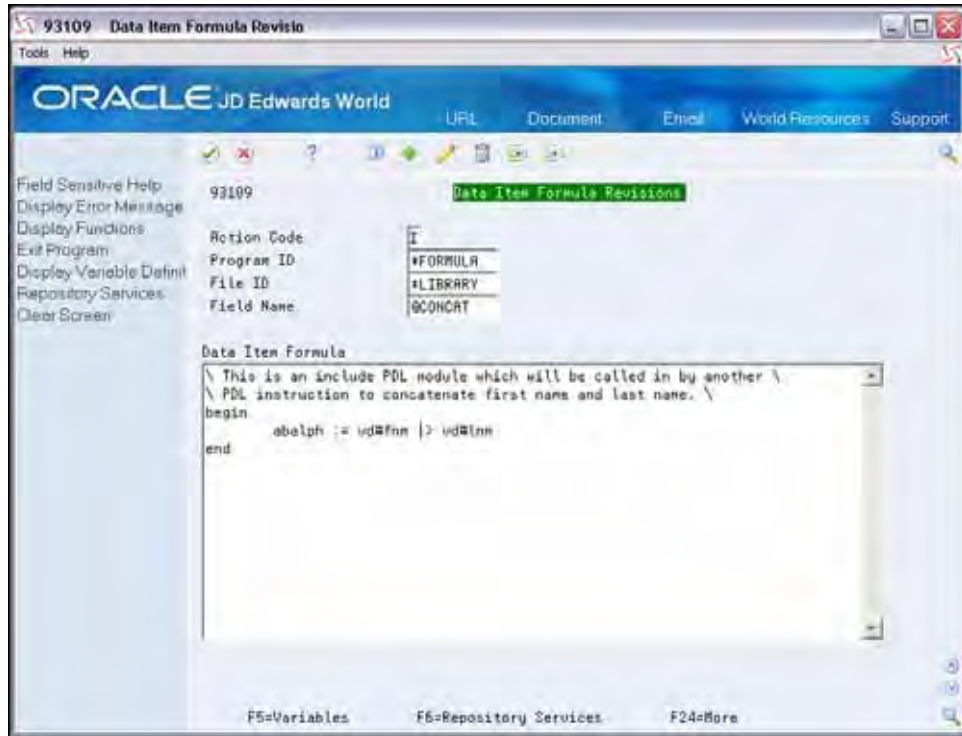
About Miscellaneous Keywords and Syntax

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;

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.

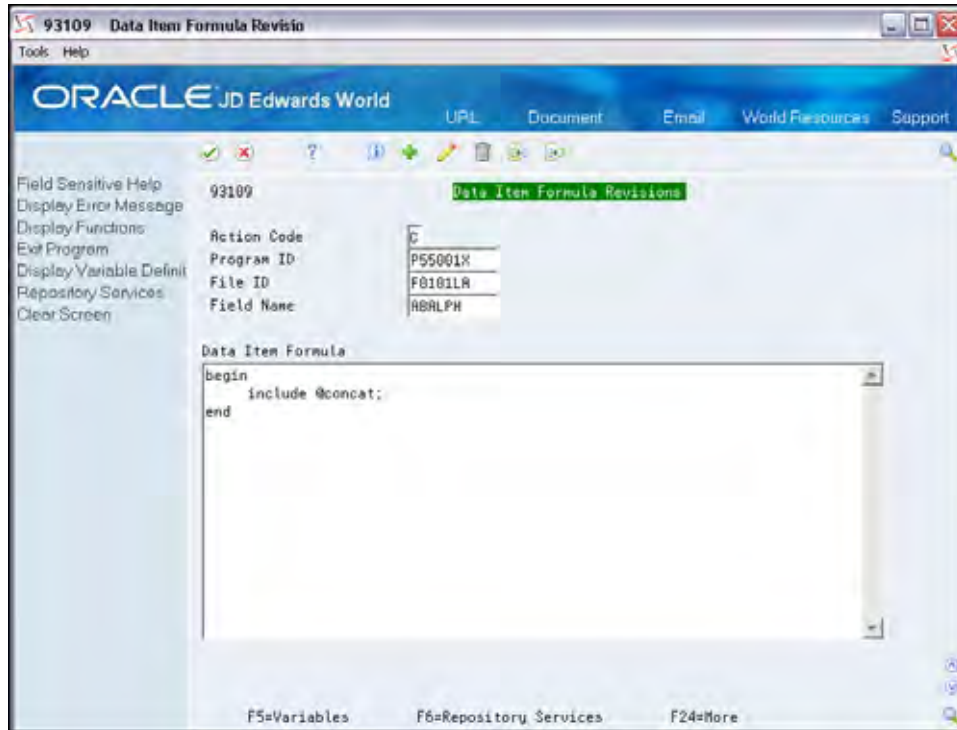


The include module should have a unique name, for example @CONCAT.

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.



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 includes additional information about Program Design Language (PDL) operations:

- [Editing](#)
- [Parsing](#)
- [Source Code Generation](#)
- [Data Item Formula Examples](#)

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).

You can enter a maximum number of 200 statements.

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.

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.

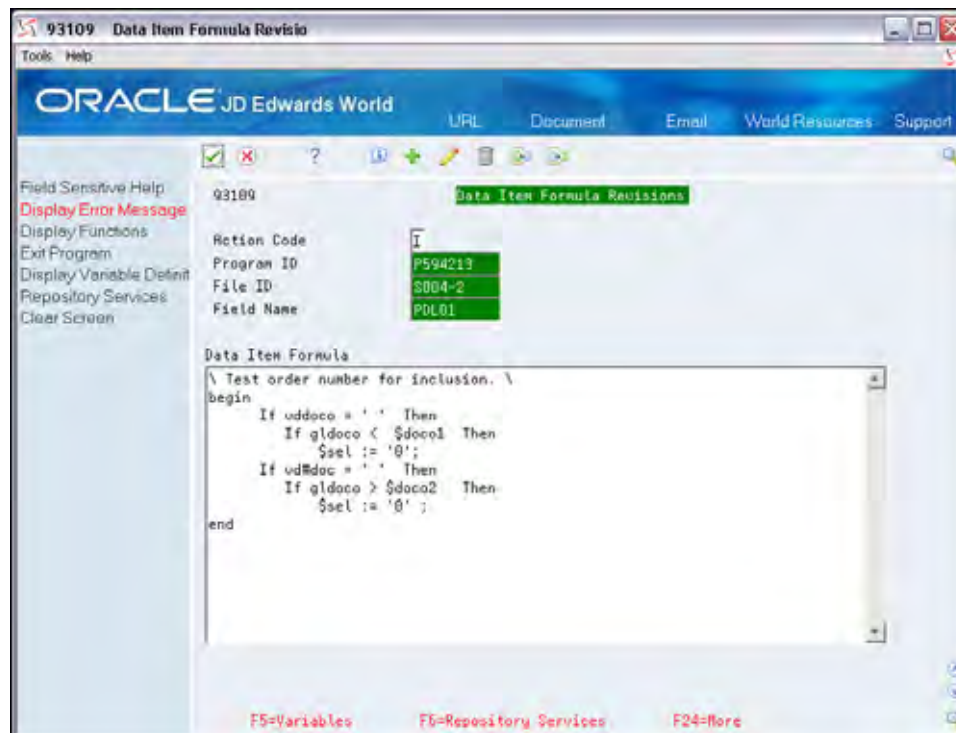
The generator lists the fields in the Detailed Programming Facility in alphabetical order, and you cannot change the order.

Data Item Formula Examples

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

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.



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';

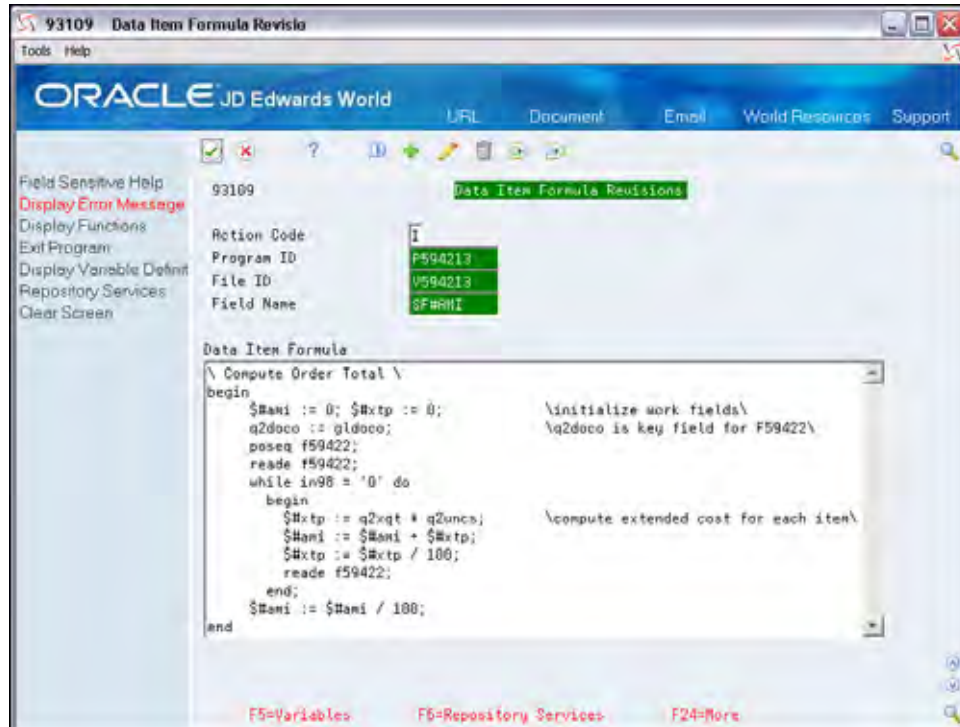
Additionally, this example illustrates the nesting of conditions.

```
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.

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.



This example illustrates the following types of PDL statements:

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

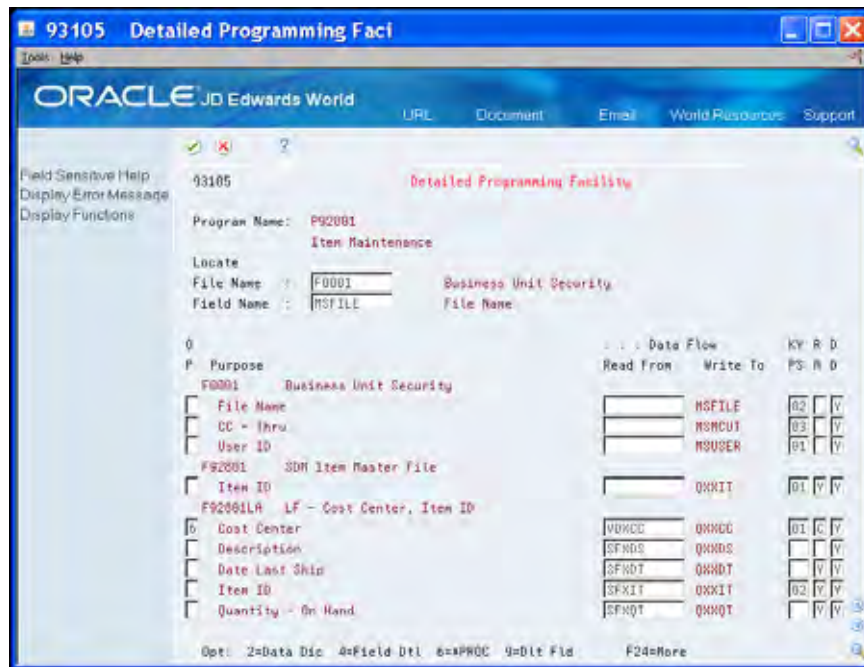
You can separate statement separators in the loop statements within the Begin...End block, and then follow the end statement with a separator.

Work with Data Item Formula Revisions

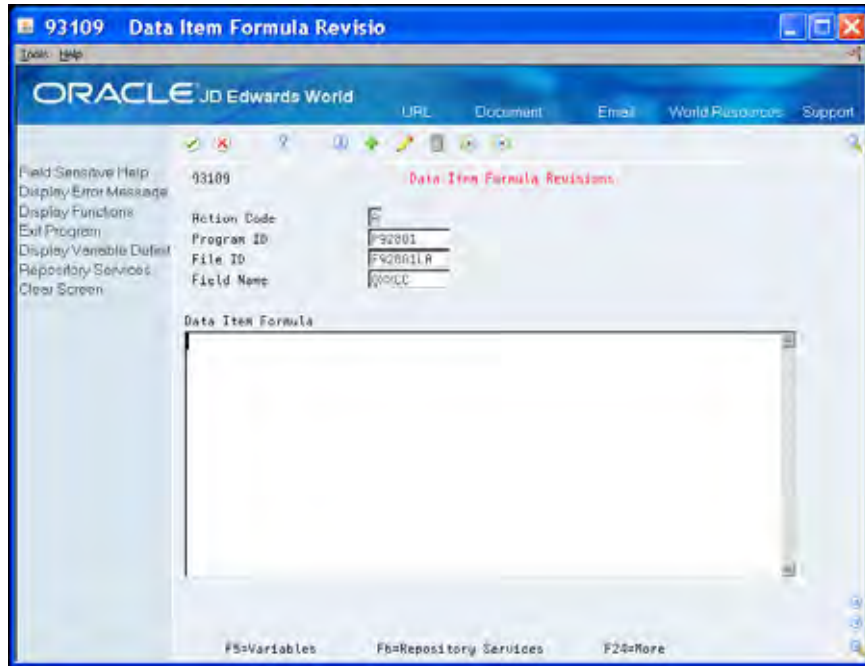
Use the Data Item Formula Revisions screen to add Program Design Language (PDL) to a field.

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.



The Data Item Formula Revisions screen displays.



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

Field	Explanation
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.

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.

5 Source Modifications

Overview to Source Modifications

Objectives

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

About Source Modifications

After you generate source code, you can make modifications to it by 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

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.

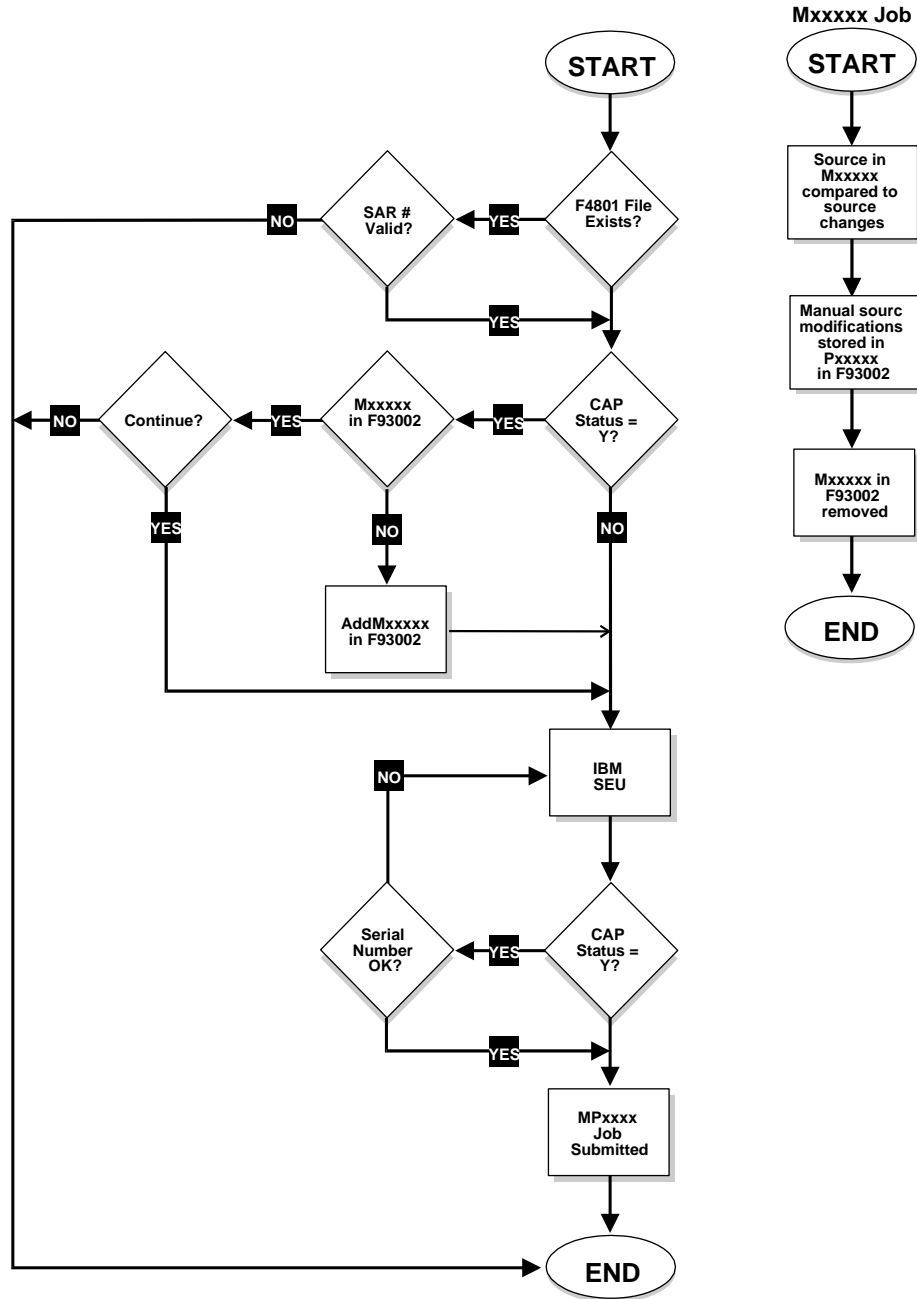
Pre-SEU and Post-SEU Process

A front-end JD Edwards World program, MPxxxx, monitors the changes on the SEU. The MPxxxx 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 Pxxxx 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.



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

When regenerating source code you should know:

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

This chapter includes the following:

- [When to Regenerate Source Code](#)
- [Changing CAP Status](#)
- [Resolving CAP Status Invalid Error](#)

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.

Changing CAP Status

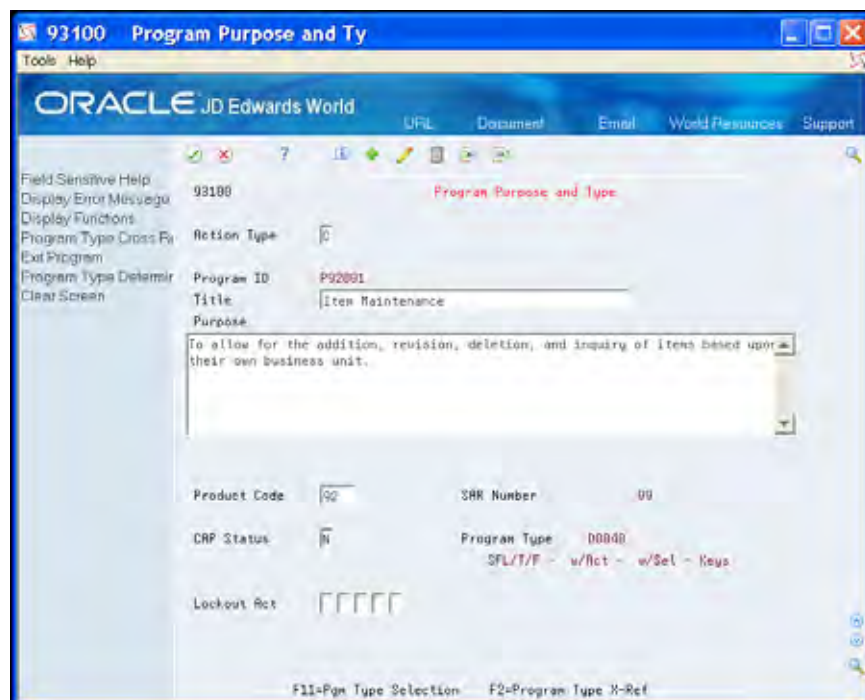
If you change the CAP Status field, the system deletes the changes in the Source Entry Utility that it stores in the P member of the Additional Help/Modifications Master file (F93002). JD Edwards World recommends that you do not change the CAP Status field unless the changes you make to your program 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



The Delete KBG Modifications screen displays.



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

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 Resolve	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>
Ensure the Pxxxxx member exists in the Additional Help/Modifications Master file (F93002).	<ul style="list-style-type: none">▪ 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.	<ul style="list-style-type: none">▪ 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.	

Work with Model Control Language Programs

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

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.

This chapter includes the following tasks:

- [To copy a model CL](#)
- [To customize a CL model](#)

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.

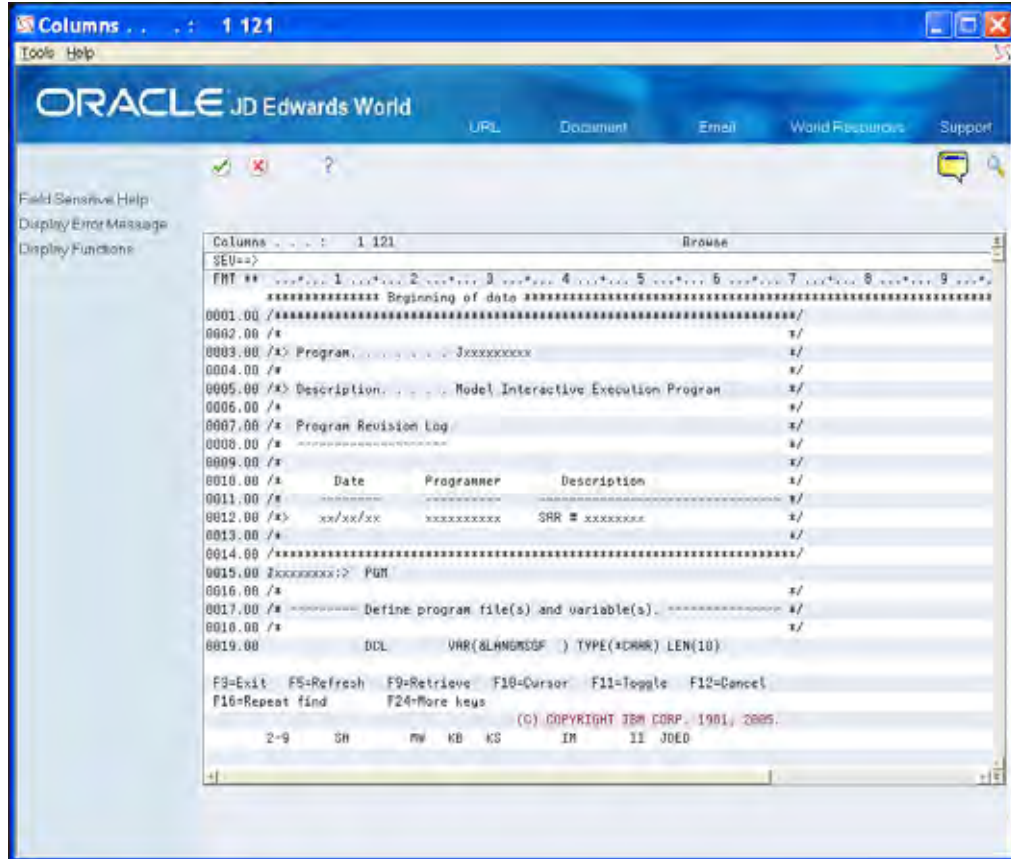
To copy a model CL

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.



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.

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.

Model CL Programs	Description
J98MODEL3	Serves as a template for interactive programs that need a prompt for parameters.
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.
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 *Work with Quick Start CL Generator* for more information.

6 CASE Programs

Overview to CASE Programs

Objectives

- To create CASE programs

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

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.

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.

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.

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.

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.

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.

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.

Quick Start Generation

You can generate this program type using Quick Start.

Overview to Subfile Maintenance Programs

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.

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.

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.

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.

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.

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.

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.

Quick Start Generation

You can generate this program type using Quick Start.

Create Report Programs

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 *Technical Tools Guide* for more information.

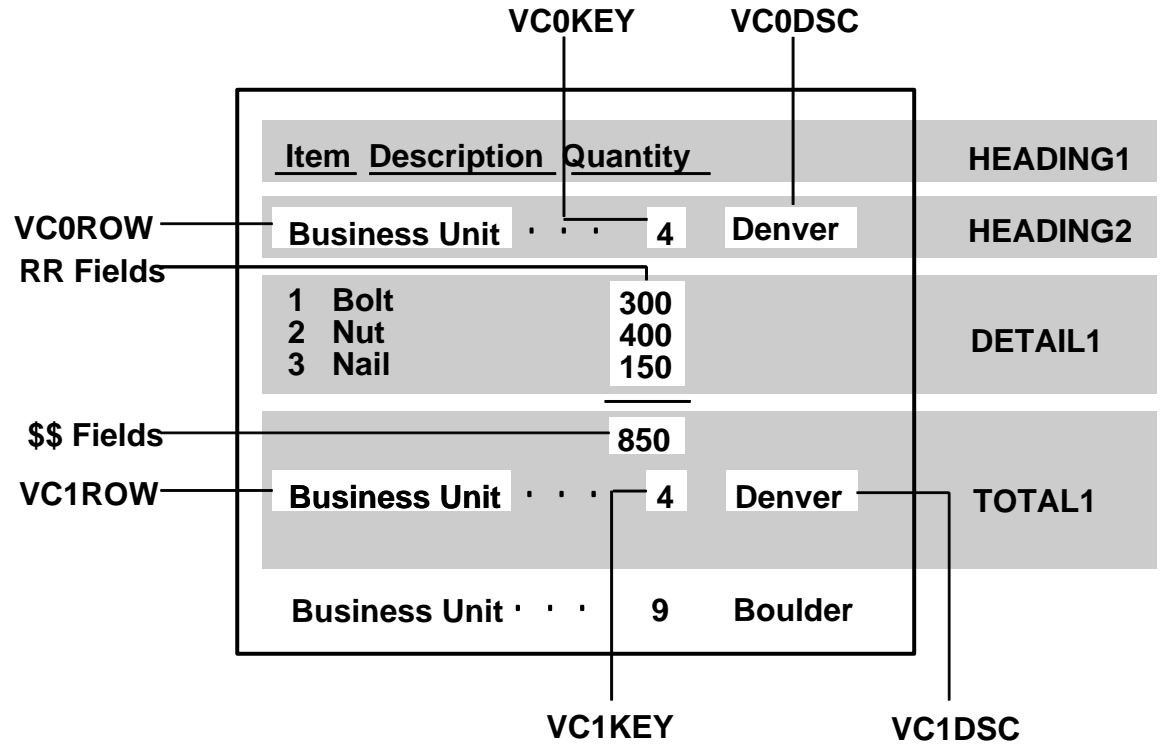
Note: The Report Design Aid is currently not available in the Java platform for JD Edwards World software. You must create reports using the green screen platform of the software.

This chapter includes the following:

- [Understanding RDA Special Use Fields](#)
- [Creating a Total Format](#)
- [Defining a Subheading](#)
- [Understanding DREAM Writer Considerations](#)

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.



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.

Field	Explanation
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.

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).

```

92520                      Record Formats List
Report: R929400

  Opt  Format Name  Type  Past Path  Start / End  Related  # Fields  Fld
      Format Name  Type  Path      File      Lines   Record   Selected Pfx
-----
  1  HEADING1     REPORT  P929401   001 008         000      EE
  1  DETAIL1      REPORT  P929401   009 009         000      EE
  1  TOTAL1       REPORT  P929401   010 011         000      EE
-----
  Opt: 1=DB Field Selection 3=Field List 4=Delete 5=Format Keywords
    
```

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.

```

928400                      000000000000000000000000000000000000000000
                                Inventory by Cost Center
                                000000000000000000000000000000000000000000
                                000000000000000000000000000000000000000000

  Business      Cost Center      It      Item Type
  Unit          Description      T I      Description      u
-----
000000000000  00000000000000000000000000000000  00  00000000000000000000000000000000  0
*
                                000000000000000000000000000000000000000000

Report: R928400          Field Definition          Format: TOTAL1
Dict Name      Text
Data Type     Field Name    VC1ROW      Cond Ind
Row/Column    11 55        Field Use   @           Highlight =
Size         30          Text Form   -           Underline -
               Lines Cond Ind
Space Before
Space After
Skip Before
Skip After
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 VCIKEY in the Field Name field.

```

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

```

Item Type	Item	Item	Ship	Quantity
Description	umber	r	Date	On Hand
00000000000000000000000000000000	00000000	00000000000000000000000000000000	00000000	00000000000000
00000000000000000000000000000000*	000	00000000000000000000000000000000		00000000000000

Report: R928400 Field Definition Format: TOTAL1

Dict Name	Text	Field Name	VCIKEY	Cond Ind
Data Type		Field Use	0	
Row/Column	11 52			
Size	12	Text Form		
	Lines	Cond Ind		
Space Before		Highlight		
Space After		Underline		
Skip Before		Field Cond		
Skip After		Char per Inch		
		Edit Code		
		Asterisk Fill		
		Float Symbol		

F3=Exit F12=Prev Screen F17=Dictionary

11. Click Enter.

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 of 5.

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

The Field Definition screen displays.

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

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

Item Type	Item	Item	Ship	Quantity
Description	Number	Description	Date	On Hand U-

00000000000000000000000000000000	00000000	000000000000000000000000	00000000	00000000000000
00000000000000000000000000000000	000	00000000000000000000000000000000		00000000000000*

Report: R928400		Field Definition		Format: TOTAL1	
Dict Name	Text	Field Name	VC1DSC	Cond Ind	
Data Type	A	Field Use			
Row/Column	11 66		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

14. Click Enter.

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.

15. On Report Design Aid, add the field to be accumulated to the report.

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.

928400	J.D. Edwards & Company						Page No. . . . 2
Inventory by Business Unit Report						Date - . 12/02/17	
Bus Unit	Description	Ty	Description	Number	Item Description	Date Ship On 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	

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: screen

92520 Report: R928400		Record Formats List					
Opt	Format Name	Type	Data Base File	Start / End Lines	Related Record	No. Flds Selected	Fld Pfx
-	HEADING1	REPORT		001 008		000	RR
-	DETAIL1	REPORT		009 009		000	RR
-	TOTAL1	REPORT		010 012		000	\$\$
-	HEADING2	REPORT		013 013			RR
=							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
-							
Opt: 1=DB Field Selection 3=Field List 4=Delete 5=Format Keywords							

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.

928400		J.D. Edwards & Company				Page No. . . . 2	
		Inventory by Business Unit Report				Date - . . . 12/02/17	
Bus Unit	Description	It Ty	Description	Item Number	Description	Ship Date	Quantity On Hand UM
5	San Francisco Branch	N	Non-Refrigerated	2524 1	Inch Nail	06/01/17	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
							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.

928400		J.D. Edwards & Company				Page No. . . . 2	
		Inventory by Business Unit Report				Date - . . . 12/02/17	
Business Unit	5 San Francisco Branch	It Ty	Description	Item Number	Description	Ship Date	Quantity On Hand UM
5	San Francisco Branch	N	Non-Refrigerated	2524 1	Inch Nail	06/01/17	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
							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 *Technical Foundation Guide*.

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 includes on the cover page are in the following example:

Program ID . . . P928400	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 VCOCO 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 VCOCO field contains the name of the company. The DREAM Writer fields have an RR prefix. For example, RRTTL@ field contains line 1 of the DREAM Writer version ID description. The RRTXT2 and RRTXT3 are lines 2 and 3 of the DREAM Writer version description, respectively.

928400		J.D. Edwards & Company				Page No. . . . 2	
		Inventory by Business Unit Report				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.

7 Additional Tools

Overview to Additional Tools

Objectives

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

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

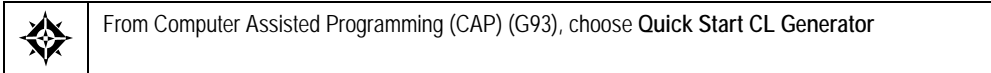
The Quick Start CL Generator provides a quick and easy way for you to create a Control Language (CL) program for any of the following four types of programs.

- 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.

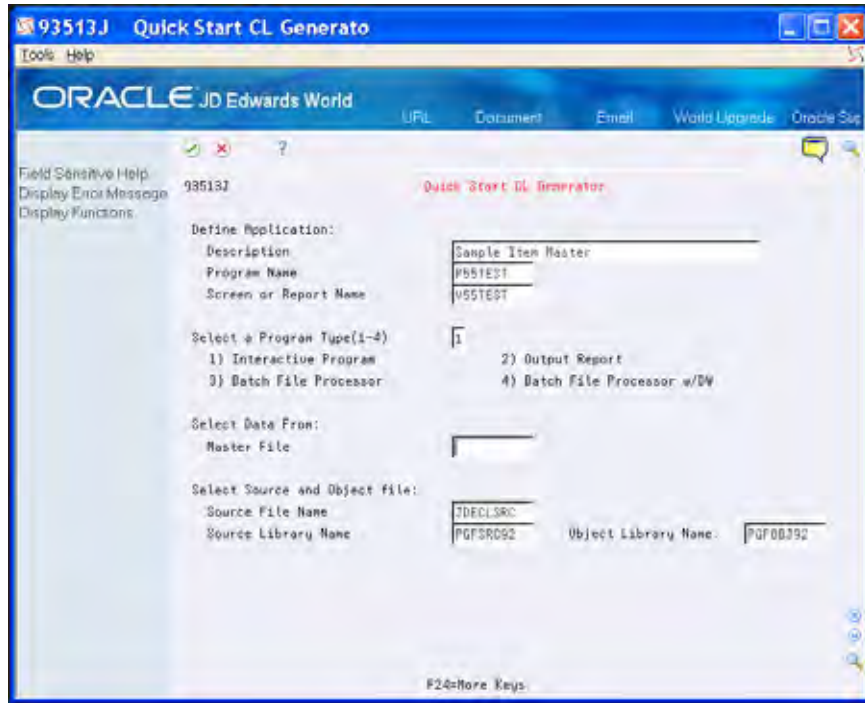
Perform the following tasks:

- [To create a program using the Quick Start CL Generator](#)
- [To compile a CL program](#)



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.

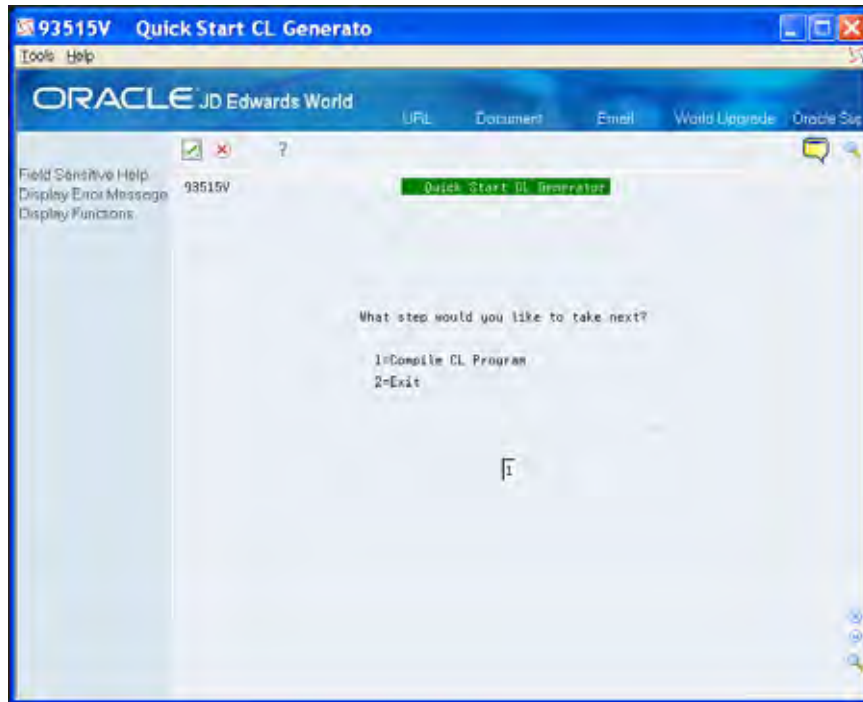


Field	Explanation
Description	Use this field to enter a short one-line description of the program you are creating.
Program Name	Type the name of the RPG program that the CL program will call. This is a required field. Do not leave it blank. The name of the CL program generated will be the same as the 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.

Field	Explanation
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.



Work with the Quick Start Application Tool

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.

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.



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

This chapter includes the following:

- [Defining the Application](#)
- [Selecting Data Fields](#)
- [Browsing or Updating the Screens or Reports \(Optional\)](#)
- [Compiling the Screens or Report \(Optional\)](#)
- [Modifying Specifications \(Optional\)](#)
- [Submitting the Program to Compile \(Optional\)](#)
- [Updating the Data Dictionary and Glossary](#)

Defining the Application

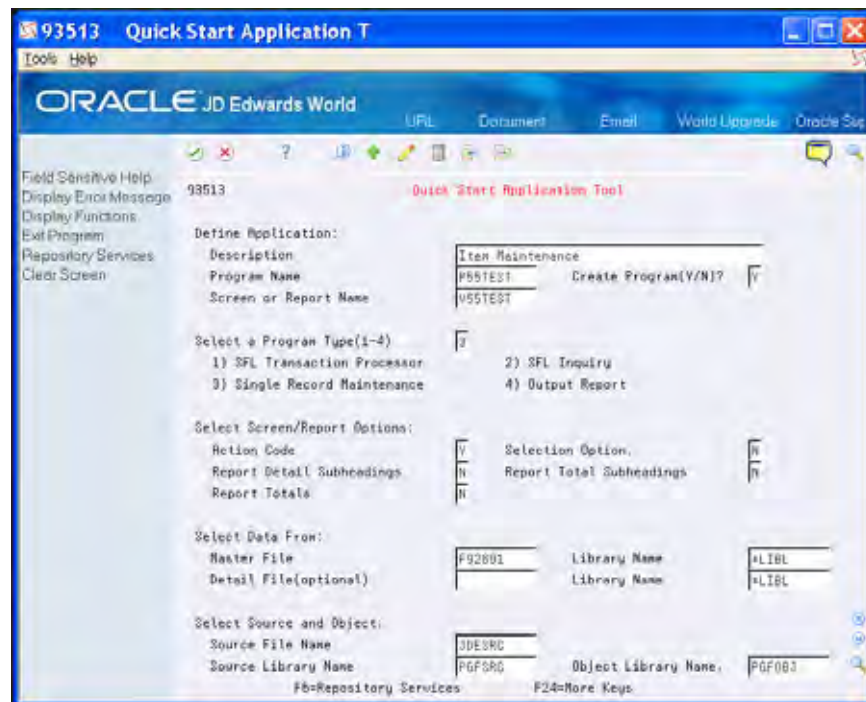
You create versions of programs, screens, and reports by defining the initial criteria.

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



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.
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: <ul style="list-style-type: none"> 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.

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

1. 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.



Browsing or Updating the Screens or Reports (Optional)

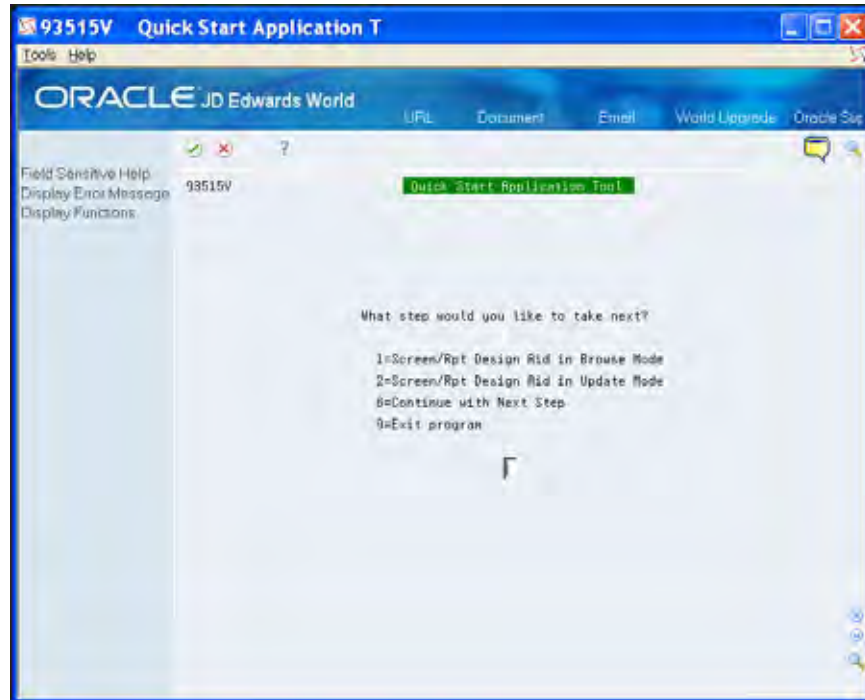
At this point in the process, you can view or update the screens or reports you are create using the browse or update mode.

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.



Compiling the Screens or Report (Optional)

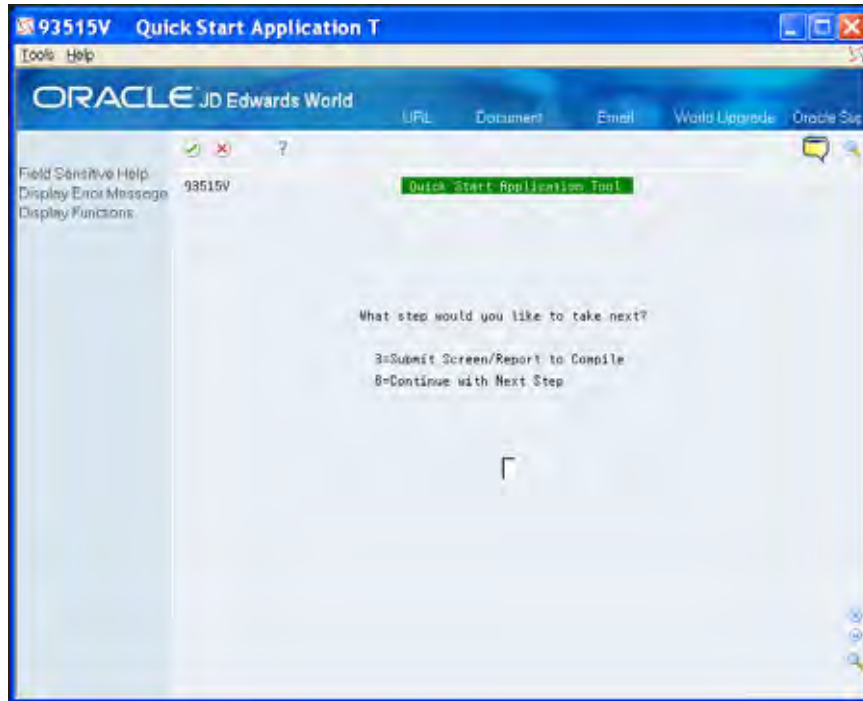
At this point in the process, you can compile the screens or report. Before you compile the program, you must complete this step.

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.



Modifying Specifications (Optional)

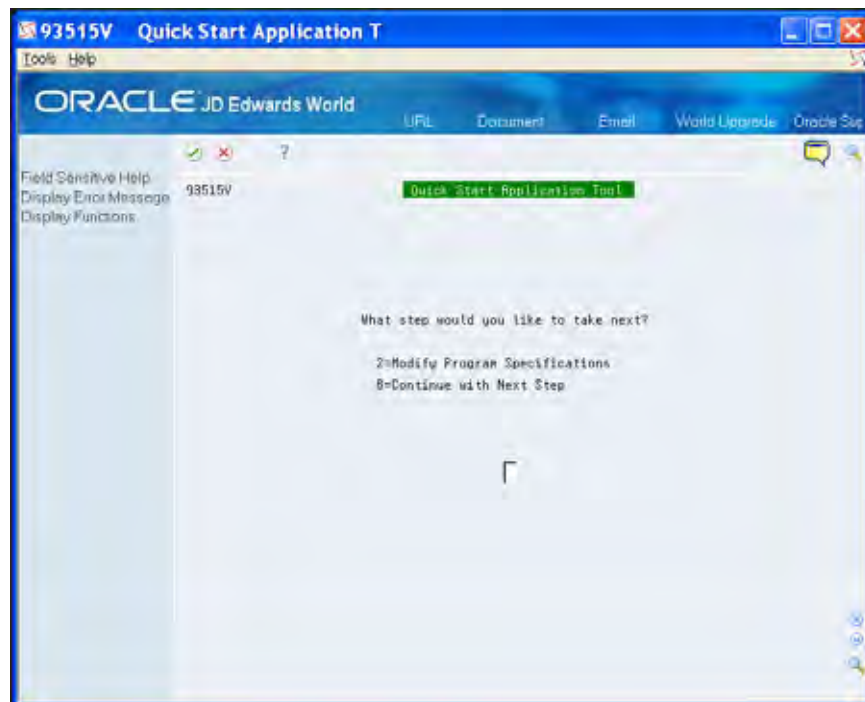
At this point in the process, you can access the Program Generator Specifications screen to modify specifications.

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.



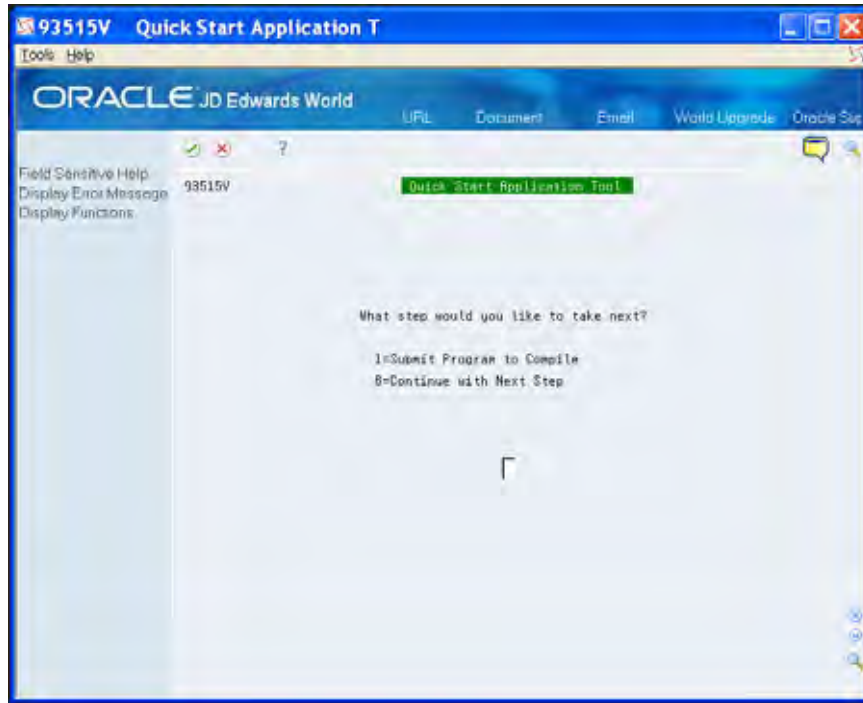
Submitting the Program to Compile (Optional)

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.

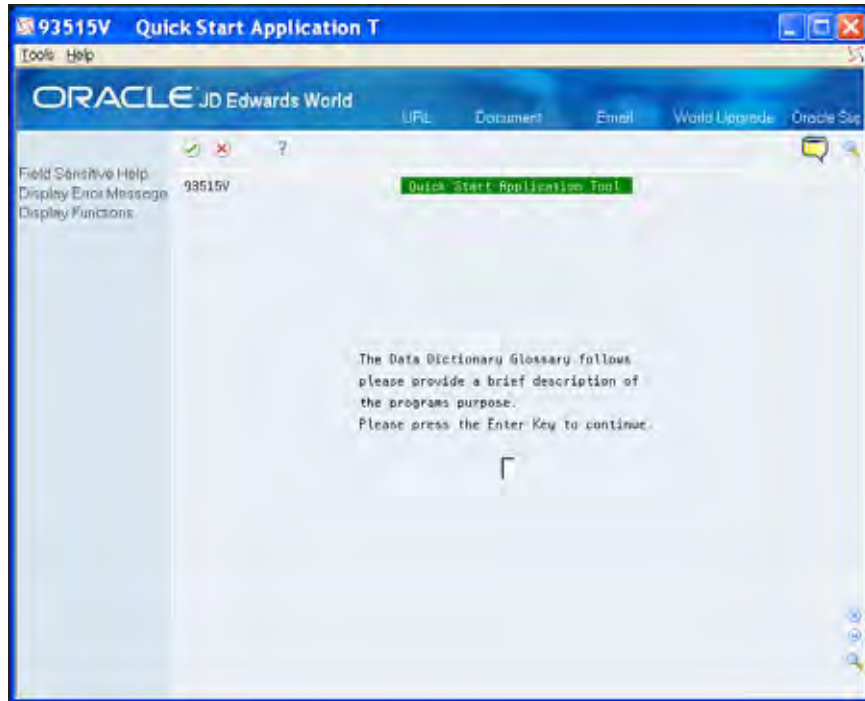


Updating the Data Dictionary and Glossary

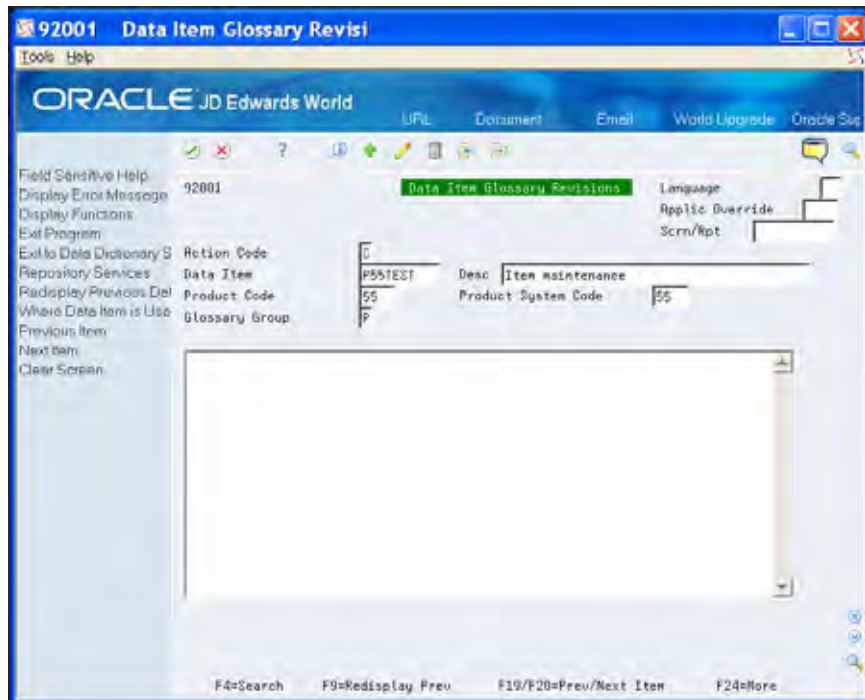
You must enter the purpose of the program on the Data Item Glossary Revisions screen.

To update the data dictionary and glossary

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

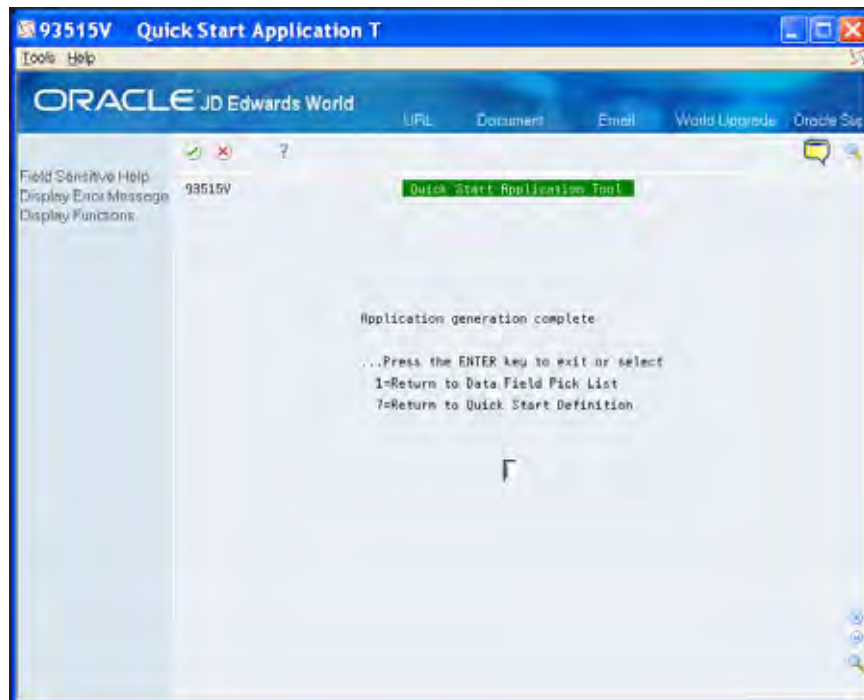


2. On Data Item Glossary Revisions, enter the description of the program's purpose that displays in the online help instructions.



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



Work with Action Diagramming

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.

This chapter describes the following:

- [Building an Action Diagram](#)
- [Viewing an Action Diagram](#)
- [Accessing the Logic Translation Feature](#)

Building an Action Diagram

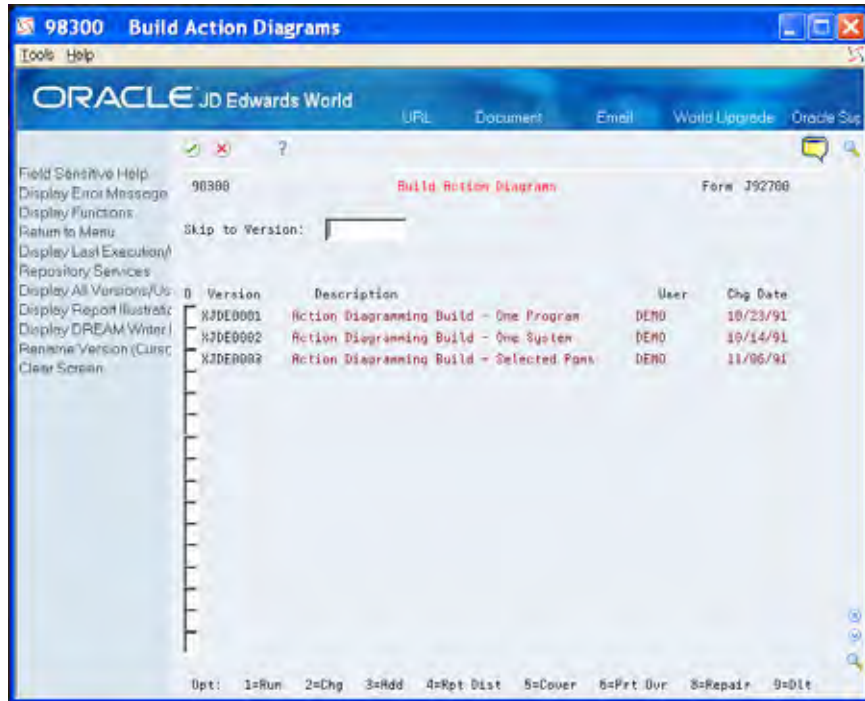
The Build Action Diagram program allows you to build the necessary cross reference items to produce the action diagram. Using DREAM Writer as the initial screen to the batch job, you can specify the programs for which you want to build an action diagram.

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



On Build Action Diagram, choose a version.



Viewing an Action Diagram

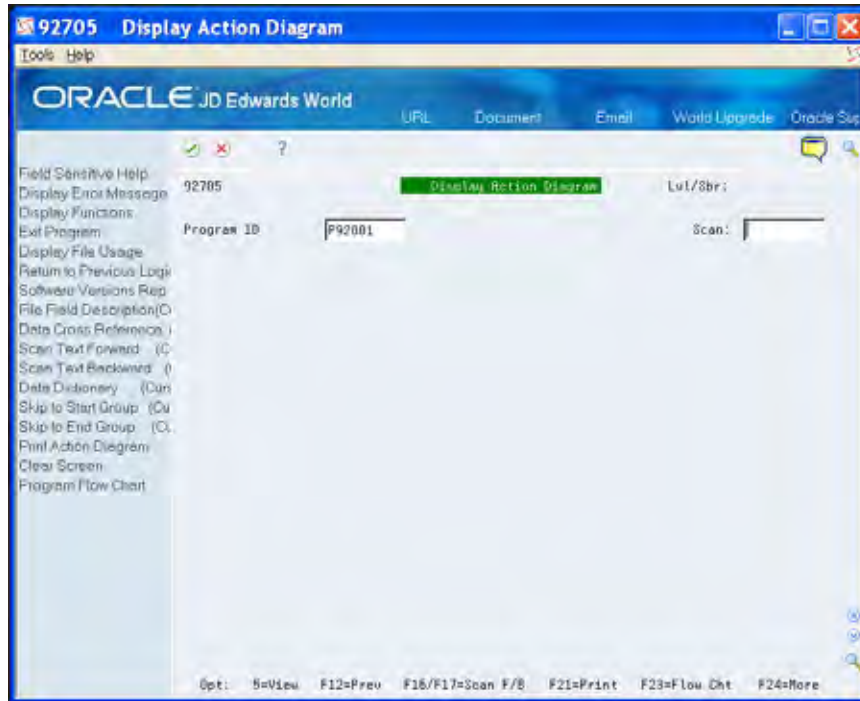
When you view an action diagram, you are viewing a graphical representation of the code's hierarchy within the program and how different subsets of code relate to other subsets of code. You can view the code for a subroutine from the program or exit to facilities that show more detail for fields, files,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

To view an action diagram

	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.



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.

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.

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.
	Data Dictionary (F18)	Displays the Data Dictionary definition for the data item.
Files	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.


Option Field Values**View (5)**

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

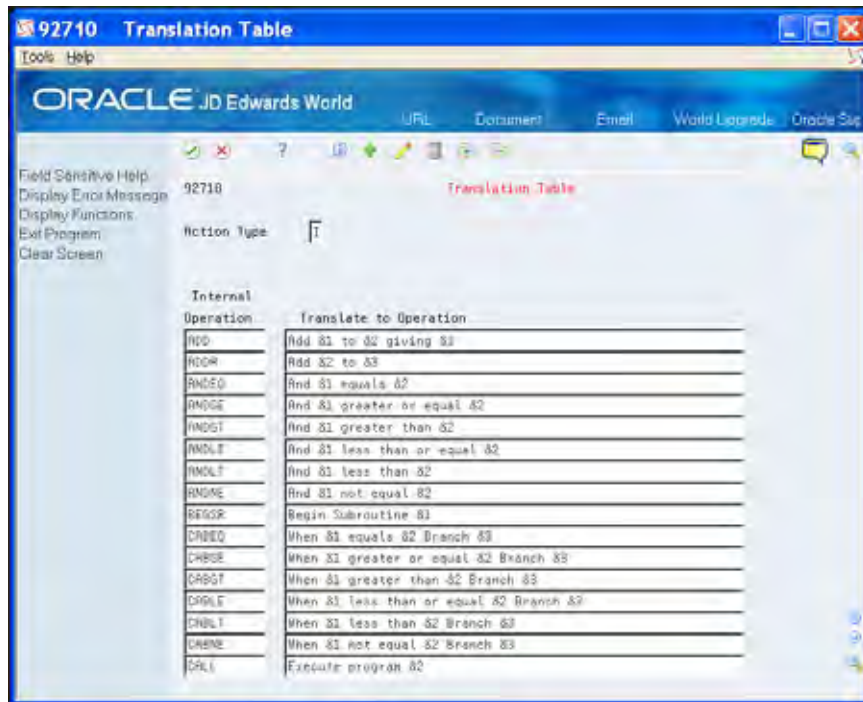
Accessing the Logic Translation Feature

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

	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.



8 Source Code Inventory and Database

Overview to Source Code Inventory and Database

Objectives

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

About 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

When you use the program generator, it is important that you understand how the system manages the source code in the program. This includes the following key elements that the system assigns:

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

Source Serial Numbers

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

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.

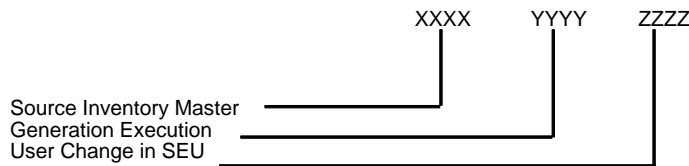
Element	Description
Date Last Change	When the program generator creates a program, it places the date you add or change the code within the source sequence line.

The following illustrates the parts of the source sequence line.

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

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

Structure of the Serial Number



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.

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.

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

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.

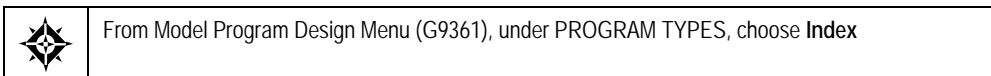
JD Edwards World provides you with program types for the most common programs. You can create your own program types for your organization's needs.

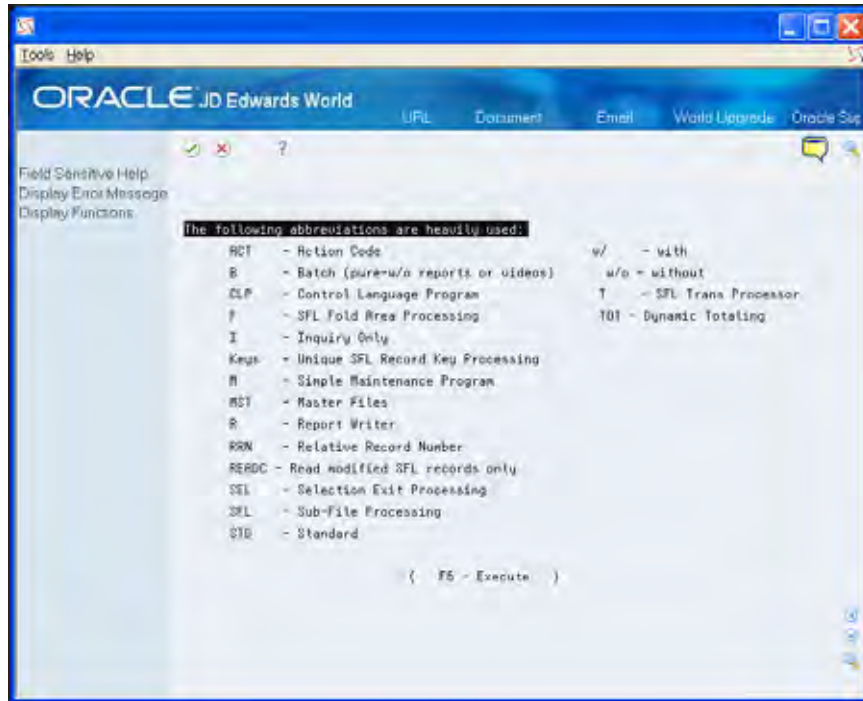
This chapter includes the following tasks:

- [Reviewing Abbreviations for Program Types](#)
- [Reviewing Program Types Index](#)
- [Reviewing Program Types Cross Reference](#)
- [Creating or Modifying Program Types](#)

Reviewing Abbreviations for Program Types

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





Reviewing Program Types Index

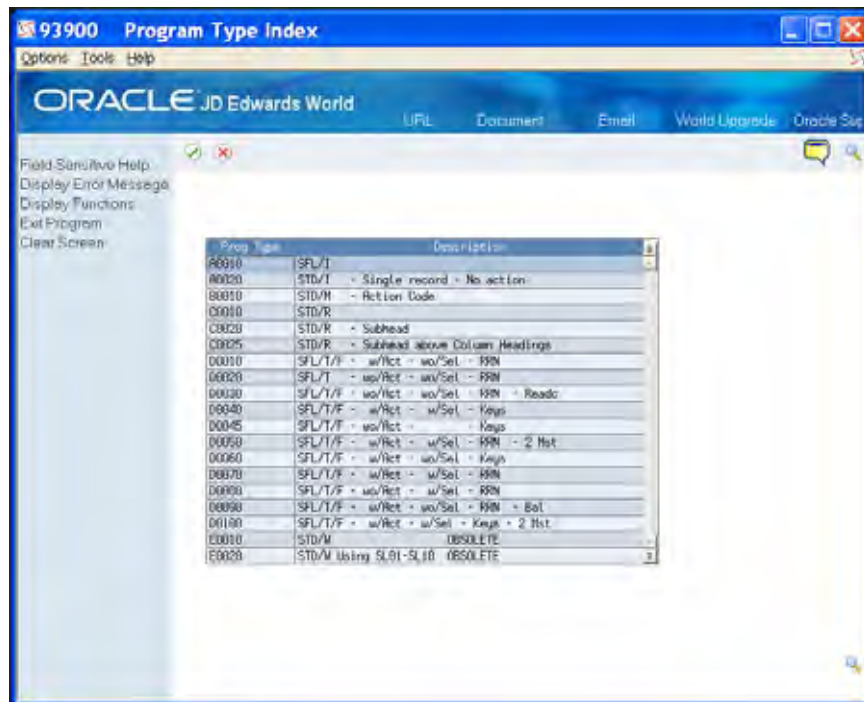
You can use the Program Type Index program (P93900) to locate program types. You can access this from the Program Types Cross Reference screen, Index screen, as well as the Create/Modify screen.

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.

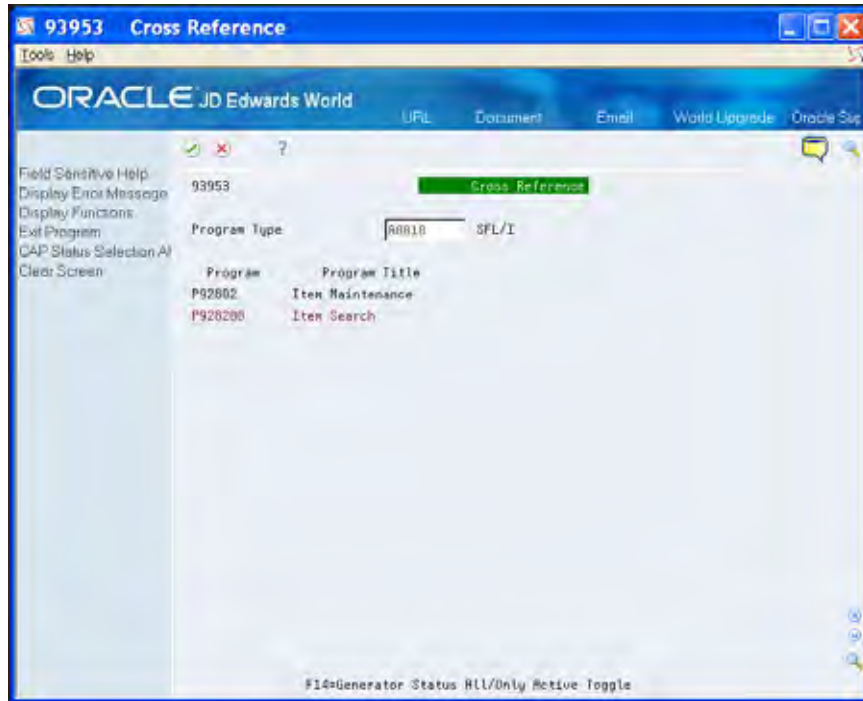


Reviewing Program Types Cross Reference

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



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



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.

Creating or Modifying Program Types

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

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.

Primary Module

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

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

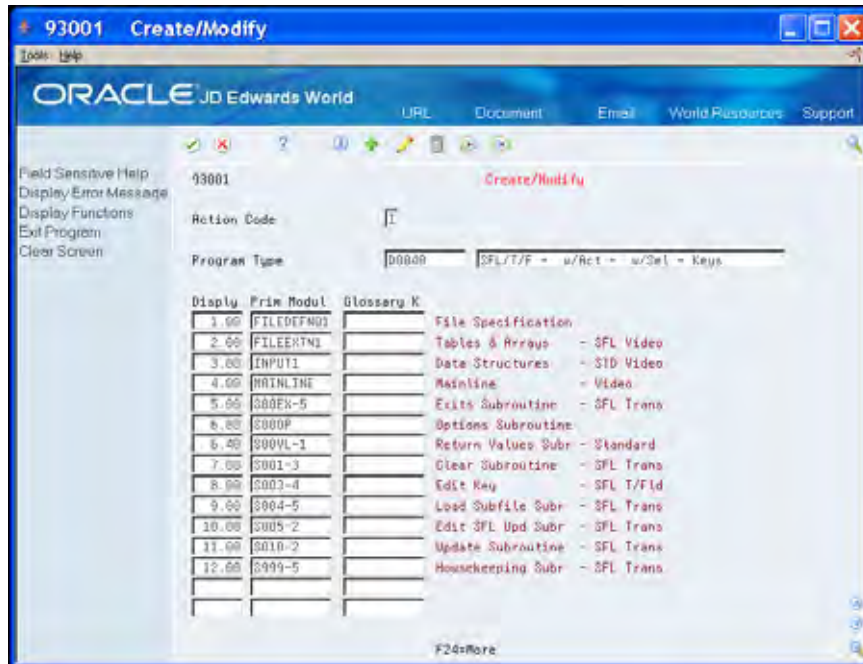
Ensure you are of the alpha order requirement when creating new program types.

The following is an example using program type D0040.



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

On Create/Modify, locate an existing program type.



Work with Logic Modules

There are two types of logic modules:

- Primary
- Detail

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

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 *Understand Directives* for more information about directives.

Generation Options

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

Help Instructions Edit/Build

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

All Help Instructions

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

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::

- [Viewing the Logic Module Index](#)
- [Viewing Logic Module Cross Reference](#)
- [Viewing Logic Module Op Codes](#)
- [Maintaining the Logic Module File](#)
- [Creating or Modifying Logic Modules](#)
- [Creating or Modifying Formula Library Entry](#)
- [Copying or Moving Program Specifications](#)
- [Printing Program Generator Specifications](#)
- [Reviewing Source Modifications](#)
- [Using Program Generator Updates](#)
- [Using CASE Specifications Inquiry](#)

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



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



Viewing Logic Module Cross Reference

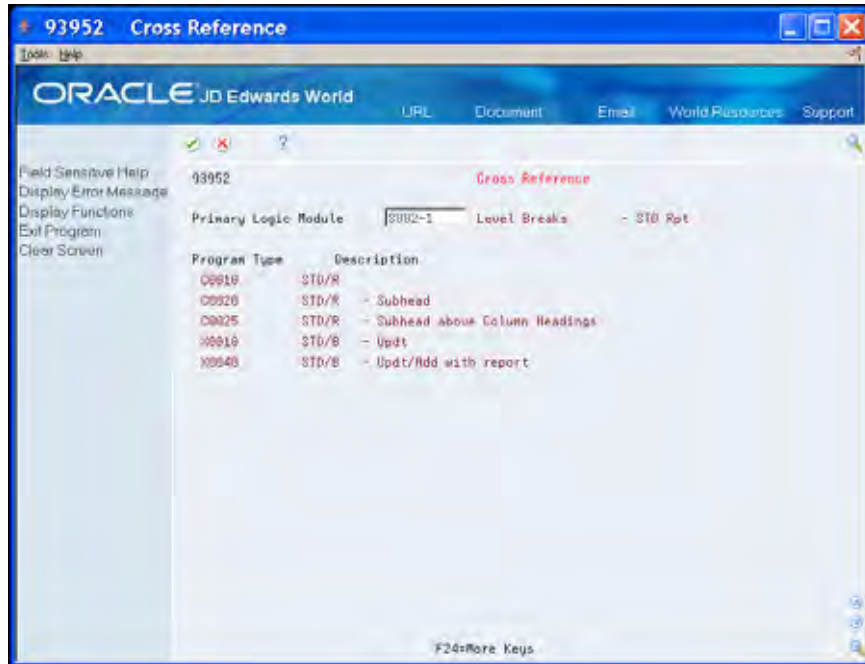
The Logic Module Cross Reference allows you to determine which program types use a particular logic module.

To view the logic module cross reference



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

Enter a primary logic module name.



Viewing Logic Module Op Codes

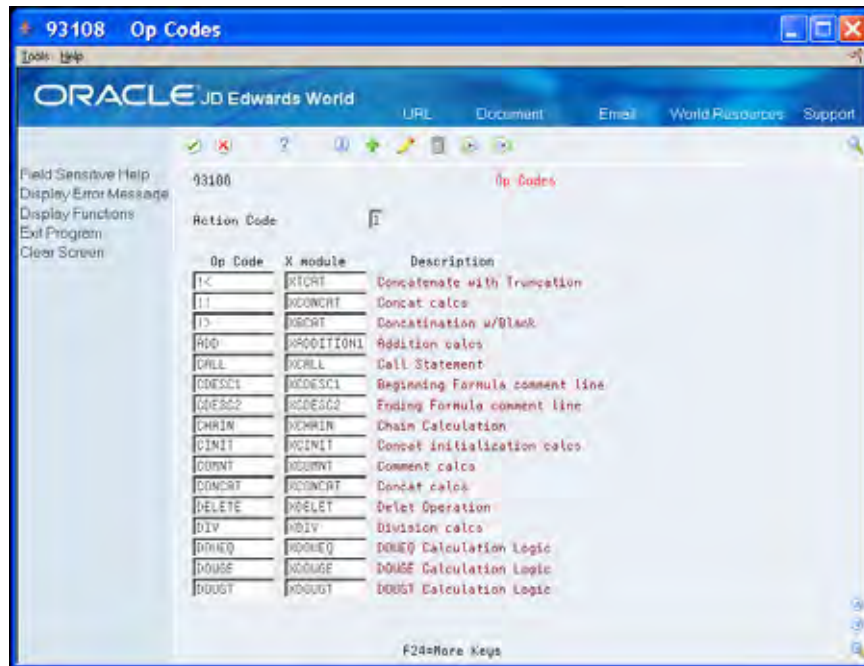
On the Op Codes screen, the:

- 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

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



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.

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).

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).

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:

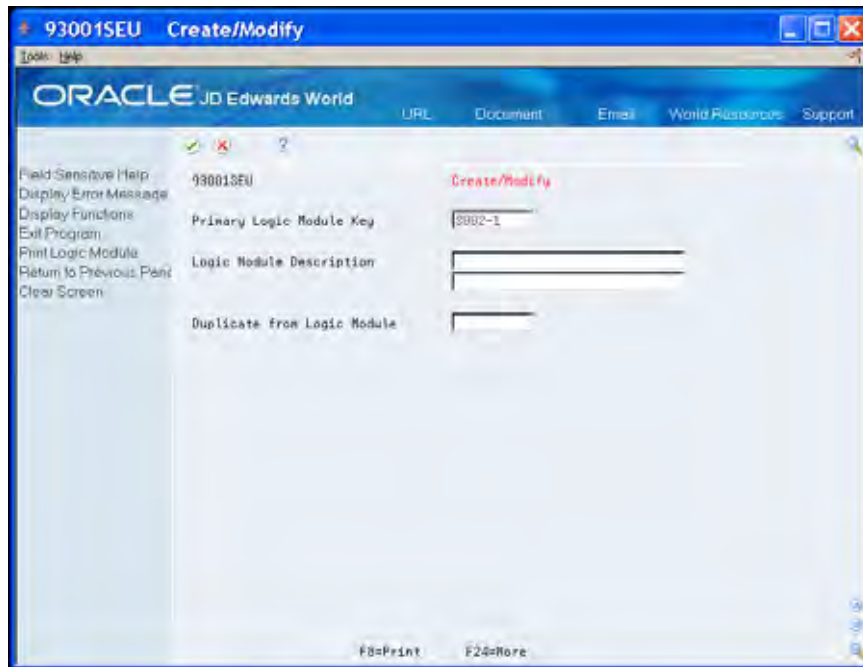
- Creates a work file in QTEMP/F93001WRK.
- Adds a member to F93001WRK.
- Clears the member in F93001WRK.

You can exit the code without saving your changes.

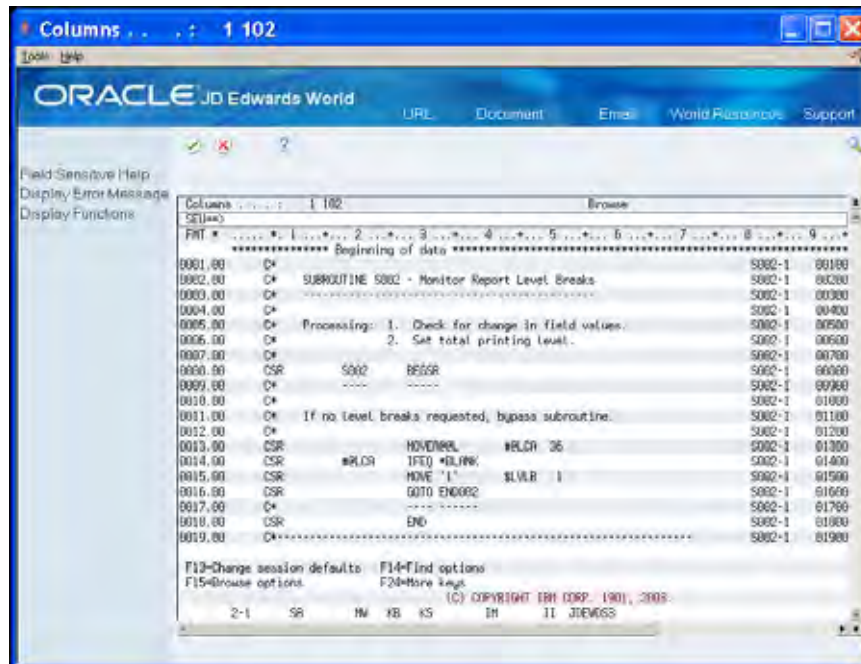
To create or modify logic modules

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

1. Enter a logic module name.



2. Create or change the appropriate lines of code




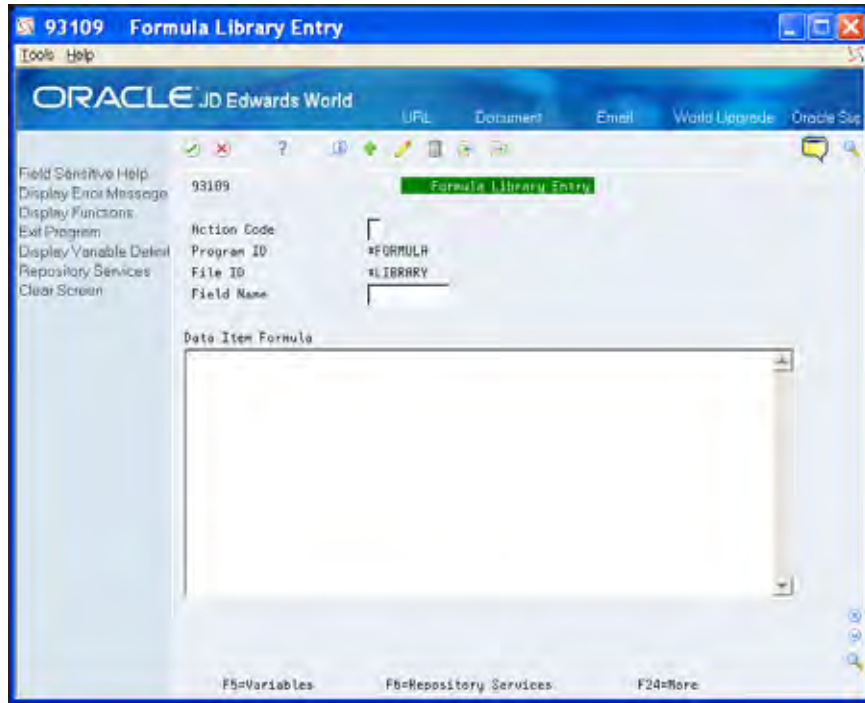
Creating or Modifying Formula Library Entry

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

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



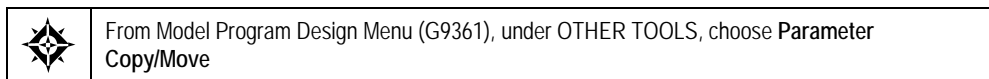
Copying or Moving Program Specifications

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

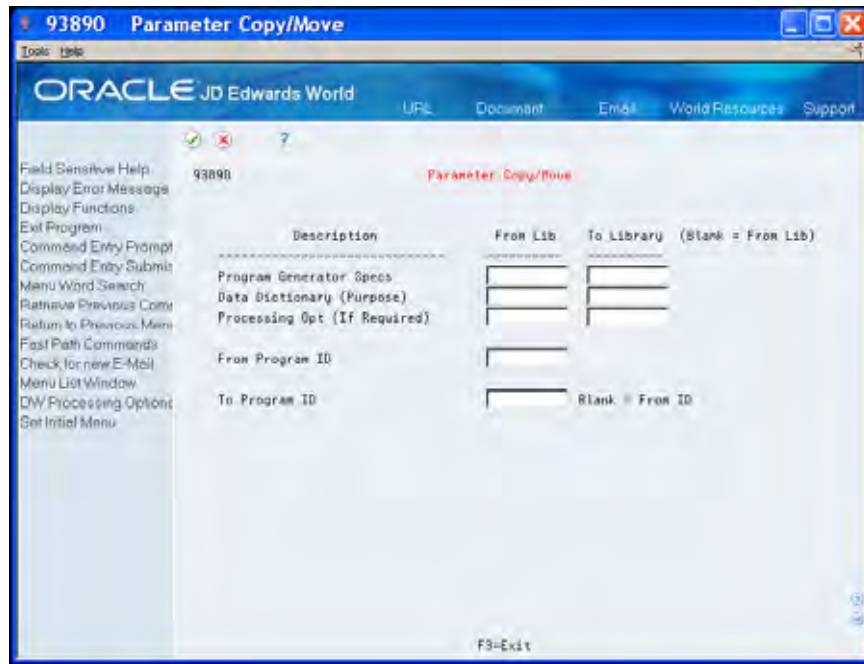
Alternatively, you can enter 3 in the Option field on Software Versions Repository to copy Program Generator specifications within a library.

To copy or move program specifications



On Parameter Copy/Move complete the following fields:

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




Printing Program Generator Specifications

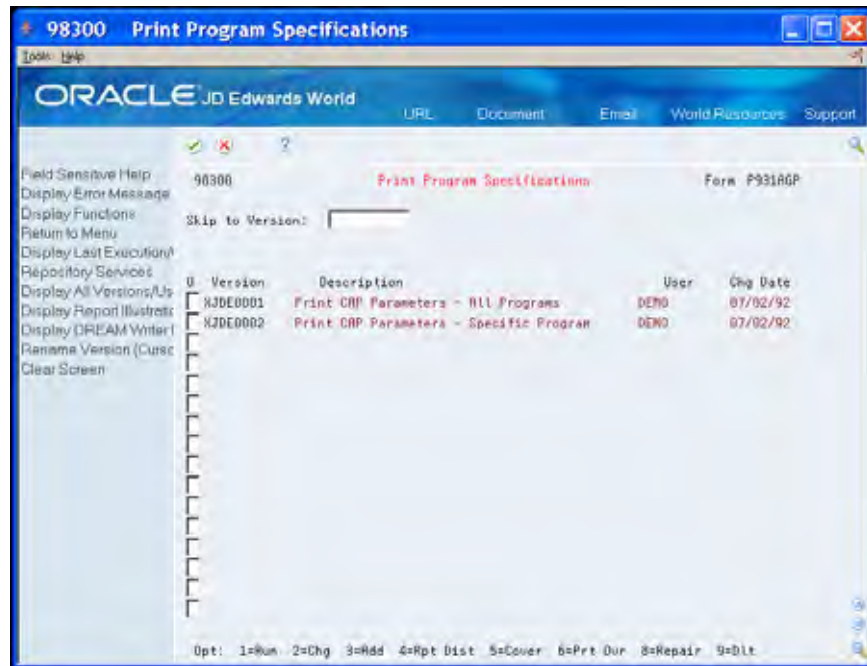
Use the Print Program Specifications program to print the program specifications. You must use a logical file.

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

	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.




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 Pxxxxx member in the Additional Help/Modifications Master file (F93002).

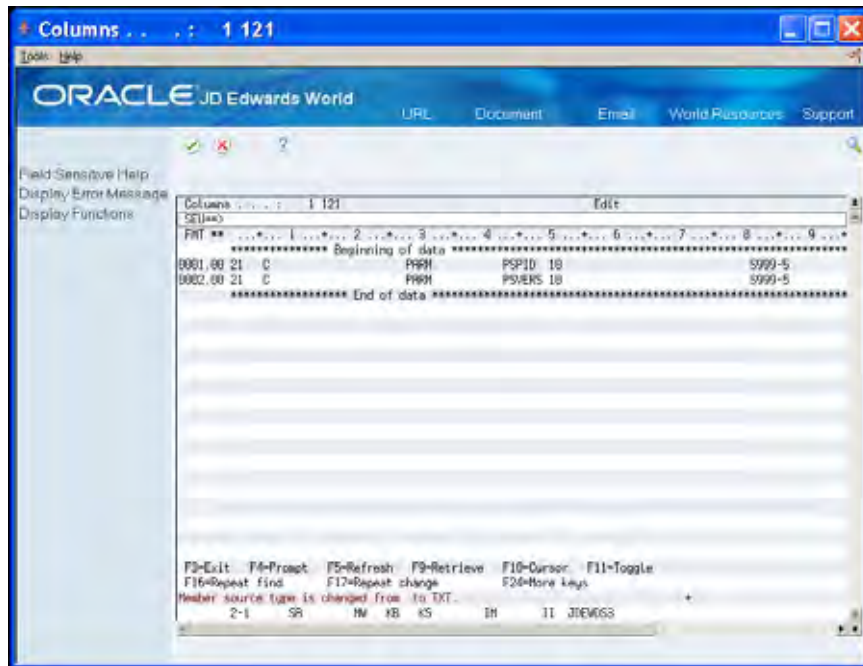
The lines of code are the result of the MPxxxxx job that runs and compares the before image of the source code with the source code after the user makes changes and stores the code in the Pxxxxx member in the F93002.

To review source modifications

	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.




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

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

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

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

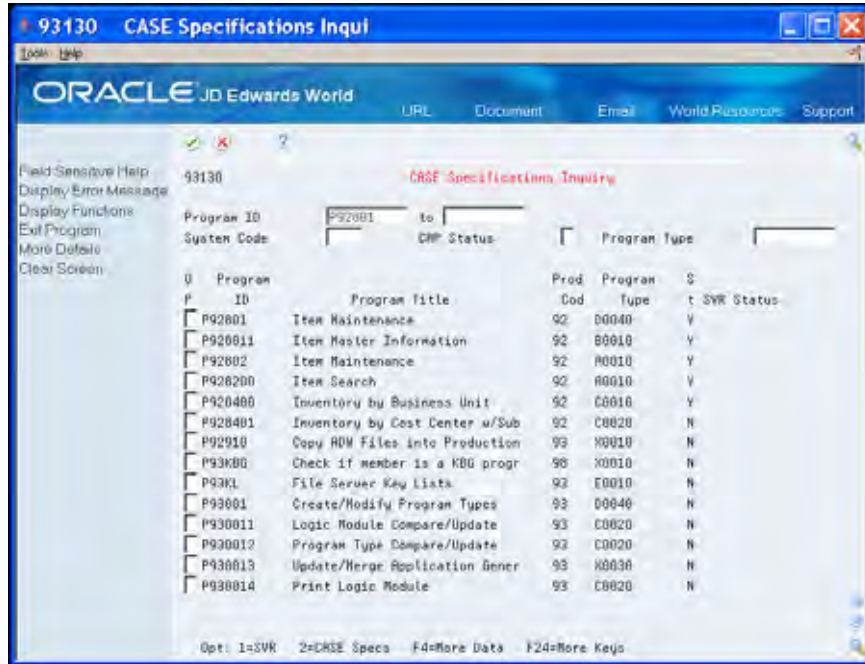
1. On CASE Specifications Inquiry, complete any of the following fields:

- Program ID
- System Code
- CAP Status
- Program Type

The system displays the records that meet your search criteria.

2. Complete the following field:

- Option



Understand 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.

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
AUTHR	None	F spec	Program author
*AUTOI	X*ENTRYI	S999	Automatic inquiry at execution test logic
*CLRN	None	S001	Clear user requested fields

Directive Code	Detail Logic Module	Source Created	Functional Directive
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			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

Directive Code	Detail Logic Module	Source Created	Functional Directive
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	XLOADEMK	S999	Load user defined error messages
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
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n=Master - Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master - Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTC	S005	Cost center edit
	XFIELDDEDTE	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTS	S005	Validation = Master - Alpha
	XFIELDDEDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master - Numeric
FILES	None	F spec	Program file descriptions

Directive Code	Detail Logic Module	Source Created	Functional Directive
*FLDxx			Active Data Dictionary field validation for primary data
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n=Master - Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master - Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTC	S005	Cost center edit
	XFIELDDEDTE	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTS	S005	Validation = Master - Alpha
	XFIELDDEDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master - Numeric
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
	XNEXT-NBR	S003	Load video input - Next Numbering

Directive Code	Detail Logic Module	Source Created	Functional Directive
KEYS2			Load master file key fields in primary video format
	XFIELDLD1	S005	Load video input - Alpha
	XFIELDLD2	S005	Load video input - Numeric
	XFIELDLD3	S005	Load video input - Cost Center
	XFIELDLD4	S005	Load video input - Julian Date
	XFIELDLD5	S005	Load video input - Gregorian Date
	XNEXT-NBR	S005	Load video input - Next Numbering
KLIST	XKEYLIST	S999	Create data file key list
*LVLS	XSAVVAL1		Save report level break data
MF	None	Various	Variable name substitution for master database files
*MCUxx	None	S003	Business Unit security logic where xx = master field designation 1 thru 9
		S004	
		S00EX	
*OPEN	XFILEOPN1	S999	Open report program data files
OPTE	None	S005	Subfile processing condition test based on mandatory entry fields in subfile format
*OTOT	XPRTTOT1	S010	Print all report level totals
PDL	None	Various	User defined entry point
*RKYxx	None	S999	Load softcoding record key for reports where xx = master file designation 1 thru 9
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

Directive Code	Detail Logic Module	Source Created	Functional Directive
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	XTOTARRY	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.
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n = Master - Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master - Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTC	S005	Cost center edit
	XFIELDDEDTE	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTS	S005	Validation = Master - Alpha
	XFIELDDEDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master- Numeric
SLDxx			Active Data Dictionary data field validation for subfile data fields. Where xx = specified master file 1 thru 9.
	XFIELDDEDT1	S005	Data Dictionary alpha edit
	XFIELDDEDT2	S005	Validation n = Master - Alpha
	XFIELDDEDT3	S005	Gregorian edit
	XFIELDDEDT4	S005	Julian edit
	XFIELDDEDT5	S005	Data Dictionary numeric edit
	XFIELDDEDT6	S005	Alpha field size 10
	XFIELDDEDT7	S005	User defined code edit
	XFIELDDEDT8	S005	No dictionary
	XFIELDDEDT9	S005	Validation n = Master - Numeric
	XFIELDDEDTA	S005	Account ID
	XFIELDDEDTC	S005	Cost center edit
	XFIELDDEDTC	S005	Numeric field size 7
	XFIELDDEDTR	S005	Right adjust
	XFIELDDEDTS	S005	Validation = Master - Alpha
	XFIELDDEDTT	S005	Validation = Master - Alpha Rt Adj
	XFIELDDEDTU	S005	Validation = Master- Numeric

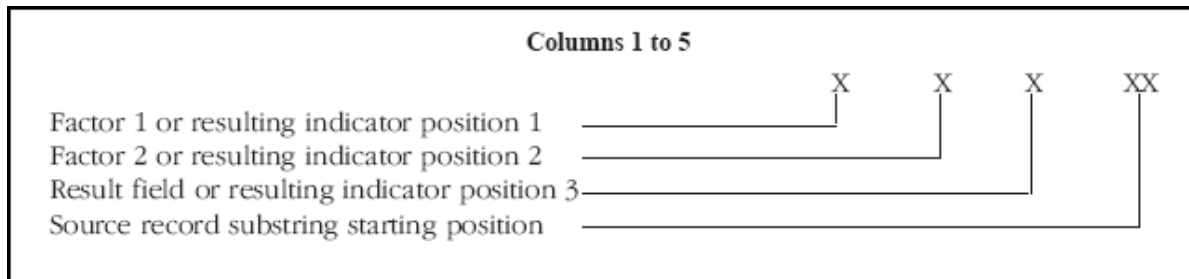
Directive Code	Detail Logic Module	Source Created	Functional Directive
*S00VL	None	I spec	Cursor Control, F1
TITLE	None	H spec	Program title
*VKYxx	None	S999	Load softcoding record key for display files where xx=display file designation 1 - 9.
*VTS	None	I spec	Display file softcoding array
*VTX	None	I spec	Update softcoding text field ending positions based upon size definition in display file
*VTXI	XVTXIDX	S999	Set maximum VTX index used

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

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.



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

Directive	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.

Directive	Column Allowed				Function
B					Unused at this time.
C	x				Function key exit indicator test
D	x			x	Descriptions for fields, files, and copy modules
E		x			Error message key
F		x			Validation file name
G		x			User defined calculation logic result field name
H		x			Descriptive display file name
I	x	x	x		Display field error condition attribute indicator
J	x	x	x	x	Data file names
K	x	x	x		Descriptive display file key field name
L	x				Data file key list name and optional file/format name
M	x				File information data structure name
N	x	x	x		Full data field name (write to)
O	x			x	Common subroutine name
P		x			Function key/selection exit program to execute
Q			x		Field name to receive description value
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)

Directive	Column Allowed			Function
0	x			Gregorian date Data Structure numeric 6 byte date
1	x			Gregorian date Data Structure numeric 2 byte month
2	x			Gregorian date Data Structure numeric 2 byte day
3	x			Gregorian date Data Structure numeric 2 byte year
4	x	x	x	Parameter 1 from *PROC calculations
5	x	x	x	Parameter 2 from *PROC calculations
6	x	x	x	Parameter 3 from *PROC calculations
7	x	x	x	Parameter 4 from *PROC calculations
8	x	x	x	Parameter 5 from *PROC calculations

Exception Directives

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

Factor 1	Factor 2	Result	Keyword	Function
	x		&xxFORMATC	Subfile control record format name
	x		&xxFORMATS	Subfile record format name
x	x		&xxKEYFLD	Master file primary key field name
x	x		&xxPGCTL	Number of subfile records in 1 page
x			%	Factor 1 intentionally left blank
x	x	x	=	User defined calculation logic result

Conditional Directives

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

Position/Factor/Result	Description
	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
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

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.

The question and answer system includes the following:

- [Reviewing Questions in a Master Dialogue](#)
- [Adding New Q & A Dialogue](#)
- [Working with an Existing Dialogue](#)

	From Model Program Design Menu (G9361), choose Maintain Q/A From World CASE Q & A Menu (G9364), choose Simple Question and Answer
--	--

About Simple Question & Answer

The Simple Question & Answer screen, from which you begin all Question & Answer tasks, includes the following three fields.

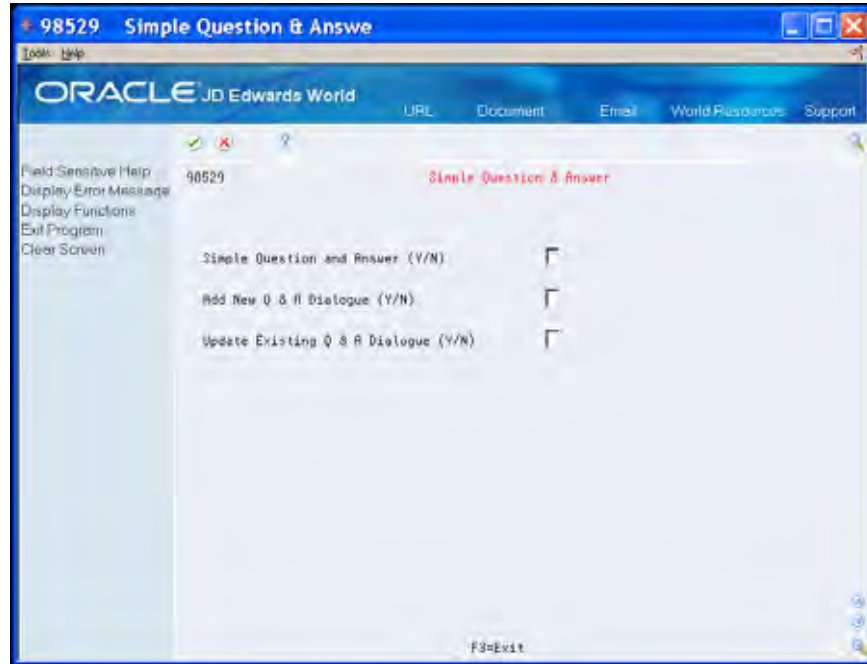
Option	Description
Simple Question and Answer	Access the Question Entry screen.
Add New Q & A Dialogue	Access the Dialogue Descriptions screen.
Update Existing Q & A Dialogue	Access the Dialogue Lists screen.

Reviewing Questions in a Master Dialogue

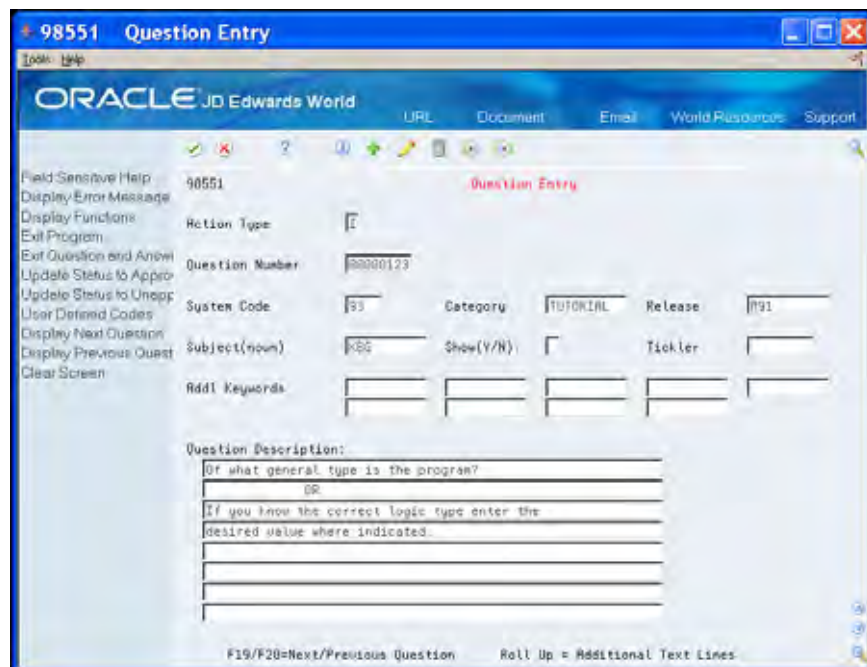
You choose Display Next Question (F19) and Display Previous Question (F20) to review all of the master questions. Additionally, you can make changes to the master question that displays on the Question Entry screen.

To review questions in a master dialogue

1. On Simple Question and Answer, enter Y in the following field:
 - Simple Question and Answer



2. On Question Entry, complete the following field:
 - Question NumberThe question detail displays.



3. To review the answers to the master question Click Change.

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

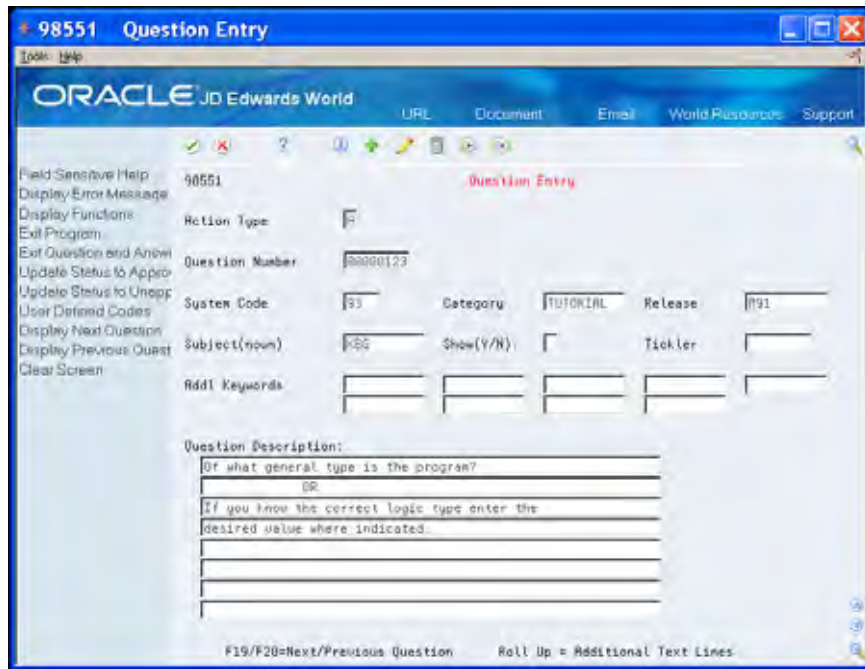
1. On Simple Question & Answer, enter Y in the following field:
 - Add New Q & A Dialogue

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

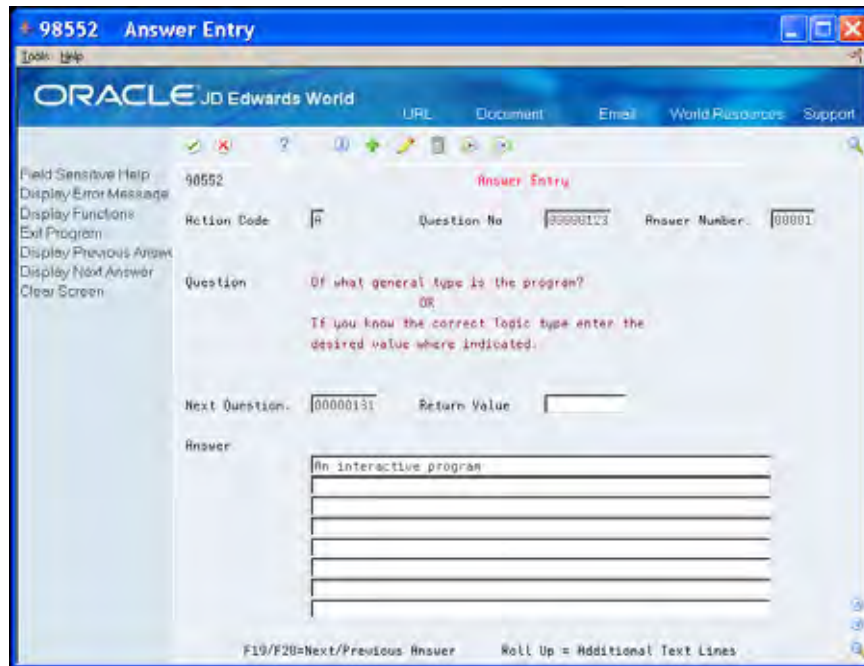
The first Question Entry screen displays.

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



The Answer Entry screen displays.

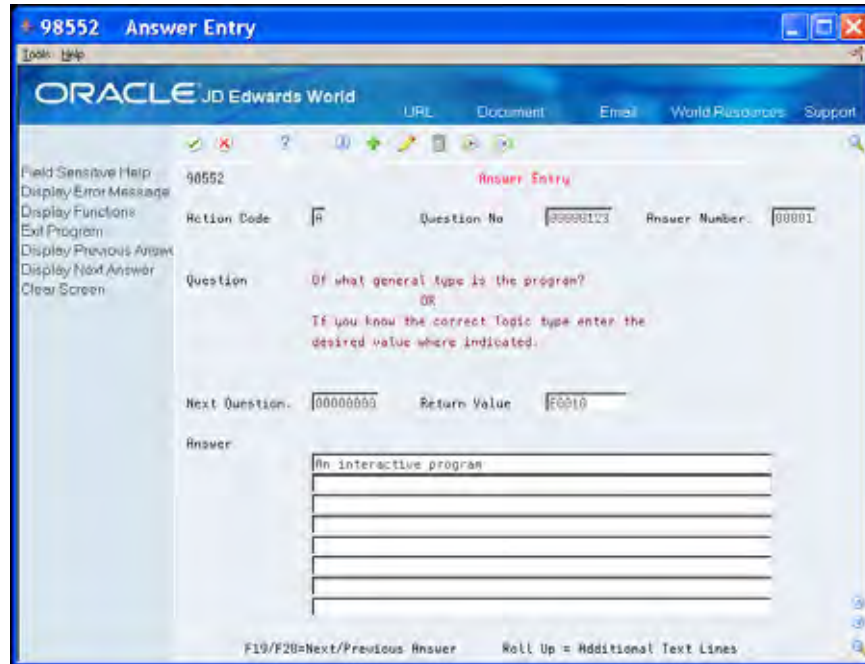


5. Complete the following fields:
 - Question Number
 - Answer Number
 - Next Question
 - Answer
6. Optionally, complete the Return Value field.

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.



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.

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](#)

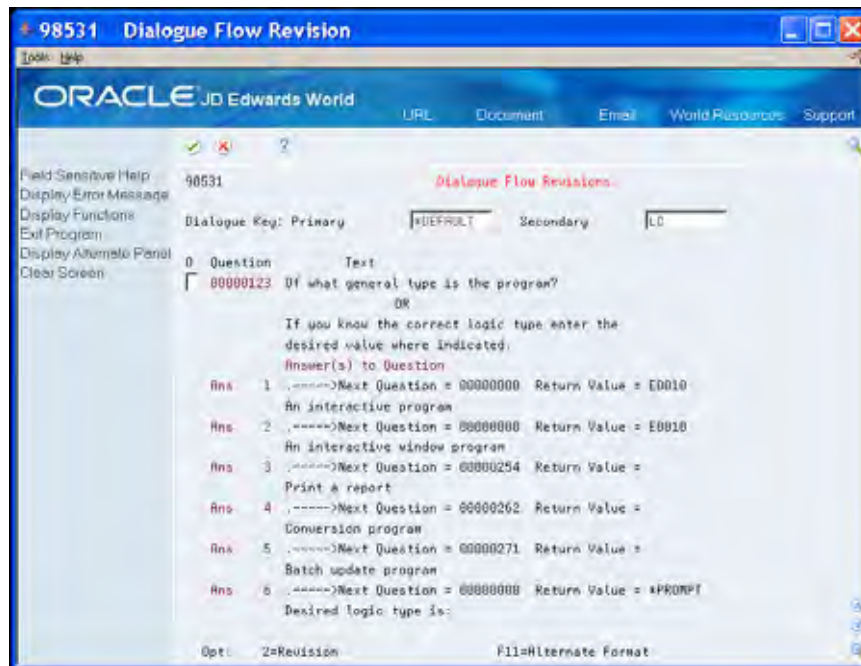
To perform these tasks, choose Update Existing Q & A Dialogue on the Simple Question and Answer screen to access the Dialogue Lists screen.

On Dialogue Lists, you can use the following values in the Option field:

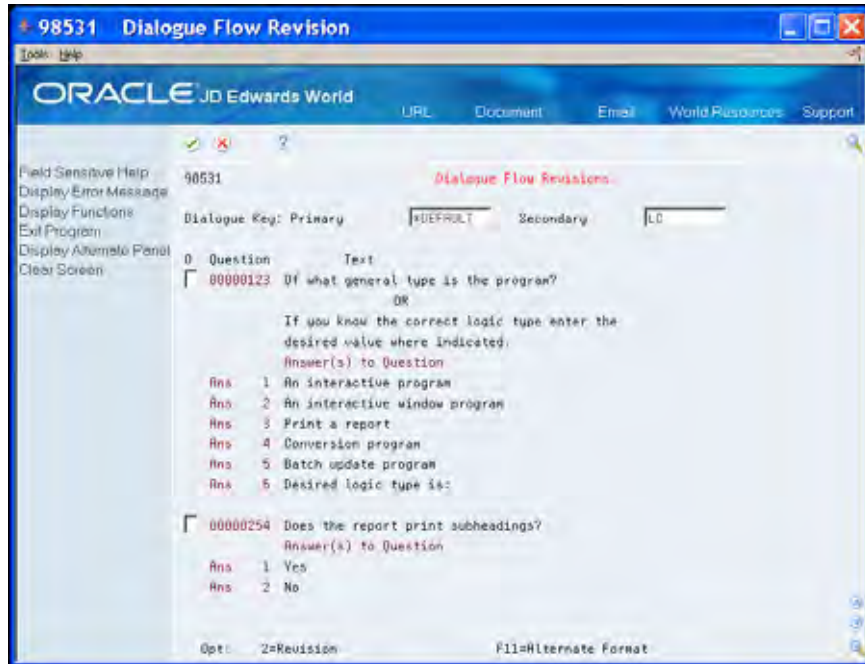
- 2 - Change the questions and answers for the Dialogue
- 3 - Copy one dialogue to another dialogue
- 5 - Run the Q & A. You can specify the number of responses to allow.
- 6 - Flow displays the flow of the Q&A and how one question leads to another. You can access Q&A Revisions from the flow.
- 7 - Rename
- 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.



2. Choose Alternate Format (F11) to view the Alternate Format.



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.



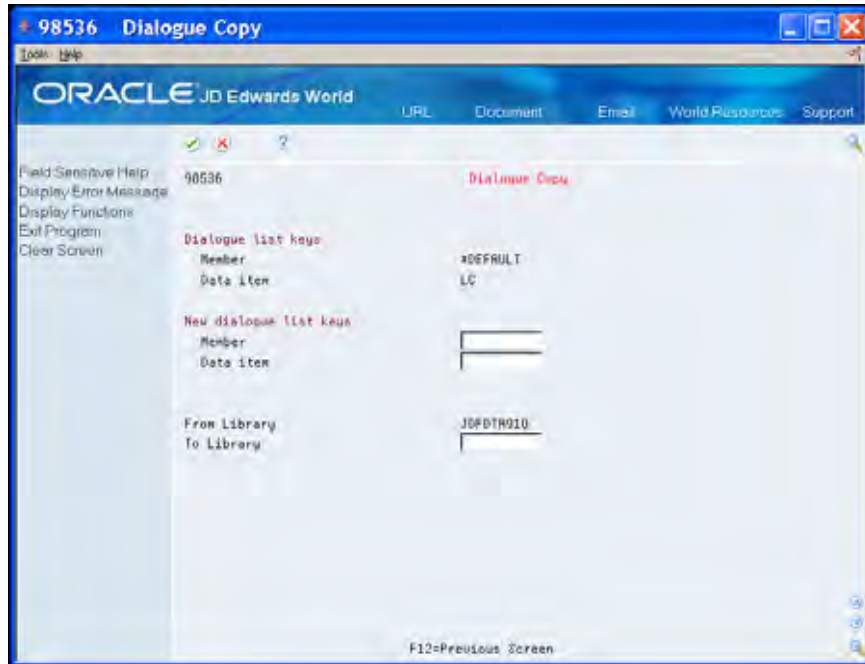
2. Enter the name of the new dialogue list keys (primary key) in the following field:
 - Member
3. Enter the name of the new dialogue list keys (secondary key) in the following field:
 - Data item
4. Enter the name in which the new dialogue resides in the following field:
 - To Library

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.

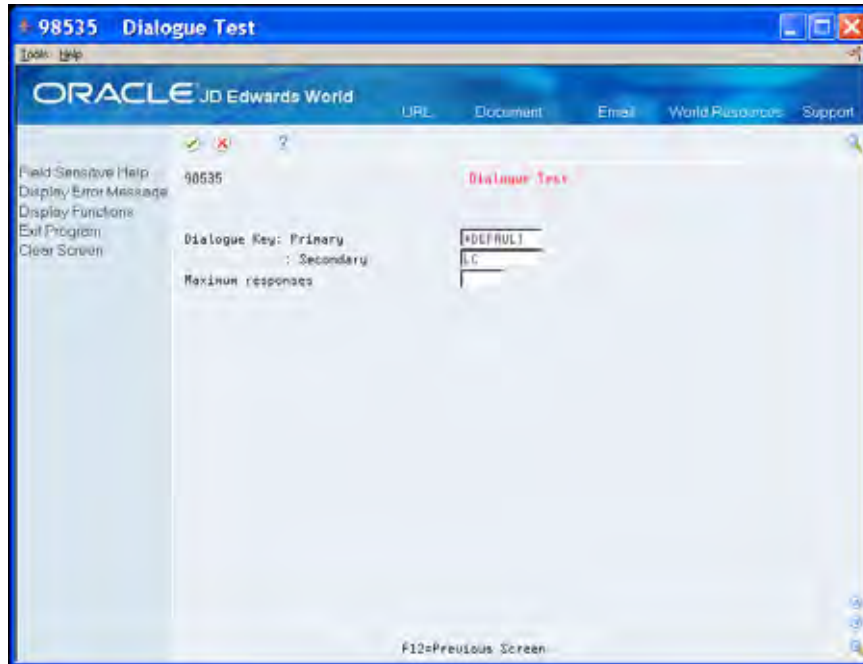
The Dialogue Copy screen displays.



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 LibraryThe 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.

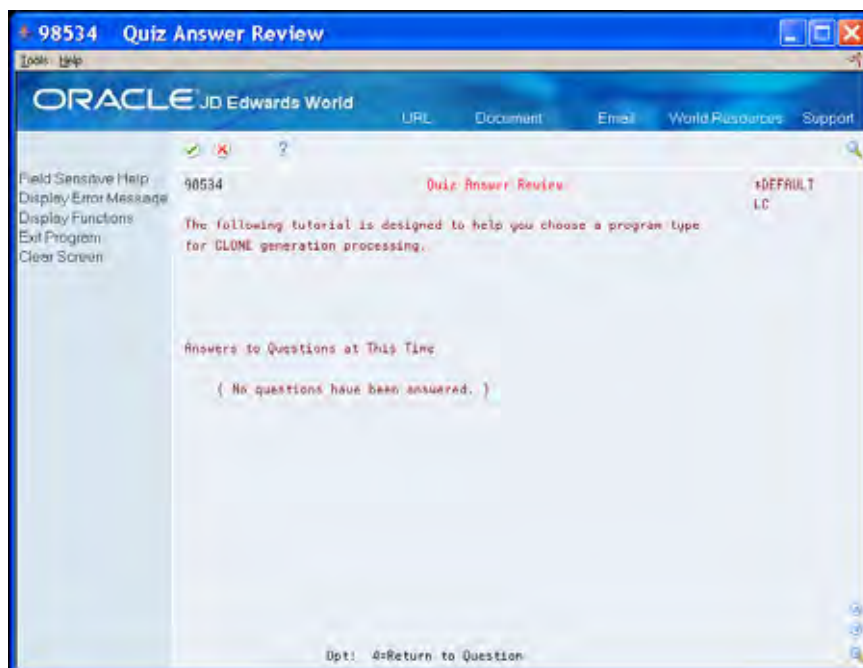


2. Enter a number for the maximum number of times you want to run this dialogue in the following field:

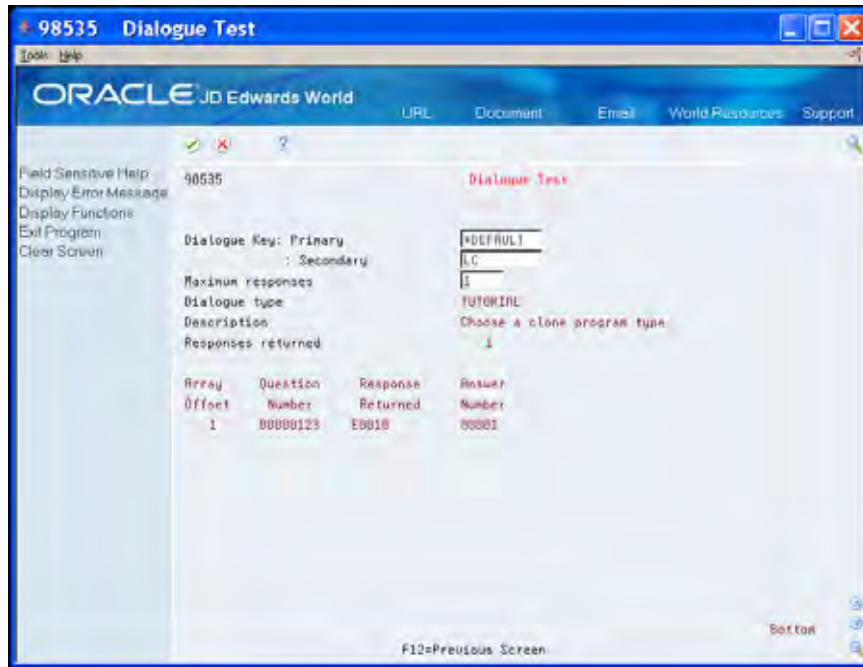
- Maximum responses

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.



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.



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.

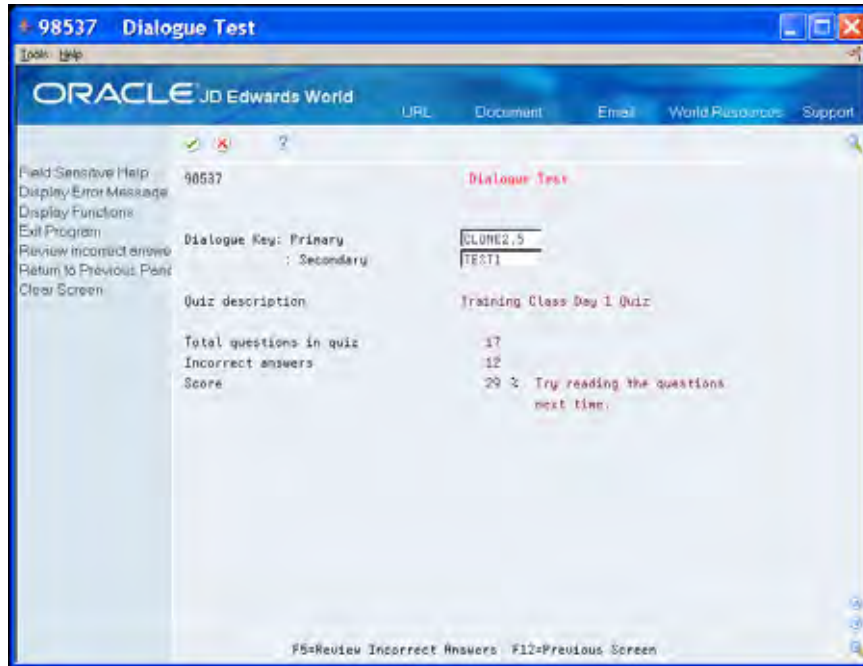
1. On Dialogue Lists, enter 11 in the Option field next to the quiz you want to run.
The first question of the quiz displays.
2. Answer the questions.

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.



Create 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

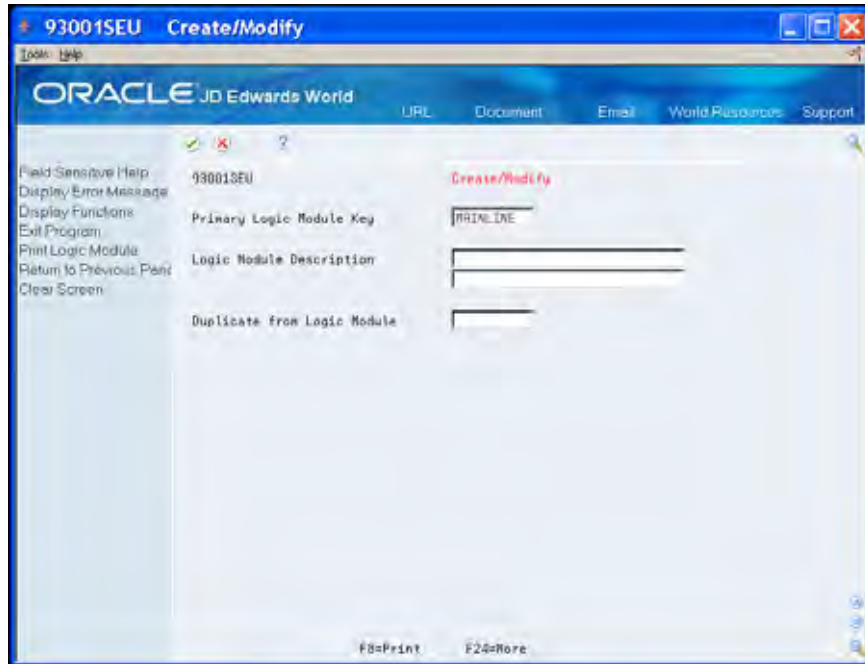
You must be able to locate program types and logic modules. See *Create or Modify Program Types* and *Work with Logic Modules*.

To create user defined PDL



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

1. On Create/Modify (Logic Module), access the Edit screen.

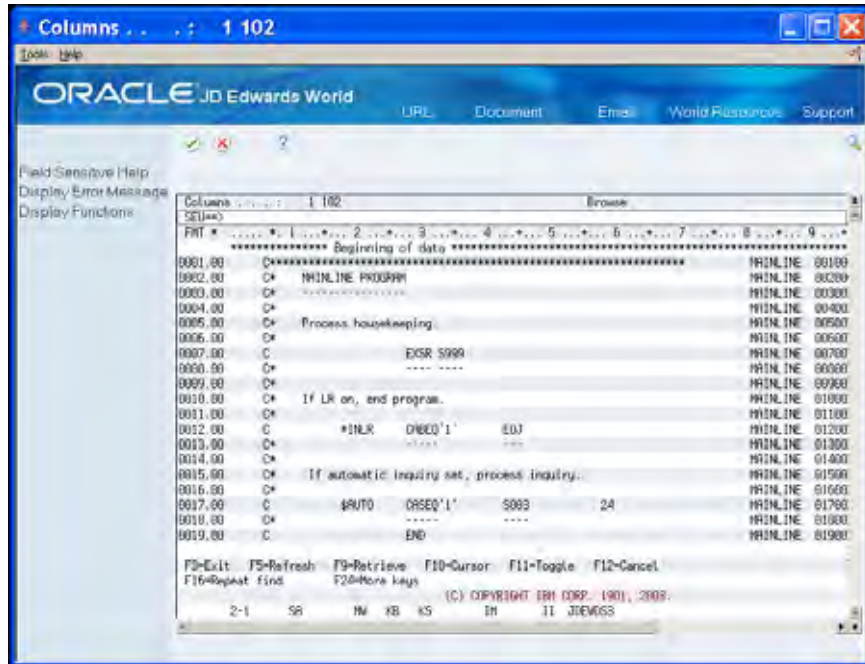


2. On the Edit screen, create the user defined PDL entry points within the affected primary logic modules.

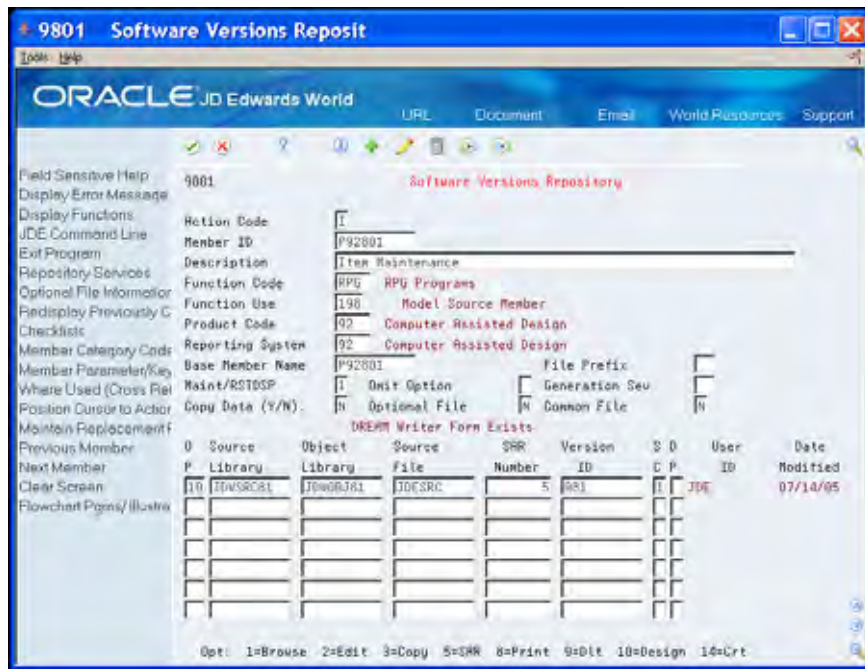
The naming convention for user defined PDL entry points is PDLxx, where xx is a two digit number between 01 and 99.

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.)

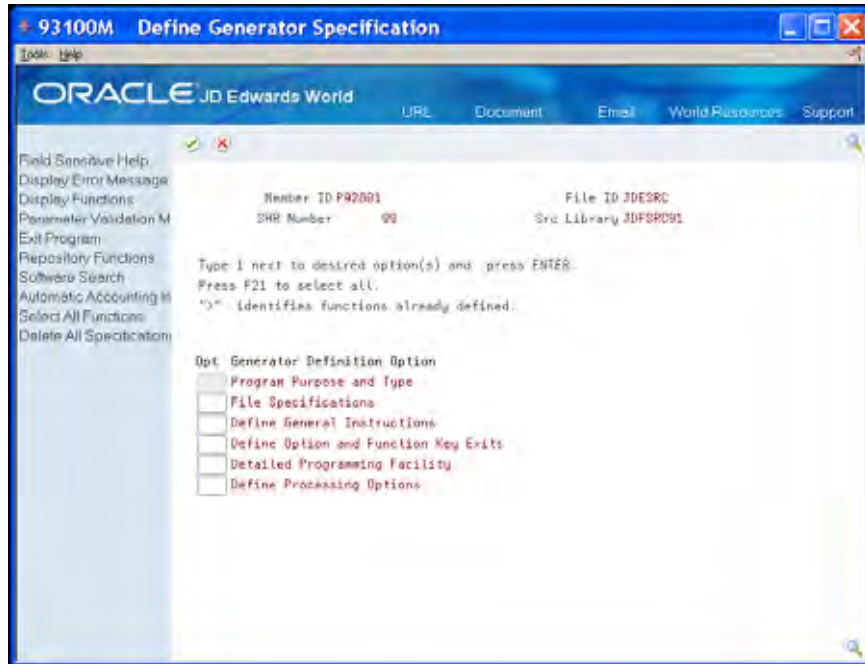
In this example, you enter PDL01 on line 9. Any PDL code that you enter for this entry point will come immediately after the statement EXSR S999 and before the test for *INLR.



3. Access the Software Versions Repository and locate the member.

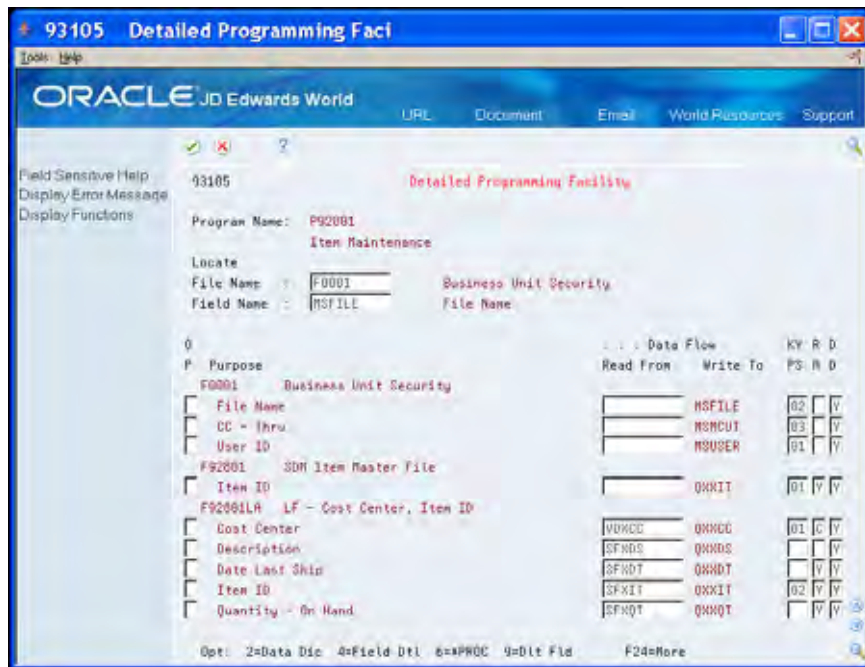


4. Enter 10 (Design) next to the environment in the Option field:
The Program Generator Specification screen displays.



5. On Define Generator Specification, enter 1 in the Option field to access the Detailed Programming Facility:

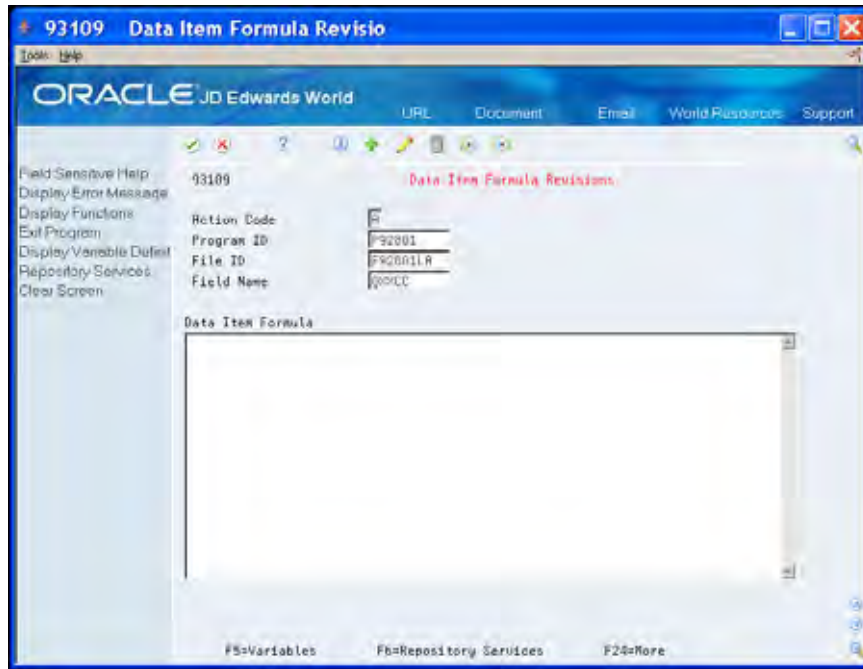
The Detailed Programming Facility screen displays.



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.



9 Appendices

Appendix A - Program Generator Checklist

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.

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.

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.

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

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.

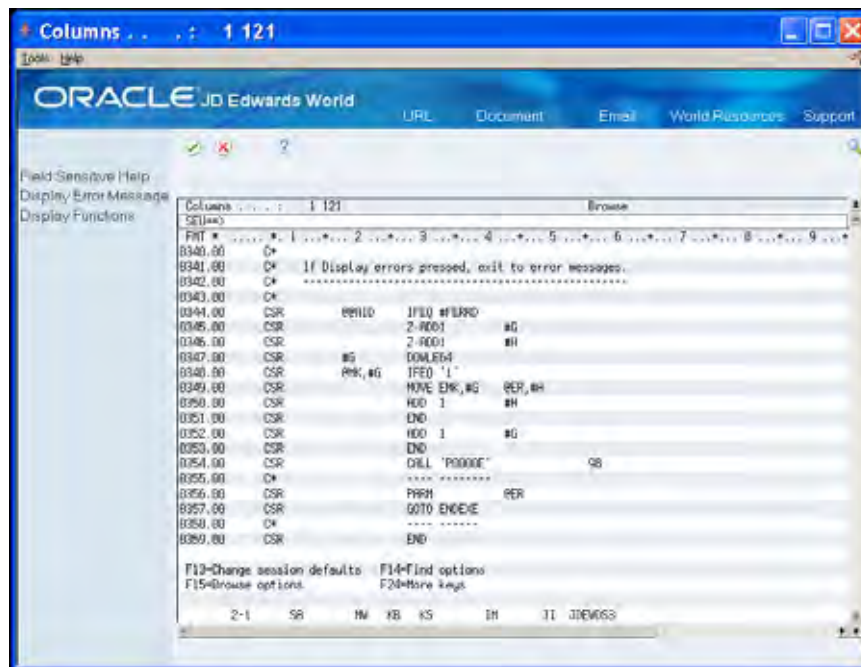
Appendix B - Programming Standards

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.



The screenshot shows the Oracle JD Edwards World interface for the Single Record Maintenance Program (SRM). The window title is "Columns . . . : 1 121". The main display area shows a list of error codes and their corresponding data dictionary names. The code is as follows:

```
0340.00 C*
0341.00 C* If Display errors pressed, exit to error messages.
0342.00 C*
0343.00 C*
0344.00 CSR      @ER10  IFED #IFED
0345.00 CSR      @ER01  2-0001  #G
0346.00 CSR      @ER01  2-0001  #H
0347.00 CSR      @G     DONLE64
0348.00 CSR      @MK,@G  IFED 'L'
0349.00 CSR      NONE @MK,@G  @ER,@H
0350.00 CSR      @ER01  @H
0351.00 CSR      @ER01  @H
0352.00 CSR      @ER01  @G
0353.00 CSR      @ER01  @G
0354.00 CSR      @ER01  'P0000'  @B
0355.00 C*
0356.00 CSR      @ER01  @ER
0357.00 CSR      GOTO ENDEIE
0358.00 C*
0359.00 CSR      @ER01  @ER
0360.00 CSR      @ER01  @ER
0361.00 CSR      @ER01  @ER
0362.00 CSR      @ER01  @ER
0363.00 CSR      @ER01  @ER
0364.00 CSR      @ER01  @ER
0365.00 CSR      @ER01  @ER
0366.00 CSR      @ER01  @ER
0367.00 CSR      @ER01  @ER
0368.00 CSR      @ER01  @ER
0369.00 CSR      @ER01  @ER
0370.00 CSR      @ER01  @ER
0371.00 CSR      @ER01  @ER
0372.00 CSR      @ER01  @ER
0373.00 CSR      @ER01  @ER
0374.00 CSR      @ER01  @ER
0375.00 CSR      @ER01  @ER
0376.00 CSR      @ER01  @ER
0377.00 CSR      @ER01  @ER
0378.00 CSR      @ER01  @ER
0379.00 CSR      @ER01  @ER
0380.00 CSR      @ER01  @ER
0381.00 CSR      @ER01  @ER
0382.00 CSR      @ER01  @ER
0383.00 CSR      @ER01  @ER
0384.00 CSR      @ER01  @ER
0385.00 CSR      @ER01  @ER
0386.00 CSR      @ER01  @ER
0387.00 CSR      @ER01  @ER
0388.00 CSR      @ER01  @ER
0389.00 CSR      @ER01  @ER
0390.00 CSR      @ER01  @ER
0391.00 CSR      @ER01  @ER
0392.00 CSR      @ER01  @ER
0393.00 CSR      @ER01  @ER
0394.00 CSR      @ER01  @ER
0395.00 CSR      @ER01  @ER
0396.00 CSR      @ER01  @ER
0397.00 CSR      @ER01  @ER
0398.00 CSR      @ER01  @ER
0399.00 CSR      @ER01  @ER
```

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).

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 <i>Invalid Function Key Pressed</i> 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)
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

INDICATOR	DESCRIPTION
RT	Indicates that a temporary or final halt in the program should take place and returns to the calling program leaving files open

Naming Conventions

Use the following 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

Key List (KLIST)

Define key lists in the housekeeping subroutine.

Begin the key list name with the data file prefix. For example, the Address Book Master file prefix is AB, so the key list would be ABKY01.

The program generator creates key lists using the following naming conventions:

- XXKY01 for physical files where XX = the file prefix. For example: ABKY01
- When a physical 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.
- XXKY0x for logical files where XX is equal to the file prefix and X is equal to the last letter of the logical file name. For example: ABKY0A for F0101LA, ABKY0B for F0101LB
- When a logical 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.

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.

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.

Appendix C - CASE Program Types

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.

Guidelines

The following are optional:

- General help instructions, however JD Edwards World highly recommends you include these.
- Detail (fold) areas and AAIs 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.

A0010 — Interactive Subfile Inquiry

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.

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.

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.

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.

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.

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.

Quick Start Generation

You can generate this program type using the Quick Start CL Generator.

A0020 — Interactive Single Record Inquiry

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.

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 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.

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.

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.

Special Considerations

This program type uses a key list for record retrieval from the master file. This key list should match your key field definition from the control format of the display file. One record displays per inquiry.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

B0010 — Interactive Single Record Maintenance

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.

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.

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.

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.

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.

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C0010 — Batch Report with Totals

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.

Printer File Definition

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

CL Program Definition

Copy and revise 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.

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.

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.

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C0020 — Batch Report with Totals and Subheadings

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.

Printer File Definition

This program type requires that formats HEADING1, HEADING2 and DETAIL1 exist in the printer file. Format TOTAL1 is optional for totals.

CL Program Definition

Copy and revise 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.

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.

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.

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

C0025 — Batch Report with Totals and Subheadings

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.

Printer File Definition

This program type requires that formats HEADING1, HEADING2 and DETAIL1 exist in the printer file. Format HEADING2 is the format that prints subheadings. Format TOTAL1 is optional for totals.

CL Program Definition

Copy and revise 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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0010 — Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number

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.

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.

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.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0020 — Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number

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.

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.

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.

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.

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.

Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0030 — Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record

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.

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.

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.

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.

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.

Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0040 — Interactive Subfile Maintenance with Action Code, with Options, by Key

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.

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.

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.

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.

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.

Special Considerations

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

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

D0050 — Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number

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.

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.

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.

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.

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.

Special Considerations

This program type uses the key information in the subfile control format of the display file for retrieving one record from the first master file and multiple records from the second master file. This type must also have a hidden field and an entry optional field.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key

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.

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.

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.

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.

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.

Special Considerations

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

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

D0070 — Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number

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.

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.

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.

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.

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.

Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0080 — Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number

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.

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.

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.

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.

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.

Special Considerations

This program type uses the key information in the display file for positioning within the master file. This type must also have a hidden relative record number field and an entry optional field.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0090 — Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance

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.

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.

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.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

D0100 — Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key

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.

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.

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.

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.

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.

Special Considerations

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

Quick Start Generation

Generate this program type using the Quick Start CL Generator.

E0010 — Interactive Window

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.

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.

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.

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.

Define Option and Function Key Exits

If you use this window to retrieve values to the calling program, add #SELFC to the Function Exit definitions.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

X0010 — Batch Update with Report

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.

Printer File Definition

This program type requires that formats HEADING1 and DETAIL1 exist in the printer file. Format TOTAL1 is optional exist for totals.

CL Program Definition

Copy and revise 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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

X0020 — Batch Update

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.

Printer File Definition

You do not use a printer file with this program type.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

X0030 — Batch Update with Subroutine S001

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.

Printer File Definition

You do not use a printer file with this program type.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

X0040 - Batch Update with Report

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.

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.

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.

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.

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.

Quick Start Generation

You can not generate this program type using the Quick Start CL Generator.

Y0010 — Conversion, Two Files with Error Report

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.

Printer File Definition

This program type requires that formats HEAD1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

Y0020 — Conversion, One File Update with Error Report

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.

Printer File Definition

This program type requires that formats HEADING1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

CL Program Definition

Copy and revise 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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

Y0030 — Conversion, One File Write with Error Report

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.

Printer File Definition

This program type requires that formats HEAD1, DETAIL1, and ERROR1 exist in the printer file. Format TOTAL1 is optional for totals.

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.

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.

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.

Quick Start Generation

You cannot generate this program type using the Quick Start CL Generator.

Appendix D - Source Listings

This appendix includes the following sources:

- Program Status Data Structure – I00DSPROG
- Copy Module - Retrieve Soft Coding – I00SC
- Item Master Information – P928011

Program Status Data Structure — I00DSPROG

Seq No.		Mod Date
1.00	I*****	00003 08.02.85
2.00	I*	08.02.85
3.00	I* PROGRAM STATUS DATA STRUCTURE	08.02.85
4.00	I* -----	08.02.85
5.00	I*	08.02.85
6.00	I* Portions of this data structure are loaded at the time the	08.02.85
7.00	I* program is loaded. Other portions of this data structure	08.02.85
8.00	I* are loaded as you perform I/O.	08.02.85
9.00	I*	08.02.85
10.00	I* PURPOSE	08.02.85
11.00	I* -----	08.02.85
12.00	I* This common subroutine is set up to be used with C0000	08.02.85
13.00	I* (Business Unit Security) common subroutine and C0001(Edit	08.02.85
14.00	I* Action Code) common subroutine. Those two subroutines	08.02.85
15.00	I* will retrieve ##USER for the user name.	08.02.85
16.00	I*	08.02.85
17.00	I* No program calcs are done in this subroutine.	08.02.85
18.00	I*	08.02.85
19.00	I##PSDS SDS	18.12.89
20.00	I*	08.02.85
21.00	I* Program Name	08.02.85
22.00	I 1 10 ##PROG	08.02.85
23.00	I* Status Code(09999=I/O Error)	08.02.85
24.00	I 11 150##STAT	08.02.85
25.00	I* Previous Status code	08.02.85
26.00	I 16 200##PSTA	08.02.85
27.00	I* RPG Source Statement Sequence Number	08.02.85
28.00	I 21 28 ##SEQN	08.02.85
29.00	I* RPG Routine in Which Exception/Error Occurred	08.02.85
30.00	I 29 36 ##ROUT	08.02.85
31.00	I* Number of Parameters Passed to This Program	08.02.85
32.00	I 37 390##PARM	08.02.85
33.00	I* Exception Type(MCH=Machine, CPF=CPF)	08.02.85
34.00	I 40 42 ##ETYP	08.02.85
35.00	I* Exception Message Number	08.02.85
36.00	I 43 46 ##ENBR	08.02.85
37.00	I* Machine Instruction/Object Definition Template Number	08.02.85
38.00	I 47 50 ##MINO	08.02.85
39.00	I* Work Area for Messages	08.02.85
40.00	I 51 80 ##MWRK	08.02.85
41.00	I* Name of Library in Which Program is Located	08.02.85
42.00	I 81 90 ##PLIB	08.02.85
43.00	I* Retrieved Exdeption Data. CPF Messages.	08.02.85
44.00	I 91 170 ##MSG	08.02.85
45.00	I* Identification of Exception That Caused RPG9001	08.02.85
46.00	I 171 174 ##9001	08.02.85
47.00	I* Unused	08.02.85
48.00	I 175 200 ##FLR1	08.02.85
49.00	I* Name of File for Last I/O(Only Updated if Error)	08.02.85
50.00	I 201 208 ##LFIL	08.02.85
51.00	I* Status Info on Last File Used(Only on Error)	08.02.85
52.00	I 209 243 ##LFST	08.02.85
53.00	I* Status Code on Last File Used(Only on Error)	15.12.89
54.00	I 209 213 ##LFS5	15.12.89
55.00	I* Job Name	08.02.85
56.00	I 244 253 ##JOBN	08.02.85

Seq No.		Print Source Code	Date - 27.01.17
98330		JD Edwards World	
I00DSPROG	.JDFSRC61		
			Mod Date
57.00	I*	User Name From User Profile	08.02.85
58.00	I	254 263 ##USER	08.02.85
59.00	I*	Job Number	08.02.85
60.00	I	264 2690##JOB#	08.02.85
61.00	I*	Date Job Entered the System(MMDDYY)	08.02.85
62.00	I	270 2750##JDT	08.02.85
63.00	I*	Date of Program Execution(MMDDYY)	08.02.85
64.00	I	276 2810##EDT	08.02.85
65.00	I*	Time of Program Exeception(HHMMSS)	08.02.85
66.00	I	282 2870##ETM	08.02.85
67.00	I*	Date Program Was Compiled	08.02.85
68.00	I	288 2930##CDT	08.02.85
69.00	I*	Time Program Was Compiled	08.02.85
70.00	I	294 2990##CTM	08.02.85
71.00	I*	Level of the Compiler	08.02.85
72.00	I	300 303 ##LVL	08.02.85
73.00	I*	Source File Name	08.02.85
74.00	I	304 313 ##SRCN	08.02.85
75.00	I*	Source Library Name	08.02.85
76.00	I	314 323 ##SRCL	08.02.85
77.00	I*	Source File Member Name	08.02.85
78.00	I	324 333 ##SRCL	08.02.85
79.00	I*	Unused	08.02.85
80.00	I	334 429 ##FLR2	09.06.87

Copy Module - Retrieve Soft Coding Data Structure — I00SC

Seq No.		Print Source Code	Date - 27.01.17
98330		JD Edwards World	
I00SC	.JDFSRC61		
			Mod Date
1.00	I*****		12.02.88
2.00	I*	This is part of a composite common subroutine. In	12.02.88
3.00	I*	order for the subroutine to work correctly, the	12.02.88
4.00	I*	RPG program must /COPY in the following members:	12.02.88
5.00	I*	I00SC, C00SC	12.02.88
6.00	I*		25.04.88
7.00	I*	NOTE: The "SRVFDS" file information data structure must	25.04.88
8.00	I*	be specified in a continuation record for the display	25.04.88
9.00	I*	file (File Description Specification "KINFDS").	25.04.88
10.00	I*		25.04.88
11.00	I*****		12.02.88
12.00	I*	PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES	12.02.88
13.00	I*	-----	12.02.88
14.00	I*		12.02.88
15.00	I	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	25.04.88
22.00	I	1 1 #PEOJ	17.02.88
23.00	I*		25.04.88
24.00	I*	Function - Clear Screen	25.04.88
25.00	I	2 2 #FCLR	17.02.88
26.00	I*		25.04.88
27.00	I*	Function - Help	25.04.88
28.00	I	3 3 #FHELP	17.02.88
29.00	I*		25.04.88
30.00	I*	Function - Values List Display	25.04.88
31.00	I	4 4 #FVLST	17.02.88
32.00	I*		25.04.88
33.00	I*	Function - Roll Up	25.04.88
34.00	I	5 5 #FROLU	17.02.88
35.00	I*		25.04.88
36.00	I*	Function - Roll Down	25.04.88
37.00	I	6 6 #FROLD	17.02.88
38.00	I*		25.04.88
39.00	I*	Function - Window Screen Left	25.04.88
40.00	I	7 7 #FWLFT	17.02.88
41.00	I*		25.04.88
42.00	I*	Function - Window Screen Right	25.04.88
43.00	I	8 8 #FWRGT	17.02.88
44.00	I*		25.04.88
45.00	I*	Function - Question Mark/Cursor Sensitive Help	25.04.88
46.00	I	9 9 #FQMRK	17.02.88
47.00	I*		25.04.88
48.00	I*	Function - Display Error Message(s)	25.04.88
49.00	I	10 10 #FERRD	17.02.88
50.00	I*		25.04.88
51.00	I*	Function - Exit to Address Book	25.04.88
52.00	I	11 11 #FAB	17.02.88
53.00	I*		25.04.88

54.00	I*	Function - Exit to Name Search			25.04.88
55.00	I		12	12 #FNS	17.02.88
56.00	I*				25.04.88
98330		JD Edwards World			
I00SC	.JDFSRC61	Print Source Code			Date - 27.01.17
Seq No.					Mod Date
57.00	I*	Function - Return to Previous Panel/Menu			25.04.88
58.00	I		13	13 #FPRV	17.02.88
59.00	I*				25.04.88
60.00	I*	Function - Display Alternate Panel			25.04.88
61.00	I		14	14 #FALT	17.02.88
62.00	I*				25.04.88
63.00	I*	Function - Exit to Display Valid Function Keys			19.09.89
64.00	I		15	15 #FKKEYS	19.09.89
65.00	I*				25.04.88
66.00	I*	Function - Return to Primary Menu			25.04.88
67.00	I		16	16 #FMM	17.02.88
68.00	I*				25.04.88
69.00	I*	Function - Hard Copy Print			25.04.88
70.00	I		17	17 #FPRT	21.04.88
71.00	I*				25.04.88
72.00	I*	Function - Variable by Program (1 thru 15)			25.04.88
73.00	I		18	18 #F01	21.04.88
74.00	I		19	19 #F02	21.04.88
75.00	I		20	20 #F03	21.04.88
76.00	I		21	21 #F04	21.04.88
77.00	I		22	22 #F05	21.04.88
78.00	I		23	23 #F06	21.04.88
79.00	I		24	24 #F07	21.04.88
80.00	I		25	25 #F08	21.04.88
81.00	I		26	26 #F09	21.04.88
82.00	I		27	27 #F10	21.04.88
83.00	I		28	28 #F11	21.04.88
84.00	I		29	29 #F12	21.04.88
85.00	I		30	30 #F13	21.04.88
86.00	I		31	31 #F14	21.04.88
87.00	I		32	32 #F15	21.04.88
88.00	I*				17.02.88
89.00	I*	Selections 1 thru 24.			17.02.88
90.00	I*				17.02.88
91.00	I		33	80 I00SCS	17.02.88
92.00	I*				25.04.88
93.00	I*	Selection - Select/Work With			25.04.88
94.00	I		33	340#SSELC	07.06.88
95.00	I*				25.04.88
96.00	I*	Selection - Change/Revise			25.04.88
97.00	I		35	360#SCHNG	07.06.88
98.00	I*				25.04.88
99.00	I*	Selection - Copy/Hold			25.04.88
100.00	I		37	380#SCOPY	07.06.88
101.00	I*				25.04.88
102.00	I*	Selection - Delete/Cancel			25.04.88
103.00	I		39	400#SDELT	07.06.88
104.00	I*				25.04.88
105.00	I*	Selection - Display/View			25.04.88
106.00	I		41	420#SDSPL	07.06.88
107.00	I*				25.04.88
108.00	I*	Selection - Print/Release			25.04.88
109.00	I		43	440#SPRNT	07.06.88
110.00	I*				25.04.88
111.00	I*	Selection - Rename			25.04.88
112.00	I		45	460#SRENM	07.06.88
98330		JD Edwards World			
I00SC	.JDFSRC61	Print Source Code			Date - 27.01.17
Seq No.					Mod Date
113.00	I*				25.04.88
114.00	I*	Selection - Display Attributes			25.04.88
115.00	I		47	480#SDATR	07.06.88
116.00	I*				25.04.88
117.00	I*	Selection - Variable by Program (1 thru 16)			25.04.88
118.00	I		49	500#S01	07.06.88
119.00	I		51	520#S02	07.06.88
120.00	I		53	540#S03	07.06.88
121.00	I		55	560#S04	07.06.88
122.00	I		57	580#S05	07.06.88
123.00	I		59	600#S06	07.06.88
124.00	I		61	620#S07	07.06.88
125.00	I		63	640#S08	07.06.88
126.00	I		65	660#S09	07.06.88
127.00	I		67	680#S10	07.06.88
128.00	I		69	700#S11	07.06.88
129.00	I		71	720#S12	07.06.88
130.00	I		73	740#S13	07.06.88
131.00	I		75	760#S14	07.06.88
132.00	I		77	780#S15	07.06.88
133.00	I		79	800#S16	07.06.88
134.00	I*				22.02.88
135.00	I*	Global JD Edwards World Variables			07.01.91
136.00	I*				22.02.88
137.00	I		81	120 I00SCG	07.01.91
138.00	I*	Future use space, room to grow			25.02.91

Appendix D - Source Listings

139.00	I*	-----	07.01.91
140.00	I*		07.01.91
141.00	I*	File Information Data Structure for Panel/Report file.	07.01.91
142.00	I*		07.01.91
143.00	ISRVFDS	DS	22.02.88
144.00	I*		22.02.88
145.00	I*	Internal program file name	22.02.88
146.00	I	1 8 @@IFIL	22.02.88
147.00	I*		22.02.88
148.00	I*	Open indication (1=OPEN)	22.02.88
149.00	I	9 9 @@OPEN	22.02.88
150.00	I*		22.02.88
151.00	I*	End Of File indication (1=End of file)	22.02.88
152.00	I	10 10 @@EOF	22.02.88
153.00	I*		22.02.88
154.00	I*	Status code (09999=I/O Error)	22.02.88
155.00	I	11 150@@STAT	22.02.88
156.00	I*		22.02.88
157.00	I*	Operation code	22.02.88
158.00	I	16 21 @@OPCD	22.02.88
159.00	I*		22.02.88
160.00	I*	Name of RPG routine exception/error occurred	22.02.88
161.00	I	22 29 @@ROUT	22.02.88
162.00	I*		22.02.88
163.00	I*	RPG source statement sequence number	22.02.88
164.00	I	30 37 @@SEQN	22.02.88
165.00	I*		22.02.88
166.00	I*	User-Specified reason for error on *SPECIAL file	22.02.88
167.00	I	38 420@@RESN	22.02.88
168.00	I*		22.02.88
98330		JD Edwards World	
I00SC	.JDFSRC61	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 @@FRMT	22.02.88
172.00	I*		22.02.88
173.00	I*	Machine OR CPF message number	22.02.88
174.00	I	46 52 @@EXNO	22.02.88
175.00	I*		22.02.88
176.00	I*	Machine instruction/Object definition template number	22.02.88
177.00	I	53 56 @@MI	22.02.88
178.00	I*		22.02.88
179.00	I*	UNUSED	22.02.88
180.00	I	57 80 @@FLR1	22.02.88
181.00	I*		22.02.88
182.00	I*	Open data path type (DS-Device DB-Data Base SP-Spool)	22.02.88
183.00	I	81 82 @@ODP	22.02.88
184.00	I*		22.02.88
185.00	I*	Name of file actually opened	22.02.88
186.00	I	83 92 @@FILE	22.02.88
187.00	I*		22.02.88
188.00	I*	Name of library containing file (Blank if spool file)	22.02.88
189.00	I	93 102 @@LIBR	22.02.88
190.00	I*		22.02.88
191.00	I*	Name of spooled file (set only on spool files)	22.02.88
192.00	I	103 112 @@SPNM	22.02.88
193.00	I*		22.02.88
194.00	I*	Name of library where spooled file is located	22.02.88
195.00	I	113 122 @@SPLB	22.02.88
196.00	I*		22.02.88
197.00	I*	Spooled file number (set only on spool files)	22.02.88
198.00	I	B 123 1240@@SPNO	22.02.88
199.00	I*		22.02.88
200.00	I*	Primary record length (bytes transferred at a time)	22.02.88
201.00	I	B 125 1260@@PRCL	22.02.88
202.00	I*		22.02.88
203.00	I*	Secondary record length (bytes transferred at a time)	22.02.88
204.00	I	B 127 1280@@SRCL	22.02.88
205.00	I*		22.02.88
206.00	I*	Member Name:	22.02.88
207.00	I*	. If ODP type is DB, this entry is the	22.02.88
208.00	I*	member name in file named in position	22.02.88
209.00	I*	83 through 92.	22.02.88
210.00	I*	. If ODP type is SP, this entry is the	22.02.88
211.00	I*	member name in the file named in	22.02.88
212.00	I*	positions 103 through 112.	22.02.88
213.00	I*		22.02.88
214.00	I	129 138 @@MBR	22.02.88
215.00	I*		22.02.88
216.00	I*	Input buffer length (zero if no buffer allocated)	22.02.88
217.00	I	B 139 1420@@IBLN	22.02.88
218.00	I*		22.02.88
219.00	I*	Output buffer length (zero if no buffer allocated)	22.02.88
220.00	I	B 143 1460@@OBLN	22.02.88
221.00	I*		22.02.88
222.00	I*	Device Class (supplied only if ODP type is DS or SP)	22.02.88
223.00	I*	1 = Display	22.02.88
224.00	I*	2 = Printer	22.02.88
98330		JD Edwards World	
I00SC	.JDFSRC61	Print Source Code	Date - 27.01.17
Seq No.			Mod Date

225.00	I*	3 = Card		22.02.88
226.00	I*	4 = Diskette		22.02.88
227.00	I*	5 = Tape		22.02.88
228.00	I		B 147 1480@@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 @@DKLC	22.02.88
232.00	I*			22.02.88
233.00	I*	Number of rows on display screen or lines on a page		22.02.88
234.00	I		B 152 1530@@VDRW	22.02.88
235.00	I*			22.02.88
236.00	I*	Number of columns on display screen or printed line		22.02.88
237.00	I		B 154 1550@@VDCM	22.02.88
238.00	I*			22.02.88
239.00	I*	Number of records in file at time of open		22.02.88
240.00	I		B 156 1590@@RCNT	22.02.88
241.00	I*			22.02.88
242.00	I*	Access type (only supplied if ODP type is DB)		22.02.88
243.00	I*	KU = Keyed, Unique		22.02.88
244.00	I*	KF = Keyed, FIFO W/Duplicate keys		22.02.88
245.00	I*	KI = Keyed, LIFO W/Duplicate keys		22.02.88
246.00	I*	AR = Arrival sequence		22.02.88
247.00	I		160 161 @@ACTY	22.02.88
248.00	I*			22.02.88
249.00	I*	Duplicate key indication (D=Allowed U=Not allowed)		22.02.88
250.00	I		162 162 @@DUPK	22.02.88
251.00	I*			22.02.88
252.00	I*	Source file indication (Y=Source file)		22.02.88
253.00	I		163 163 @@SRCI	22.02.88
254.00	I*			22.02.88
255.00	I*	User file control block parameters in effect		22.02.88
256.00	I		164 173 @@FCBP	22.02.88
257.00	I*			22.02.88
258.00	I*	User file control block overrides in effect		22.02.88
259.00	I		174 183 @@FCBO	22.02.88
260.00	I*			22.02.88
261.00	I*	Offset to volume label fields of open feedback		22.02.88
262.00	I*	(Supplied only for tape or diskette)		22.02.88
263.00	I		B 184 1850@@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 1870@@RTFO	22.02.88
267.00	I*			22.02.88
268.00	I*	Overflow line number (printer files only)		22.02.88
269.00	I		B 188 1890@@OFLN	22.02.88
270.00	I*			22.02.88
271.00	I*	UNUSED		22.02.88
272.00	I		190 240 @@FLR2	22.02.88
273.00	I*			22.02.88
274.00	I*	Offset to device dependent feedback information		22.02.88
275.00	I*	(See Appendix D of the CPF Programmer's Guide for		22.02.88
276.00	I*	layout of feedback information for specific		22.02.88
277.00	I*	devices)		22.02.88
278.00	I		B 241 2420@@ODFB	22.02.88
279.00	I*			22.02.88
280.00	I*	Put operation count		22.02.88
98330		JD Edwards World		
I00SC	.JDFSRC61	Print Source Code		Date - 27.01.17
Seq No.				Mod Date
281.00	I		B 243 2460@@PUTC	22.02.88
282.00	I*			22.02.88
283.00	I*	Get operation count		22.02.88
284.00	I		B 247 2500@@GETC	22.02.88
285.00	I*			22.02.88
286.00	I*	PutGet operation count		22.02.88
287.00	I		B 251 2540@@PGC	22.02.88
288.00	I*			22.02.88
289.00	I*	Non-I/O operation count (update of subfile records)		22.02.88
290.00	I		B 255 2580@@NIOC	22.02.88
291.00	I*			22.02.88
292.00	I*	Current operation (Last operation requested)		22.02.88
293.00	I*	X'01' = Get		22.02.88
294.00	I*	X'02' = Get W/Subfile record number		22.02.88
295.00	I*	X'03' = Get by key		22.02.88
296.00	I*	X'05' = Put		22.02.88
297.00	I*	X'06' = PutGet		22.02.88
298.00	I*	X'07' = Update		22.02.88
299.00	I*	X'08' = Delete		22.02.88
300.00	I*	X'09' = Force End of Data		22.02.88
301.00	I*	X'0D' = Release		22.02.88
302.00	I		259 260 @@COPR	22.02.88
303.00	I*			22.02.88
304.00	I*	Name of record format just processed:		22.02.88
305.00	I*	. Specified on the I/O request, or		22.02.88
306.00	I*	. Determined by default processing		22.02.88
307.00	I		261 270 @@CFMT	22.02.88
308.00	I*			22.02.88
309.00	I*	Device Class		22.02.88
310.00	I*	Position 271		22.02.88
311.00	I*	X'00' = Data Base		22.02.88
312.00	I*	X'01' = Keyboard display		22.02.88
313.00	I*	X'02' = Printer		22.02.88

Appendix D - Source Listings

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' = 72M Diskette	22.02.88
328.00	I*	X'07' = 5250 Display station, 1920 characters	22.02.88
329.00	I*	X'08' = Spooled	22.02.88
330.00	I	271 272 @@DCLS	22.02.88
331.00	I*		22.02.88
332.00	I*	Device name (Last completed operation)	22.02.88
333.00	I	273 282 @@DNAM	22.02.88
334.00	I*		22.02.88
335.00	I*	Length of last I/O record processed	22.02.88
336.00	I	B 283 2860@LIOL	22.02.88
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 3680@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'F5' = 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'B1' = Command Key 13	22.02.88
366.00	I*	X'B2' = Command Key 14	22.02.88
367.00	I*	X'B3' = Command Key 15	22.02.88
368.00	I*	X'B4' = Command Key 16	22.02.88
369.00	I*	X'B5' = Command Key 17	22.02.88
370.00	I*	X'B6' = Command Key 18	22.02.88
371.00	I*	X'B7' = Command Key 19	22.02.88
372.00	I*	X'B8' = Command Key 20	22.02.88
373.00	I*	X'B9' = Command Key 21	22.02.88
374.00	I*	X'BA' = Command Key 22	22.02.88
375.00	I*	X'BB' = Command Key 23	22.02.88
376.00	I*	X'BC' = 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 3720@RTRM	22.02.88
394.00	I*		22.02.88
395.00	I*	UNUSED	22.02.88
396.00	I	373 375 @@FLR4	22.02.88
397.00	I*		22.02.88
398.00	I*	RRN of last subfile record written/updated	22.02.88

399.00	I		B 376 3770@SRRN	22.02.88
400.00	I*			22.02.88
401.00	I*		RRN of first subfile record on display	22.02.88
402.00	I		B 378 3790@SRCN	22.02.88
403.00	I*			22.02.88
404.00	I*	UNUSED		22.02.88
405.00	I		380 396 @@FLR5	02.10.89
406.00	I*			22.02.88
407.00	I*		RRN of data base record	22.02.88
408.00	I		B 397 4000@RRN	22.02.88
409.00	I*			22.02.88
410.00	I*		Data base file key	22.02.88
411.00	I		401 528 @@RKEY	22.02.88
412.00	I*		-----	22.02.88
413.00	I*			30.08.89
414.00	I*		Cursor Sensitive Help Values	30.08.89
415.00	I*			30.08.89
416.00	II00CSR	DS		30.08.89
417.00	I*			30.08.89
418.00	I*		Returned field name.	30.08.89
419.00	I		1 10 ##FLDN	30.08.89
420.00	I*		Returned value.	30.08.89
421.00	I		11 40 ##RVAL	29.09.89
422.00	I*		Returned description.	30.08.89
423.00	I		41 70 ##RDSC	29.09.89
424.00	I*		Returned location: Row.	31.08.89
425.00	I		71 730##RR0W	29.09.89
426.00	I*		Returned location: Column.	31.08.89
427.00	I		74 760##RCOL	29.09.89
428.00	I*		Dictionary Field Name (non-blank=override)	03.11.89
429.00	I		77 86 ##DTAI	29.09.89
430.00	I*		Returned Display File Format	29.09.89
431.00	I		87 96 ##RFMT	29.09.89
432.00	I*		RPG Indicator Array	29.09.89
433.00	I		97 195 ##IN	29.09.89
434.00	I*		Override Reporting System (Jargon)	06.10.92
435.00	I		196 199 ###SYR	06.10.92
436.00	I*		-----	30.08.89
437.00	I*			27.11.89
438.00	I*		Hidden Fields for Subfile Attribute Indicators	27.11.89
439.00	I*			27.11.89
440.00	ISHIN	DS		27.11.89
441.00	I		1 1 SHIN01	27.11.89
442.00	I		2 2 SHIN02	27.11.89
443.00	I		3 3 SHIN03	27.11.89
444.00	I		4 4 SHIN04	27.11.89
445.00	I		5 5 SHIN05	27.11.89
446.00	I		6 6 SHIN06	27.11.89
447.00	I		7 7 SHIN07	27.11.89
448.00	I		8 8 SHIN08	27.11.89
98330			JD Edwards World	
I00SC	.JDFSRC61		Print Source Code	Date - 27.01.17
Seq No.				Mod Date
449.00	I		9 9 SHIN09	27.11.89
450.00	I		10 10 SHIN10	27.11.89
451.00	I		11 11 SHIN11	27.11.89
452.00	I		12 12 SHIN12	27.11.89
453.00	I		13 13 SHIN13	27.11.89
454.00	I		14 14 SHIN14	27.11.89
455.00	I		15 15 SHIN15	27.11.89
456.00	I		16 16 SHIN16	27.11.89
457.00	I		17 17 SHIN17	27.11.89
458.00	I		18 18 SHIN18	27.11.89
459.00	I		19 19 SHIN19	27.11.89
460.00	I		20 20 SHIN20	27.11.89
461.00	I		21 21 SHIN21	27.11.89
462.00	I		22 22 SHIN22	27.11.89
463.00	I		23 23 SHIN23	27.11.89
464.00	I		24 24 SHIN24	27.11.89
465.00	I		25 25 SHIN25	27.11.89
466.00	I		26 26 SHIN26	27.11.89
467.00	I		27 27 SHIN27	27.11.89
468.00	I		28 28 SHIN28	27.11.89
469.00	I		29 29 SHIN29	27.11.89
470.00	I		30 30 SHIN30	27.11.89
471.00	I		31 31 SHIN31	27.11.89
472.00	I		32 32 SHIN32	27.11.89
473.00	I		33 33 SHIN33	27.11.89
474.00	I		34 34 SHIN34	27.11.89
475.00	I		35 35 SHIN35	27.11.89
476.00	I		36 36 SHIN36	27.11.89
477.00	I		37 37 SHIN37	27.11.89
478.00	I		38 38 SHIN38	27.11.89
479.00	I		39 39 SHIN39	27.11.89
480.00	I		40 40 SHIN40	30.11.89
481.00	I		41 41 SHIN41	27.11.89
482.00	I		42 42 SHIN42	27.11.89
483.00	I		43 43 SHIN43	27.11.89
484.00	I		44 44 SHIN44	27.11.89
485.00	I		45 45 SHIN45	27.11.89
486.00	I		46 46 SHIN46	27.11.89
487.00	I		47 47 SHIN47	27.11.89
488.00	I		48 48 SHIN48	27.11.89

Appendix D - Source Listings

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	.JDFSRC61				Print Source Code	Date - 27.01.17
Seq No.						Mod Date
505.00	I		65	65	SHIN65	27.11.89
506.00	I		66	66	SHIN66	27.11.89
507.00	I		67	67	SHIN67	27.11.89
508.00	I		68	68	SHIN68	27.11.89
509.00	I		69	69	SHIN69	27.11.89
510.00	I		70	70	SHIN70	27.11.89
511.00	I		71	71	SHIN71	27.11.89
512.00	I		72	72	SHIN72	27.11.89
513.00	I		73	73	SHIN73	27.11.89
514.00	I		74	74	SHIN74	27.11.89
515.00	I		75	75	SHIN75	27.11.89
516.00	I		76	76	SHIN76	27.11.89
517.00	I		77	77	SHIN77	27.11.89
518.00	I		78	78	SHIN78	27.11.89
519.00	I		79	79	SHIN79	27.11.89
520.00	I		80	80	SHIN80	27.11.89
521.00	I		81	81	SHIN81	27.11.89
522.00	I		82	82	SHIN82	27.11.89
523.00	I		83	83	SHIN83	27.11.89
524.00	I		84	84	SHIN84	27.11.89
525.00	I		85	85	SHIN85	27.11.89
526.00	I		86	86	SHIN86	27.11.89
527.00	I		87	87	SHIN87	27.11.89
528.00	I		88	88	SHIN88	27.11.89
529.00	I		89	89	SHIN89	27.11.89
530.00	I		90	90	SHIN90	27.11.89
531.00	I		91	91	SHIN91	27.11.89
532.00	I		92	92	SHIN92	27.11.89
533.00	I		93	93	SHIN93	27.11.89
534.00	I		94	94	SHIN94	27.11.89
535.00	I		95	95	SHIN95	27.11.89
536.00	I		96	96	SHIN96	27.11.89
537.00	I		97	97	SHIN97	27.11.89
538.00	I		98	98	SHIN98	27.11.89
539.00	I		99	99	SHIN99	27.11.89
540.00	I*				-----	27.11.89
541.00	I*					09.06.93
542.00	I*	Hidden Fields for Subfile Mode and Cursor Position				09.06.93
543.00	I*					09.06.93
544.00	II00MDE	DS				09.06.93
545.00	I*	Subfile Mode				09.06.93
546.00	I		1	1	####MD	09.06.93
547.00	I*	Subfile Relative Record Number				09.06.93
548.00	I		2	60	####RNO	09.06.93
549.00	I*	Cursor Location - Record Format				09.06.93
550.00	I		7	16	###CRC	09.06.93
551.00	I*	Cursor Location - Field Name				09.06.93
552.00	I		17	26	###CFL	09.06.93
553.00	I*				-----	09.06.93

Item Master Information — P928011

```

1.00 H/TITLE P928011-Item Master Information
2.00 H*-----
3.00 H*
4.00 H* Copyright (c) 1993
5.00 H* J. D. Edwards & Company
6.00 H*
7.00 H*
8.00 H*
9.00 H*
10.00 H*
11.00 H*
12.00 H*
13.00 H*
14.00 H*
15.00 H*-----
16.00 F*
17.00 F* PROGRAM REVISION LOG
18.00 F* -----
19.00 F*
20.00 F*
21.00 F*
22.00 AUTHRP*
23.00 F*
24.00 F* B0010 - Standard Maintenance Program Type
25.00 F* This program provides the standard single cycle
26.00 F* processing for adding, changing, deleting and
27.00 F* inquiring into data records as requested.
28.00 F*
29.00 F*-----
30.00 F*
31.00 FP001 IF E K DISK
32.00 FP92801 UP E K DISK
33.00 PV928011 CP E WORKSTN KINFDS SRVENDS A
34.00 F*-----
35.00 F*
36.00 F* Copy Member for Composite Common Subroutine - C0001
37.00 F*
38.00 F*/COPY JDECPY,D0001
39.00 F*-----
40.00 E*-----
41.00 E* PROGRAM TABLES AND ARRAYS
42.00 E* -----
43.00 E*
44.00 E
45.00 E
46.00 E
47.00 E
48.00 E
49.00 E*
50.00 E*
51.00 E* Copy Member for Composite Common Subroutine - C0001
52.00 E*
53.00 E*/COPY JDECPY,E0001
54.00 E*-----
55.00 E*
56.00 E* Copy Member for Composite Common Subroutine - C0012
57.00 E*
58.00 E/COPY JDECPY,E0012
59.00 E*-----
60.00 E* Copy Member for Composite Common Subroutine - C997
61.00 E*
62.00 E*
63.00 E/COPY JDECPY,E997
64.00 I*-----
65.00 I*-----
66.00 I* PROGRAM INPUT SPECIFICATIONS AND DATA STRUCTURES
67.00 I* -----
68.00 I*

```

This unpublished material is proprietary to J. D. Edward & Company. All rights reserved. The methods and techniques described herein are considered trade secrets and/or confidential. Reproduction or distribution, in whole or in part, is forbidden except by express written permission of J. D. Edwards & Company.

Copyright statement can be changed through the Program Generator

Date	Programmer	Nature of Revision
12/07/93	Quarles	SAR # 241883 (AS/400 A/G)

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.

File Name	Device	Access	File Type	File Name	Device	Access	File Type
FP001	IF	E	K	DISK			
FP92801	UP	E	K	DISK			
PV928011	CP	E		WORKSTN	KINFDS	SRVENDS	A

Informational data structure for the video

Copy Member for Composite Common Subroutine - C0001
F/COPY JDECPY,D0001

Symbol	Count	Usage	Description
EMK	64	4	Error Msg
@MK	64	1	Error Msg
@ER	64	4	Error Msg
@DV	40	1	Dflt Wrk
@C	256	1	Literal Work

Arrays that handle error messages

Copy Member for Composite Common Subroutine - C0001
E*/COPY JDECPY,E0001

Will copy in additional specifications for copy module C0001

```

69.00 I*   Data Structures to Load Video Screen Text
70.00 I*
71.00 IDSTXT      DS              1000
72.00 I          1          18 VTX001
73.00 I*         41          58 VTX002
74.00 I*         81          92 VTX003
75.00 I*        121         138 VTX004
76.00 I*        161         178 VTX005
77.00 I*        201         218 VTX006
78.00 I*        241         258 VTX007
79.00 I*        281         298 VTX008
80.00 I*        321         338 VTX009
81.00 I*        361         378 VTX010
82.00 I*        401         418 VTX011
83.00 I*        441         458 VTX012
84.00 I*        481         498 VTX013
85.00 I*        521         536 VTX014
86.00 I*        561         576 VTX015
87.00 I*        601         616 VTX016
88.00 I*        641         656 VTX017
89.00 I*        681         696 VTX018
90.00 I*        721         736 VTX019
91.00 I*        761         776 VTX020
92.00 I*        801         816 VTX021
93.00 I*        841         856 VTX022
94.00 I*        881         896 VTX023
95.00 I*        921         936 VTX024
96.00 I*        961         976 VTX025
97.00 I*
98.00 I/COPY JDECFY, IOODSINX  — Data structure for commonly used indexes
99.00 I/COPY JDECFY, IOOPS@@  — Data structure used with file servers
100.00 I/COPY JDECFY, IOODSPROG — Program status data structure
101.00 I*
102.00 I*
103.00 I*
104.00 I*   Copy Member for Composite Common Subroutine - COOSC
105.00 I*
106.00 I/COPY JDECFY, IOOSC  — 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 JDECFY, IO005U  — 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 JDECFY, IO00661
117.00 I*****
118.00 I*
119.00 I*   Copy Member For Server - x9800E
120.00 I*
121.00 I/COPY JDECFY, 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*
130.00 C*   EXSR S999 — One time only functions
131.00 C*   -----
132.00 C*   If LR on, end program.
133.00 C*
134.00 C*   *INLR   CABEQ'1'   EOJ
135.00 C*   -----   ---
136.00 C*
137.00 C*   If automatic inquiry set, process inquiry.
138.00 C*
139.00 C*   $AUTO   CASEQ'1'   S003
140.00 C*   -----   ---
141.00 C*   End
142.00 C*
143.00 C*   Begin normal program processing.
144.00 C*   -----
145.00 C*
146.00 C*   *INLR   DOWEQ'0'
147.00 C*
148.00 C*   Write video screen.
149.00 C*

```

Each VTX field is 40 long but may not use all 40. Pulls in text from Vocabulary Overrides.

24 — If information is passed to this program, it will automatically inquire on the record


```

150.00 C WRITEV9280111
151.00 C MOVE /1/ @@AID
152.00 C EXSR S001
153.00 C* ----- Clears fields
154.00 C*
155.00 C* Load data field dictionary parameters (one cycle only).
156.00 C*
157.00 C $998 CASEQ' ' S998
158.00 C* ----- One time only. Pulls in Data
159.00 C END Dictionary editing information
160.00 C* functions
161.00 C* Begin video screen read processing.
162.00 C*
163.00 C SETOF 999301
164.00 C READ V928011 9998
165.00 C Z-ADDO ##RR0W
166.00 C Z-ADDO ##RCOL Used for cursor sensitive help.
167.00 C* Tells where the cursor is.
168.00 C* If video read timed out, end program.
169.00 C*
170.00 C *IN99 CABEQ'1' EOJ LR
171.00 C* -----
172.00 C @@AID CABEQ#FEOJ EOJ LR
173.00 C* -----
174.00 C*
175.00 C* If valid function key pressed, process and return.
176.00 C*
177.00 C *IN15 IPBQ '1' All function keys are assigned indicator 15 so
178.00 C EXSR S00EX if 15 is on, a function key has been pressed
179.00 C* -----
180.00 C INLR CABEQ'1' EOJ
181.00 C* -----
182.00 C *IN15 CABEQ'1' END
183.00 C* -----
184.00 C END
185.00 C*
186.00 C* Edit the action code.
187.00 C*
188.00 C EXSR C0001 Edits the action code.
189.00 C* ----- Checks action code security.
190.00 C*
191.00 C* If end of job requested, end program.
192.00 C*
193.00 C @@AID CABEQ#FEOJ EOJ
194.00 C* -----
195.00 C*
196.00 C* If clear screen requested, process and return.
197.00 C*
198.00 C @@AID IPBQ #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
208.00 C* ----- to load the video/report fields
209.00 C*
210.00 C* If add or change, validate all video input.
211.00 C*
212.00 C *IN93 CASEQ'0' S005 If an error has occurred,
213.00 C* ----- validates and edits data
214.00 C END
215.00 C*
216.00 C* If no errors and not inquiry, update file.
217.00 C*
218.00 C *IN93 IPBQ '0'
219.00 C *IN24 CASEQ'0' S010 Updates files
220.00 C* -----
221.00 C END
222.00 C END
223.00 C*
224.00 C* Return for next input.
225.00 C*
226.00 C END TAG
227.00 C* -----
228.00 C*

```

```

229.00 C*      Set correct message in line 24.
230.00 C*
231.00 C          *IN93      IFEQ '1'
232.00 C                      MOVELSVL24E      VOL24
233.00 C                      ELSE
234.00 C                      MOVELSVL24M      VDL24
235.00 C                      END
236.00 C*
237.00 C          END
238.00 C*
239.00 C          EOJ      TAG
240.00 C*          ---      ---
241.00 C*
242.00 C*      END MAINLINE PROGRAM
243.00 C*
244.00 C*****
245.00 C*
246.00 C*      Copy Common Subroutine - Edit Action Code
247.00 C*
248.00 C/COPY JDECPY, C0001
249.00 C*****
250.00 C*
251.00 C*      SUBROUTINE SOOEX - Process Function Keys
252.00 C*
253.00 C*
254.00 C      Processing: 1. Determine function key pressed.
255.00 C*                2. Process function key request.
256.00 C*
257.00 CSR          SOOEX      BEGSR
258.00 C*          ---      ---
259.00 CSR          TOOEXA      Tag
260.00 C*          ---      ---
261.00 C*
262.00 C*      If EOJ requested, exit subroutine.
263.00 C*
264.00 CSR          @@AID      CABBE#PEOJ      ENDEXE      LR
265.00 C*          ---      ---
266.00 C*
267.00 C*      If Display Keys pressed, exit to help facility and return.
268.00 C*
269.00 C*
270.00 CSR          @@AID      IFEQ #FKEYS
271.00 CSR          CALL 'P9601H'          98
272.00 C*          ---
273.00 CSR          PARM          IOOSC
274.00 CSR          PARM          SRVFDS
275.00 CSR          PARM          IOOCSR
276.00 C*
277.00 CSR          @@AID      CABNE#FKEYS      TOOEXA
278.00 C*          ---
279.00 CSR          GOTO ENDEXE
280.00 C*          ---
281.00 CSR          END
282.00 C*
283.00 C*
284.00 C*      If Cursor Sensitive Help Pressed, exit to CS Help.
285.00 C*
286.00 CSR          @@AID      IFEQ #FQMRK
287.00 CSR          MOVEA*IN      ##IN
288.00 CSR          CALL 'X96CCX'          98
289.00 C*
290.00 CSR          PARM          IGOSC
291.00 CSR          PARM          SRVFDS
292.00 CSR          PARM          IOOCSR
293.00 CSR          PARM ' '          ##CCFP 2
294.00 CSR          PARM          IOGMDE
295.00 C*
296.00 CSR          ##FLDN      IPNE *BLANKS
297.00 CSR          EXSR SOOVL
298.00 C*          ---
299.00 C*          MOVEA##IN      *IN,1
300.00 CSR          END
301.00 CSR          MOVEL*BLANKS      ##DTAI
302.00 CSR          GOTO ENDEXE
303.00 C*          ---
304.00 CSR          END
305.00 C*

```

Sets the message for Line 24

Contains what function key was pressed by the user

External programs start with an X. This is the cursor sensitive help program

Parameters passed identifying where the cursor was when F1 was pressed

```

306.00 C*      If Display errors pressed, exit to error messages
307.00 C*      -----
308.00 C*
309.00 CSR      @@AID      IFEQ #FERRD
310.00 CSR      Z-ADD1      #G
311.00 CSR      Z-ADD1      #H
312.00 CSR      #G      DOWLE64
313.00 CSR      @MK,#G    IFEQ '1'
314.00 CSR      MOVE EMK,#G      @ER,#H
315.00 CSR      Add 1      #H
316.00 CSR      END
317.00 CSR      ADD 1      #G
318.00 CSR      END
319.00 CSR      CALL 'POOOOE'      98
320.00 C*      -----
321.00 CSR      PARM      @ER
322.00 CSR      GOTO ENDEXE
323.00 C*      -----
324.00 CSR      END
325.00 C*
326.00 C*      If HELP key pressed, exit to help facility and return.
327.00 C*      -----
328.00 C*
329.00 C*      @@AID      IFEQ #PHELP
330.00 C*      CALL 'POOHELP'      98 Access JDE Help information
331.00 C*      -----
332.00 CSR      PARM      HS@@
333.00 CSR      PARM      HE@@
334.00 CSR      PARM      IOOSC
335.00 CSR      PARM      SRVPDS
336.00 CSR      GOTO ENDEXE
337.00 C*      -----
338.00 CSR      END
339.00 C*
340.00 C*      If Clear screen pressed, clear screen and return.
341.00 C*      -----
342.00 C*
343.00 CSR      @@AID      IFEQ #FCLR
344.00 CSR      EXSR S001
345.00 C*      -----
346.00 CSR      GOTO ENDEXE
347.00 C*      -----
348.00 CSR      END
349.00 C*
350.00 C*      Process roll up and down keys.
351.00 C*      -----
352.00 C*
353.00 CSR      @@AID      IFEQ #PROLU
354.00 CSR      @@AID      OREQ #FROLD
355.00 C*      $SECUR      DOUEQ' '
356.00 CSR      MOVE ' '      $SECUR 1
357.00 C*
358.00 C*      If ROLL UP key pressed, process read next.
359.00 C*      -----
360.00 C*
361.00 CSR      @@AID      IFEQ #PROLU
362.00 C*
363.00 C*      Reset error indicators if roll
364.00 C*
365.00 CSR      MOVEA$RESET      *IN,41
366.00 CSR      MOVE '0'      *IN,40
367.00 CSR      SETOP      818299
368.00 CSR      READ I92801      9981
369.00 CSR      *IN81      IFEQ '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      IFEQ '1'
377.00 CSR      SETON      9341
378.00 CSR      MOVE '1'      @MK,2
379.00 CSR      GOTO ENDEXE
380.00 C*      -----
381.00 CSR      END
382.00 CSR      END

```

```

384.00   CSR                END
385.00   C*
386.00   C*      If ROLL DOWN key pressed, process read prior.
387.00   C*      -----
388.00   C*
389.00   CSR                @@AID   IFEQ #PROLD
390.00   C*
391.00   C*      Reset error indicators if roll
392.00   C*
393.00   CSR                MOVEA$RRESET      *IN, 41
394.00   CSR                MOVE '0'          *IN, 40
395.00   CSR                SETOP                    818299
396.00   CSR                READPI92801          9981
397.00   CSR                *IN81   IFEQ '1'
398.00   CSR                $RDKEY  SETLLI92801
399.00   CSR                SETOP                    8299
400.00   CSR                READPI92801          9982
401.00   C*
402.00   C*      If error on read, set error.
403.00   C*
404.00   CSR                *IN82   IFEQ '1'
405.00   CSR                SETON                    9341
406.00   CSR                MOVE '1'          @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   IFEQ #PROLU
417.00   CSR                @@AID   OREQ #PROLD
418.00   C*
419.00   C*      Release record lock or report record in use.
420.00   C*
421.00   CSR                *IN99   IFEQ '0'
422.00   CSR                EXCPTUNLOCK
423.00   CSR                ELSE
424.00   CSR                CALL 'P98BLCK'          81
425.00   C*      -----
426.00   CSR                PARM                ##PSDS
427.00   CSR                SETON                    9341
428.00   CSR                MOVE '1'          @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'LQXXCC      '#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  CRSEQ' '      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'
460.00   CSR                SETON                    0193
461.00   CSR                GOTO ENDEXE
462.00   C*      -----
463.00   CSR                END
464.00   C*
465.00   CSR                ENDEXE  ENDSR

```

Program that will display a record lock window when a record in use error is encountered

Could not find a match in the Function Key Definitions for the function key pressed, so program displays *Invalid Function Key* message.

```

466.00 C*****
467.00 C*
469.00 C*   Copy Common Subroutine - Coat Center Security Check
469.00 C*
470.00 C/COPY 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 update and move in the
477.00 C*   returned value.  If the format is a subfile, the record
478.00 C*   to change is found in @@RRN.
479.00 C*
480.00 CSR           S00VL   BEGSR
481.00 C*           -----
482.00 C*
483.00 CSR           ##RVAL   IPEQ 'BLANK'
484.00 CS           MOVE *BLANK   ##RVAL
485.00 C*
486.00 C*           END
487.00 C*   Return values for fields in format V9280111
488.00 C*
489.00 CSR           ##RFMT   IPEQ 'V9280111'
490.00 C*
491.00 CSR           ##PLDN   IPEQ 'ACTION'
492.00 CSR           MOVE##RVAL   ACTION
493.00 CSR           GOTO ENDOVL
494.00 C*           -----
495.00 CSR           END
496.00 C*
497.00 CSR           ##PLDN   IPEQ 'VDXIT'
498.00 CSR           MOVE##RVAL   VDXIT
499.00 CSR           GOTO ENDOVL
500.00 C*           -----
501.00 CSR           END
502.00 C*
503.00 CSR           ##PLDN   IPEQ 'VDXDS'
504.00 CSR           MOVE##RVAL   VDXDS
505.00 CSR           GOTO ENDOVL
506.00 C*           -----
507.00 CSR           END
508.00 C*
509.00 CSR           ##PLDN   IPEQ 'VDXCC'
510.00 CSR           MOVE##RVAL   VDXCC
511.00 CSR           GOTO ENDOVL
512.00 C*           -----
513.00 CSR           END
514.00 C*
515.00 CSR           ##PLDN   IPEQ 'VDXTY'
516.00 CSR           MOVE##RVAL   VDXTY
517.00 CSR           GOTO ENDOVL
518.00 C*           -----
519.00 CSR           END
520.00 C*
521.00 CSR           ##PLDN   IPEQ 'VDXDT'
522.00 CSR           MOVE##RVAL   VDXDT
523.00 CSR           GOTO ENDOVL
524.00 C*           -----
525.00 CSR           END
526.00 C*
527.00 CSR           ##PLDN   IPEQ 'VDXQT'
528.00 CSR           MOVE##RVAL   VDXQT
529.00 CSR           GOTO ENDOVL
530.00 C*           -----
531.00 CSR           END
532.00 C*
533.00 CSR           ##PLDN   IPEQ 'VDXUM'
534.00 CSR           MOVE##RVAL   VDXUM
535.00 CSR           GOTO ENDOVL
536.00 C*           -----
537.00 CSR           END
538.00 C*
539.00 CSR           ##PLDN   IPEQ 'VDX001'
540.00 CSR           MOVE##RVAL   VDX001
541.00 CSR           GOTO ENDOVL
542.00 C*           -----

```

For cursor sensitive help.
Information was retrieved in
program X96CCX. The retrieved
information is returned to the
video fields in this subroutine.

```

543.00 CSR          END
544.00 C*
545.00 CSR          ##FLDN  IPFQ 'VDX002 '
546.00 CSR          MOVEL##RVAL  VDX002
547.00 CSR          GOTO ENDOVL
548.00 C*          -----
549.00 CSR          END
550.00 C*
551.00 CSR          ##FLDN  IPFQ 'VDX003 '
552.00 CSR          MOVEL##RVAL  VDX003
553.00 CSR          GOTO ENDOVVL
554.00 C*          -----
555.00 CSR          END
556.00 C*
557.00 CSR          ##FLDN  IPFQ 'VDX004 '
558.00 CSR          MOVEL##RVAL  VDX004
559.00 CSR          GOTO ENDOVL
561.00 CSR          END
562.00 C*
563.00 CSR          ##FLDN  IPFQ 'VDX005 '
564.00 CSR          MOVEL##RVAL  VDX005
565.00 CSR          GOTO ENDOVL
566.00 C*          -----
567.00 CSR          END
568.00 CSR          END
569.00 C*
570.00 CSR          ENDOVL  ENDSR
571.00 C*****
572.00 C*
573.00 C*          SUBROUTINE S001 - Clear Fields
574.00 C*          -----
575.00 C*
576.00 C*          Processing: 1. Reset all video screen and data file fields
577.00 C*                      for next transaction.
578.00 C*                      2. Clear action code only if requested.
579.00 C*
580.00 CSR          S001  BEGSR
581.00 C*          -----
582.00 C*
583.00 C*          Reset fields for next transaction.
584.00 C*
585.00 CSR          NOKEY  CLEARI92801
586.00 CSR          MOVE *BLANK  ###CLF
587.00 CSR          MOVE *BLANK  ###CRC
588.00 CSR          Z-ADD*ZERO  ###RCOL
589.00 CSR          Z-ADD*ZERO  ###RROW
590.00 CSR          MOVE *BLANK  VDXCC
591.00 CSR          MOVE *BLANK  VDXDS
592.00 CSR          MOVE *BLANK  VDXD3
593.00 CSR          MOVE *BLANK  VDXIT
594.00 CSR          MOVE *BLANK  VDXQT
595.00 CSR          MOVE *BLANK  VDXTY
596.00 CSR          MOVE *BLANK  VDXUM
597.00 CSR          MOVE *BLANK  VDX001
598.00 CSR          MOVE *BLANK  VDX002
599.00 CSR          MOVE *BLANK  VDX003
600.00 CSR          MOVE *BLANK  VDX004
601.00 CSR          MOVE *BLANK  VDX005
602.00 CSR          MOVE*SVL24M  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  IPFQ #FCLR
608.00 CSR          MOVE *ALL' 0' $RESET
609.00 CSR          MOVEA$RESET *IN,41
610.00 CSR          MOVE ' ' ACTION 1
611.00 CSR          Z-ADD*ZERO QXXIT
612.00 CSR          MOVE *BLANK VC0001
613.00 CSR          MOVE *BLANK VC0002
614.00 CSR          MOVE *BLANK VC0003
615.00 CSR          MOVE *BLANK VC0004
616.00 CSR          MOVE *BLANK VC0005
617.00 CSR          MOVE *BLANK VC0006
618.00 CSR          MOVE *BLANK VC0007
619.00 CSR          MOVE *BLANK VC0008
620.00 CSR          Z-ADD*ZERO $$EDT 60
621.00 CSR          END
622.00 C*
623.00 CSR          END001  ENDSR

```

Clears all the fields in the record format for F92801

Clears the video fields

These fields will only be cleared if the user presses the function key to clear the screen. We want to save certain information like key fields and descriptions of they don't get cleared everytime S001 is executed.

```

624.00 C*****
625.00 C*
626.00 C* SUBROUTINE S003 - Edit Key
627.00 C* Sets the file pointer and
628.00 C* edit the key
629.00 C* Processing: 1. Clear error indicators and arrays.
630.00 C* 2. Load input keys.
631.00 C* 3. Validate Master file key.
632.00 C* 4. Release master file record lock.
633.00 C* 5. Load video screen output on inquiry.
634.00 C*
635.00 CSR S003 BEGSR
636.00 C* ----
637.00 C*
638.00 C* Load data field dictionary parameters (one cycle only).
639.00 C*
640.00 CSR $998 CASEQ' ' S998
641.00 C* ----
642.00 CSR END
643.00 C*
644.00 C* Reset error indicators and arrays.
645.00 C*
646.00 CSR MOVE *ALL' 0' $RESET 39
647.00 CSR MOVE *BLANK $REST1 63
648.00 CSR MOVEA$RESET *IN, 41
649.00 CSR MOVEA$REST1 @MK, 2
650.00 CSR CLEAR@ER
651.00 C*-----
652.00 C*
653.00 C* Load video input field for - Item ID
654.00 C*
655.00 CSR MOVEAVDXIT @NM
656.00 CSR EXSR C0012
657.00 C*
658.00 CSR Z-ADD#NUMR $NBR08 80
659.00 CSR MOVE $NBR08 QXXIT
660.00 C*
661.00 C* Automatic Next Number for - Item ID
662.00 C*
663.00 CSR *IN21 IFEQ '1'
664.00 CSR VDXIT ANDEQ*BLANK
665.00 CSR SETON 81
666.00 CSR *IN81 DOWEQ'1'
667.00 CSR MOVE #NXTIT PSIDX 2
668.00 CSR CALL 'X0010' 82
669.00 C*
670.00 CSR PARM S@XIT NNSY 4
671.00 CSR PARM PSIDX
672.00 CSR PARM *ZERO #NXTNO 80
673.00 CSR MOVE #NXTNO QXXIT
674.00 CSR MOVE #NXTTNO VDXIT
675.00 CSR QXXIT SETLLF92801 8281
676.00 CSR END
677.00 CSR END
678.00 C*-----
679.00 CSR QXKY01 CHAIN192801 9899
680.00 C*
681.00 C* Cost Center security edit.
682.00 C*
683.00 CSR MOVE'P92801 ' #FILE
684.00 CSR MOVELQXXCC #MCU
685.00 CSR #AUT IFNE '1'
686.00 CSR #FAUT ANDNE'1'
687.00 CSR EXSR C0000 Checks cost center security
688.00 C*
689.00 CSR END
690.00 CSR #AUT IFNE '1'
691.00 CSR #FAUT ANDNE'1'
692.00 C* #MAUT ANDNE'1'
693.00 CSR MOVE '1' $$SECR 1
694.00 CSR END
695.00 C*
696.00 C* If security violation, set error condition.
697.00 C*
698.00 CSR $$SECR IFEQ '1'
699.00 CSR MOVE '1' @MK,8
700.00 CSR SETON 9341

```

```

701.00  CSR          MOVE ' '          $$$SEFCR  1
702.00  CSR          GOTO END003
703.00  C*          -----
704.00  CSR          END
705.00  C*
706.00  C*          Edit result of read and action code.
707.00  C*
708.00  CSR          *IN98  IFEQ '1'
709.00  CSR          *IN21  COMP '0'          41 *error*
710.00  CSR          ELSE
711.00  CSR          *IN21  COMP '1'          41 *error*
712.00  CSR          END
713.00  C*
714.00  C*          If indicator 41 on, invalid key for action code.
715.00  C*
716.00  CSR          *IN41  IFEQ '1'
717.00  CSR          MOVE '1'          @MK, 2
718.00  CSR          SETON          93
719.00  CSR          END
720.00  C*
721.00  C*          If indicator 99 on, record in use.
722.00  C*
723.00  CSR          *IN99  IFEQ '1'
724.00  CSR          CALL 'P96RLCK'          81
725.00  CSR          -----
726.00  CSR          PARM          ##PSDS
727.00  CSR          MOVE '1'          @MK, 6
728.00  CSR          SETON          9341
729.00  CSR          END
730.00  C*          -----
731.00  C*
732.00  C*          If not inquiry, skip remainder of subroutine.
733.00  C*
734.00  CSR          *IN24  CABEQ '0'          END003
735.00  CSR          -----
736.00  C*          -----
737.00  C*
738.00  C*          Release record lock on master file
739.00  C*
740.00  CSR          *IN98  IFEQ '0'
741.00  CSR          (IN99  ANDEQ '0'
742.00  CSR          EXCPTUNLOCK
743.00  CSR          END
744.00  C*
745.00  CSR          If errors, skip remainder of subroutine.
746.00  C*
747.00  CSR          *IN93  CABEQ '1'          END003
748.00  C*          -----
749.00  C*          -----
750.00  C*
751.00  C*          Move data base information to video screen.
752.00  C*
753.00  CSR          EXSR S004
754.00  CSR          -----
755.00  C*          -----
756.00  CSR          END003  ENDSR
757.00  C*          *****
758.00  C*
759.00  C*          Copy Common Subroutine - Right Justify Numeric Fields
760.00  C*
761.00  C/COPY JDECPY, C0012
762.00  C*          *****
763.00  C*
764.00  C*          SUBROUTINE S004 Load Video Screen Data
765.00  C*          -----
766.00  C*
767.00  C*          Processing 1. Move data base information to video screen.
768.00  C*          All video screen fields re alpha and
769.00  C*          therefore numeric information must be
770.00  C*          processed through subroutine C0014 to set
771.00  C*          proper decimals and provide editing for
772.00  C*          display on screen.
773.00  C*
774.00  C*          Date fields must be converted from their
775.00  C*          internal format of month, day and year or
776.00  C*          Julian to the system format using program
777.00  C*          X0028.

```

JDE uses this or SETLL to release record locks

Moves information to the video/report fields


```

778.00 C*
779.00 CSR          S004  BEGSR
780.00 C*          ----  ----
781.00 C*
782.00 C*
783.00 C*      Move to output -Description for Cost Center
784.00 C*
785.00 CSR          CALL 'X0006'          81
786.00 C*          ----  ----
787.00 CSR          PARM *BLANKS      PSCMOD 1
788.00 CSR          PARM '1'          PSIMOD 1
789.00 CSR          PARM QXXCC        PSMCU 12
790.00 CSR          PARM *BLANKS      PSERRM 4
791.00 CSR          PARM              I0006
792.00 C*
793.00 CSR          MOVE *BLANK        VC0001
794.00 CSR          PDRTRM  IFEQ *BLANK
795.00 CSR          MOVELMCDL01      VC0001
796.00 CSR          END
797.00 C*-----
798.00 C*
799.00 C*      Description display for - Item Type
800.00 C*
801.00 CSR          CLEARI005U
802.00 CSR          MOVELS@XTY        #USX
803.00 CSR          MOVE R@XTY         #URT
804.00 CSR          MOVE QXXTY         #UKY
805.00 CSR          CALL 'X0005'          81
806.00 C*          ----  ----
807.00 CSR          PARM              *0005U
808.00 CSR          MOVE *BLANK        VC0002
809.00 CSR          #UERR  IFEQ '0'
810.00 CSR          MOVEL#UDL01      VC0002
811.00 CSR          END
812.00 C*-----
813.00 C*
814.00 C*      Description display for - Item Unit of Measure
815.00 C*
816.00 CSR          CLEARI0005U
817.00 CSR          MOVELS@XUM        #USY
818.00 CSR          MOVE R@XUM         #URT
819.00 CSR          MOVE QXXUM         #UKY
820.00 CSR          CALL 'X0005'          81
821.00 C*          ----  ----
822.00 CSR          PARM              I0005U
823.00 CSR          MOVE *BLANK        VC0003
824.00 CSR          #UERR  IFEQ '0'
825.00 CSR          MOVEL#UDL01      VC0003
826.00 CSR          END
827.00 C*-----
828.00 C*
829.00 C*      Description display for - Item Category Code 001
830.00 C*
831.00 CSR          CLEARI0005U
832.00 CSR          MOVELS@X001        #USY
833.00 CSR          MOVE R@X001         #URT
834.00 CSR          MOVE QXX001         #UKY
835.00 CSR          CALL 'X0005'          81
836.00 C*          ----  ----
837.00 CSR          PARM              I0005U
838.00 CSR          MOVE *BLANK        VC0004
839.00 CSR          #UERR  IFEQ '0'
840.00 CSR          MOVEL#UDL01      VC0004
841.00 CSR          END
842.00 C*-----
843.00 C*
844.00 C*      Description display for - Item Category Code 002
845.00 C*
846.00 CSR          CLEARI0005U
847.00 CSR          MOVELS@X002        #USY
848.00 CSR          MOVE R@X002         #URT
849.00 CSR          MOVE QXX002         #UKY
850.00 CSR          CALL 'X0005'          81
851.00 C*          ----  ----
852.00 CSR          PARM              I0005U
853.00 CSR          MOVE *BLANK        VC0005
854.00 CSR          #UERR  IFEQ '0'

```

File server for user defined codes

```

855.00  CSR          MOVE#UDL01      VC0005
856.00  CSR          END
857.00  C*-----
858.00  C*
859.00  C*      Description display for - Item Category Code 003
860.00  CSR
861.00  CSR          CLEARI0005U
862.00  CSR          MOVE#S@X003      #USY
863.00  CSR          MOVE R@X003      #URT
864.00  CSR          MOVE QXX003      #UKY
865.00  C*          CALL 'X0005'          81
866.00  CSR
867.00  CSR          PARM              I0005U
868.00  CSR          MOVE *BLANK      VC0006
869.00  CSR          #UERR IFEQ '0'
870.00  CSR          MOVE#UDL01      VC0005
871.00  CSR          END
872.00  C*-----
873.00  C*
874.00  C*      Description display for - Item Category Code 004
875.00  C*
876.00  CSR          CLEARI0005U
877.00  CSR          MOVE#S@X004      #USY
878.00  CSR          MOVE R@X004      #URT
879.00  CSR          MOVE QXX004      #UKY
880.00  C*          CALL 'X0005'          81
881.00  CSR
882.00  CSR          PARM              I0005U
883.00  CSR          MOVE *BLANK      VC0007
884.00  CSR          #UERR IFEQ '0'
885.00  CSR          MOVE#UDL01      VC0007
886.00  CSR          END
887.00  C*-----
888.00  C*
889.00  C*      Description display for - Item Category Code 005
890.00  C*
891.00  CSR          CLEARI0005U
892.00  CSR          MOVE#S@X005      #USY
893.00  CSR          MOVE R@X005      #URT
894.00  CSR          MOVE QXX005      #UKY
895.00  C*          CALL 'X0005'          81
896.00  CSR
897.00  CSR          PARM              I0005U
898.00  CSR          MOVE *BLANK      VC0008
899.00  CSR          #UERR IFEQ '0'
900.00  CSR          MOVE#UDL01      VC0008
901.00  CSR          END
902.00  C*-----
903.00  C*
904.00  C*      Move to output - Cost Center
905.00  C*
906.00  CSR          MOVE *BLANK      #SINBR
907.00  CSR          MOVE#QXXCC      #SINBR
908.00  CSR          MOVE T@XCC      #DTYP
909.00  CSR          MOVE W@XCC      #EWRD
910.00  CSR          MOVE E@XCC      #EC
911.00  CSR          MOVE F@XCC      #DSPD
912.00  CSR          MOVE G@XCC      #DATD
913.00  CSR          MOVE J@XCC      #ALR
914.00  CSR          MOVE ' '      #ECCOR
915.00  CSR          MOVE ' '      #DCOR
916.00  CSR          EXSR C00161
917.00  CSR
918.00  CSR          #ALR IFEQ 'L'
919.00  CSR          MOVE#SINBR      VDXCC
920.00  CSR          ELSE
921.00  CSR          MOVE #SINBR      VDXCC
922.00  CSR          END
923.00  C*-----
924.00  C*
925.00  C*      Move to output - Description
926.00  C*
927.00  CSR          MOVE#QXXDS      VDXDS
928.00  C*-----
929.00  C*
930.00  C*      Move to Output - Date Last Ship
931.00  C*

```

Editing information
retrieved in S998

Copy module to edit field
for use on screen/report

```

932.00 CSR MOVE QXXDT #SIDAT 6
933.00 CSR MOVE *BLANK #EDAT 8
934.00 CSR MOVE *JUL #PFMT 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
939.00 C*
940.00 CSR PARM #SIDAT
941.00 CSR PARM #EDAT
942.00 CSR PARM #PFMT
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 - Iten ID
950.00 C*
951.00 CSR MOVE *BLANK #SINBR
952.00 CSR MOVE QXXIT #SINBR
953.00 CSR MOVE T@XIT #DTYP
954.00 CSR MOVE W@XIT #EWRD
955.00 CSR MOVE B@XIT #EC
956.00 CSR MOVE F@XIT #DSPD
957.00 CSR MOVE G@XIT #DATD
958.00 CSR MOVE J@XIT #ALR
959.00 CSR MOVE ' ' #DCOR
960.00 CSR MOVE ' ' #DCOR
961.00 CSR EXSR C00161
962.00 C*
963.00 CSR #ALR IFBQ '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 MOVE QXXQT #SINBR
974.00 CSR MOVE T@XQT #DTYP
975.00 CSR MOVE W@XQT #RWRD
976.00 CSR MOVE B@XQT #EC
977.00 CSR MOVE F@XQT #DSPD
978.00 CSR MOVE G@XQT #DATD
979.00 CSR MOVE J@XQT #ALR
980.00 CSR MOVE ' ' #DCOR
981.00 CSR MOVE ' ' #DCOR
982.00 CSR EXSR C00161
983.00 C*
984.00 CSR #ALR IFBQ '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 MOVE QXXTY VDXTY
994.00 C*-----
995.00 C*
996.00 C* Move to output - Item Unit of Measure
997.00 C*
999.00 CSR MOVE QXXUM 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 MOVE QXX001 #SINBR
1005.00 CSR MOVE T@X001 #DTYP
1006.00 CSR MOVE W@X001 #EWRD
1007.00 CSR MOVE B@X001 #EC
1009.00 CSR MOVE G@X001 #DATD

```

External program used to edit dates.

```

1010.00      CSR              MOVE J&X001      #ALR
1011.00      CSR              MOVE ' '          #ECOR
1012.00      CSR              MOVE ' '          #DCOR
1013.00      CSR              EXSR C00161
1014.00      C*
1015.00      CSR              #ALR      IFEQ 'L'
1016.00      CSR              MOVE#SINBR      VDX0001
1017.00      CSR              ELSE
1018.00      CSR              MOVE #SINBR      VDXIT
1019.00      CSR              END
1020.00      C*-----
1021.00      C*
1022.00      C*      Move to output - Item Category Code 002
1023.00      C*
1024.00      CSR              MOVE *BLANK      #SINBR
1025.00      CSR              MOVE#QXX002     #SINBR
1026.00      CRR              MOVE T&X002     #DTYP
1027.00      CSR              MOVE W&X002     #RWRD
1028.00      CSR              MOVE E&X002     #EC
1029.00      CSR              MOVE F&X002     #DSPD
1030.00      CSR              MOVE G&X002     #DATD
1031.00      CSR              MOVE J&X002     #ALR
1032.00      CSR              MOVE ' '          #ECOR
1033.00      CSR              MOVE ' '          #DCOR
1034.00      CSR              EXSR C00161
1035.00      C*
1036.00      CSR              #ALR      IFEQ 'L'
1037.00      CSR              MOVE#SINBR      VDX002
1038.00      CSR              ELSE
1039.00      CSR              MOVE #SINBR      VDX002
1040.00      CSR              END
1041.00      C*-----
1042.00      C*
1043.00      C*      Move to output - Item Category Code 003
1044.00      C*
1045.00      CSR              MOVE *BLANK      #SINBR
1046.00      CSR              MOVE#QXX003     #SINBR
1047.00      CSR              MOVE T&X003     #DTYP
1048.00      CSR              MOVE W&X003     #EWRD
1049.00      CSR              MOVE E&X003     #EC
1050.00      CSR              MOVE F&X003     #DSPD
1051.00      CSR              MOVE G&X003     #DATD
1052.00      CSR              MOVE J&X003     #ALR
1053.00      CSR              MOVE ' '          #ECOR
1054.00      CSR              MOVE ' '          #DCOR
1055.00      CSR              EXSR C00161
1056.00      C*
1057.00      CSR              #ALR      IFEQ 'L'
1058.00      CSR              MOVE#SINBR      VDX003
1059.00      CSR              ELSE
1060.00      CSR              MOVE #SINBR      VDX003
1061.00      CSR              END
1062.00      C*-----
1063.00      C*
1064.00      C*      Move to output - Item Category Code 004
1065.00      C*
1066.00      CSR              MOVE *BLANK      #SINBR
1067.00      CSR              MOVE#QXX004     #SINBR
1068.00      CSR              MOVE T&X004     #DTYP
1069.00      CSR              MOVE W&X004     #EWRD
1070.00      CSR              MOVE E&X004     #EC
1071.00      CSR              MOVE F&X004     #DSPD
1072.00      CSR              MOVE G&X004     #DATD
1072.00      CSR              MOVE J&X004     #ALR
1074.00      CSR              MOVE ' '          #ECOR
1075.00      CSR              MOVE ' '          #DCOR
1076.00      CSR              EXSR C00161
1077.00      C*
1078.00      CSR              #ALR      IFEQ 'L'
1079.00      CSR              MOVE#SINBR      VDX004
1080.00      CSR              ELSE
1081.00      CSR              MOVE #SINBR      VDX004
1082.00      CSR              END
1083.00      C*-----
1084.00      C*
1085.00      C*      Move to output - Item Category Code 005
1086.00      C*

```

```

1087.00 CSR          MOVE *BLANK      #SINBR
1088.00 CSR          MOVELQXX005    #SINBR
1089.00 CFR          MOVE T&X005    #DTYP
1090.00 CSR          MOVE W&X005    #EWRD
1091.00 CSR          MOVE E&X005    #EC
1092.00 CSR          MOVE F&X005    #DSFD
1093.00 CSR          MOVE G&X005    #DATD
1094.00 CSR          MOVE J&X005    #ALR
1095.00 CSR          MOVE ' '       #DCOR
1096.00 CSR          MOVE ' '       #DCOR
1097.00 CSR          EXSR C00161
1098.00 C*          -----
1099.00 CSR          #ALR  IFEQ 'L'
1100.00 CSR          MOVEL#SINBR    VDX005
1101.00 CSR          ELSE
1102.00 CSR          MOVE #SINBR    VDX005
1103.00 CSR          END
1104.00 C*          -----
1105.00 CSR          END004  ENDSR
1106.00 C*          *****
1107.00 C*
1108.00 C*          Copy Common Subroutine - Format Numeric Fields for Output with Override
1109.00 C*
1110.00 C/COPY JDECPY,C00161
1111.00 C*          *****
1112.00 C*
1113.00 C*          SUBROUTINE S005 - Scrub Input
1114.00 C*          -----
1115.00 C*
1116.00 C*          Processing:  1.  Validate all video input.
1117.00 C*                      All numeric fields must be processed
1118.00 C*                      through subroutines C0012 and C0015 in order
1119.00 C*                      to scrub the alpha input field and convert
1120.00 C*                      15 digits and 0 decimals.
1121.00 C*
1122.00 C*                      Date fields must be converted from system
1123.00 C*                      format to their internal format of month,
1124.00 C*                      day and year or julian using program X0028.
1125.00 C*          2.  Update data record fields from video.
1126.00 C*
1127.00 CSR          S005  BEGSR
1128.00 C*          -----
1129.00 C*
1130.00 C*          If not addition or change, bypass subroutine
1131.00 C*
1132.00 CSR          *IN21  IFEQ '0'
1133.00 CSR          *IN22  ANDEQ '0'
1134.00 CSR          GOTO END005
1135.00 C*          -----
1136.00 CSR          END
1137.00 C*
1138.00 C*
1139.00 C*
1140.00 C*          Scrub and edit - Cost Center
1141.00 C*
1142.00 CSR          CALL 'X0006'          99
1143.00 C*          -----
1144.00 CSR          PARM '1'          PSOMOD  1
1145.00 CSR          PARM '1'          PSIMOD  1
1146.00 CSR          PARM VDXCC       PSMCU   12
1147.00 CSR          PARM *BLANKS     PSERRM  4
1148.00 CSR          PARM              I0006
1149.00 C*
1150.00 CSR          PSERRM  IFNE *BLANK
1151.00 CSR          SETON              4393
1152.00 CSR          MOVELPSERRM      EMK,10
1153.00 CSR          MOVE '1'         @MK,10
1154.00 CSR          END
1155.00 CSR          MOVE PSMCU       QXXCC
1156.00 C*          -----
1157.00 C*
1158.00 C*          Scrub and edit - Description
1159.00 C*
1160.00 CSR          MOVELVDXDS        QXXDS
1161.00 C*
1162.00 C*          Set default value - Description
1163.00 C*

```

SUBROUTINE S005 - Scrub Input

Validates and edits data entered by the user

If not addition or change, bypass subroutine
*IN21 IFEQ '0'
*IN22 ANDEQ '0'
GOTO END005

END

Only performs this subroutine if a record is added or changed

```

1165.00 CSR      QXXDS      UFEQ *BLANK
1166.00 CSR      D@XDS      IFNE *BLANK
1167.00 CSR      MOVEAD@XDS @DV
1168.00 CSR      MOVE@DV      QXXDS
1169.00 CSR      @DV,1      IFEQ ' ' ' '
1170.00 CSR      MOVE ' ' ' ' @DV,1
1171.00 CSR      Z-ADD2      #M
1172.00 CSR      #M      DOWLE40
1173.00 CSR      @DV,#M      IFEQ ' ' ' '
1174.00 CSR      MOVE ' ' ' ' @DV,#m
1175.00 CSR      END
1176.00 CSR      ADD 1      #M
1177.00 CSR      END
1178.00 CSR      MOVEA@DV,2 QXXDS
1179.00 CSR      END
1180.00 CSR      END
1181.00 CSR      END
1182.00 C*
1183.00 C*      Edit allowed values - Description
1184.00 C*
1185.00 CSR      A@XDS      IFEQ '*NB'
1186.00 CSR      QXXDS      ANDEQ*BLANK
1187.00 CSR      MOVE '1'      @MK,03
1188.00 CSR      SETON      4293
1189.00 CSR      END
1190.00 C*
-----
1191.00 C*
1192.00 C*      Scrub and edit - Date Last Ship
1193.00 C*
1194.00 CSR      MOVEAVDXDT @NM
1195.00 CSR      EXSR C0012
1196.00 C*      ----
1197.00 CSR      Z-ADD#NUME NBR6 60
1198.00 CSR      MOVE $NBR6 QXXDT
1199.00 C*
1200.00 C*      Edit julian date - Date Last Ship
1201.00 C*
1202.00 CSR      VDXDT      IFNE * BLANK
1203.00 CSR      MOVE QXXDT      #SIDAT 6
1204.00 CSR      MOVE *BLANK      #EDAT 8
1205.00 CSR      MOVEV *SYSVAL      '#PFMT 7
1206.00 CSR      MOVEV *JUL      '#TFMT 7
1207.00 CSR      MOVEV *NONE      '#SKP 7
1208.00 CSR      MOVEV ' '      $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      #PFMT
1214.00 CSR      PARM      #TFMT
1215.00 CSR      PARM      #SKP
1216.00 CSR      PARM      $KRTST
1217.00 CSR      MOVEV#SIDAT QXXDT
1218.00 CSR      $ERTST      IFEV '1'
1219.00 CSR      MOVE '1'      @MK,04
1220.00 CSR      SETON      4593
1221.00 CSR      END
1222.00 CSR      END
1223.00 C*
-----
1224.00 C*
1225.00 C*      Scrub and edit - Item ID
1226.00 C*
1227.00 CSR      MOVEAVDXIT @NM
1228.00 CSR      EXSR C0012
1229.00 C*      ----
1230.00 CSR      MOVE F@XIT      #DSPD
1231.00 CSR      MOVE G@XIT      #DATD
1232.00 CSR      EXSR C00151
1233.00 C*
-----
1234.00 CSR      MOVE #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      D@XIT      ANDNE*BLANK
1240.00 CSR      MOVEAD@XIT @NM
1241.00 CSR      EXSR C0012

```

Work fields used in the RPG program begin with \$

Work fields used in a copy module begin with #

```

1242.00 C*
1243.00 CSR MOVE P@XIT #DSPD
1244.00 CSR MOVE G@XIT #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 L@XIT IPNE *BLANK
1253.00 CSR MOVE *BLANK X@XIT 15
1254.00 CSR MOVE '1' $ERTST 1
1255.00 CSR MOVE L@XIT X@XIT
1256.00 CSR X@XIT IPEG L@XIT
1257.00 CSR X@XIT ANDLEU@XIT
1258.00 CSR MOVE ' ' $ERTST
1259.00 CSR END
1260.00 CSR $ERTST IPEQ '1'
1261.00 CSR MOVE '1' @MK,07
1262.00 CSR SETON 4193
1263.00 CSR END
1264.00 CSR END
1265.00 C*-----
1266.00 C*
1267.00 C* Scrub and edit - Quantity - On Hand
1268.00 C*
1269.00 CSR MOVEAVDXQT @NM
1270.00 CSR EXSR C0012
1271.00 C*
1272.00 CSR MOVE P@XQT #DSPD
1273.00 CSR MOVE G@XQT #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
1279.00 C*
1280.00 CSR VDXQT IPEQ *BLANK
1281.00 CSR D@XQT ANDNE*BLANK
1282.00 CSR MOVEAD@XQT @NM
1283.00 CSR EXSR C0012
1284.00 C*
1285.00 CSR MOVE P@XQT #DSPD
1286.00 CSR MOVE G@XQT #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
1293.00 C*
1294.00 CSR L@XQT IPNE *BLANK
1295.00 CSR MOVE *BLANK X@XQT 15
1296.00 CSR MOVE '1' $ERTST 1
1297.00 CSR MOVE L@XQT X@XQT
1298.00 CSR X@XQT IPEG L@XQT
1299.00 CSR X@XQT ANDLEU@XQT
1300.00 CSR MOVE ' ' $ERTST
1301.00 CSR END
1302.00 CSR $ERTST IPEQ '1'
1303.00 CSR MOVE '1' @MK,07
1304.00 CSR SETON 4693
1305.00 CSR END
1306.00 CSR END
1307.00 C*-----
1308.00 C*
1309.00 C* Scrub and edit - Item Type
1310.00 C*
1311.00 CSR MOVELVDXTY QXXTY
1312.00 C*
1313.00 C* Set default value - Item Type
1314.00 C*
1315.00 CSR QXXTY IPEQ *BLANK
1316.00 CSR D@XTY IPNE *BLANK
1317.00 CSR MOVEAD@XTY @40
1318.00 CSR MOVEA@40 QXXTY

```

1319.00	CSR	@40,1	IFEQ ''''			
1320.00	CSR		MOVE ' '	@40,1		
1321.00	CSR		Z-ADD2	#M		
1322.00	CSR	#M	DOWLE40			
1323.00	CSR	@40,#M	IFEQ ''''			
1324.00	CSR		MOVE ' '	@40,#M		
1325.00	CSR		END			
1326.00	CSR		ADD 1	#M		
1327.00	CSR		END			
1328.00	CSR		MOVEA@40,2	QXXTY		
1329.00	CSR		END			
1330.00	CSR		END			
1331.00	CSR		END			
1332.00	C*					
1333.00	C*	Edit allowed values - Item Type				
1334.00	C*					
1335.00	CSR	A@XTY	IPNE *BLANK			
1336.00	CSR	A@XTY	IFEQ '*NB'			
1337.00	CSR	QXXTY	ANDEQ*BLANK			
1338.00	CSR		MOVE '1'	@MK,03		
1339.00	CSR		SETON		4493	
1340.00	CSR		ELSE			
1341.00	CSR		MOVEA@XTY	@40		
1342.00	CSR		MOVE *HIVAL	@AV		
1343.00	CSR		EXSR C997			
1344.00	C*		-----			
1345.00	CSR		MOVE ' '	\$ERTST 1		
1346.00	CSR		MOVE *BLANK	\$WRK10 10		
1347.00	CSR		MOVELQXXTY	\$WRK10		
1348.00	CSR	@AV,1	IPNE *HIVAL			
1349.00	CSR	\$WRK10	LOKPUP@AV		81	
1350.00	CSR	*IN81	IFEQ '0'			
1351.00	CSR		MOVE '1'	\$ERTST		
1352.00	CSR		END			
1353.00	CSR	\$ERTST	IFEQ '1'			
1354.00	C*		MOVE '1'	O*,07		
1355.00	CSR		SETON		4493	
1356.00	CSR		END			
1357.00	CSR		END			
1358.00	CSR		END			
1359.00	CSR		END			
1360.00	C*					
1361.00	C*	Edit upper and lower range - Item Type				
1362.00	C*					
1363.00	CSR	LQXTY	IPNE *BLANK			
1364.00	CSR		MOVE '1'	\$ERTST		
1365.00	CSR	QXXTY	IPGE L@XTY			
1366.00	CSR	QXXTY	ANDLEU@XTY			
1367.00	CSR		MOVE ' '	\$ERTST		
1368.00	CSR		END			
1369.00	CSR	\$ERTST	IFEQ '1'			
1370.00	CSR		MOVE '1'	@MK,07		
1371.00	CSR		SETON		4493	
1372.00	CSR		END			
1373.00	CSR		END			
1374.00	C*					
1375.00	C*	Edit from User Defined Codes - Item Type				
1376.00	C*					
1377.00	CSR	R@XTY	IPNE *BLANK			
1378.00	CSR		CLEARZ0005U			
1379.00	C*		MOVELS@XTY	#USY		
1380.00	CSR		MOVE R@XTY	#URT		
1381.00	CSR		MOVE QXXTY	#UKR		
1382.00	CSR		CALL 'X0005'		81	
1383.00	C*					
1384.00	CSR		PARM	I0005U		
1385.00	CSR	#UERR	IFEQ '1'			
1386.00	CSR		MOVE '1'	@MK,09		
1387.00	CSR		SETON		4493	
1388.00	CSR		END			
1389.00	CSR		END			
1390.00	C*	-----				
1391.00	C*					
1392.00	C*	Scrub and edit - Item Unit of Measure				
1393.00	C*					
1394.00	CSR		MOVELVDXUN	QXXUM		
1395.00	C*					


```

1396.00 C*      Set default value - Item Unit of Measure
1397.00 C*
1398.00 CSR      QXXUM  IFBQ *BLANK
1399.00 CSR      E!XUM   IFNE *BLANK
1400.00 CSR      MOVEAD@XUM @40
1401.00 CSR      MOVEA@40  QXXUM
1402.00 CSR      @40,1  IFBQ ' ' '
1403.00 CSR      MOVE ' ' @40,1
1404.00 CSR      Z-ADD2   #M
1405.00 CSR      #M      DOWLE40
1406.00 C*      @40,#M  IFBQ ' ' '
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   IFNE *BLANK
1419.00 CSR      A@XUM   IFBQ '*NB'
1420.00 CSR      QXUM    ANDEQ*BLANK
1421.00 CSR      MOVE '1' @MK,03
1422.00 CSR      SETON                               4793
1423.00 CSR      ELSE
1424.00 CSR      MOVEA@XUM @40
1425.00 CSR      MOVE *HIVAL @AV
1426.00 CSR      EXSR C997
1427.00 C*      ----
1428.00 CSR      MOVE ' ' $ERTST 1
1429.00 CSR      MOVE *BLANK $WRK10 10
1430.00 CSR      MOVE LQXXUM $WRK10
1431.00 CSR      @AV,1  IFNE *HIVAL
1432.00 CSR      $WRK10 LOKUP@AV 81
1433.00 CSR      *IN8I  IFBQ '0'
1434.00 CSR      MOVE '1' $ERTST
1435.00 CSR      END
1436.00 C*      $ERTST IFBQ '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   IFNE *BLANK
1447.00 CSR      MOVE '1' $ERTST
1448.00 CSR      QXXUM   IFBQ L@XUM
1449.00 CSR      OXTON   kNO~UOXON
1450.00 CSR      MOVE ' ' $ERTST
1451.00 CSR      END
1452.00 CSR      $ERTST IFBQ '1'
1453.00 C*      MOVE '1' @MK,07
1454.00 CSR      SETON                               4793
1455.00 CSR      END
1456.00 C*      END
1457.00 C*
1458.00 C*      Edit from User Defined Codes - Item Unit of Measure
1459.00 C*
1460.00 CSR      R@XUM   IFNE *BLANK

```

```

1473.00 C*-----
1474.00 C*
1475.00 C*   Scrub and edit - Item Category Code 001
1476.00 C*
1477.00 CSR           MOVELVDX001   QXX001
1478.00 C*
1479.00 C*   Set default value - Item Category Code 001
1480.00 C*
1481.00 CSR           QXX001   IFEQ *BLANK
1482.00 CSR           D@X001   IFNE *BLANK
1483.00 CSR           MOVEAD@X001   @40
1484.00 CSR           MOVEA@40   QXX001
1485.00 CSR           @40,1   IFEQ ' ' '
1486.00 CSR           MOVE ' '   @40,1
1487.00 CSR           Z-ADD2   #M
1488.00 CSR           #M   DOWLE40
1489.00 CSR           @40,#m   IFEQ
1490.00 CSR           MOVE ' '   @40,#M
1491.00 CSR           END
1492.00 CSR           ADD 1   #M
1493.00 CSR           END
1494.00 CSR           MOVEA@40,2   QXX001
1495.00 CSR           END
1496.00 CSR           END
1497.00 CSR           END
1498.00 C*
1499.00 C*   Edit allowed values - Item Category Code 001
1500.00 C*
1501.00 CSR           A@X001   IFNE *BLANK
1502.00 CSR           A@X001   IFEQ '*NB'
1503.00 CSR           QXX001   ANDEQ*BLANK
1504.00 CSR           MOVE '1'   @MK,03
1505.00 CSR           SETON   4893
1506.00 CSR           ELSE
1507.00 CSR           MOVEAA@X001   @40
1508.00 CSR           MOVE *HIVAL   @AV
1509.00 CSR           EXSR C997
1510.00 C*
1511.00 CSR           MOVE ' '   $ERTST 1
1512.00 CSR           MOVE *BLANK   $WRK10 10
1513.00 CSR           MOVELVQXX001   $WRK10
1514.00 CSR           @AV,1   IFNE *HIVAL
1515.00 CSR           $WRK10   LOKUP@AV   81
1516.00 CSR           *IN81   IFEQ '0'
1517.00 CSR           MOVE '1'   $ERTST
1518.00 CSR           END
1519.00 CSR           $ERTST   IFEQ '1'
1520.00 CSR           MOVE '1'   @MK,07
1521.00 CSR           SETON   4893
1522.00 CSR           END
1523.00 CSR           END
1524.00 CSR           END
1525.00 CSR           END
1526.00 C*
1527.00 C*   Edit upper and lower range - Item Category Code 001
1528.00 C*
1529.00 CSR           L@X001   IFNE *BLANK
1530.00 CSR           MOVE '1'   $ERTST
1531.00 CSR           QXX001   IFGE L@X001
1532.00 CSR           QXX001   ANDLEU@X001
1533.00 CSR           MOVE ' '   $ERTST
1534.00 CSR           END
1535.00 CSR           $ERTST   IFEQ '1'
1536.00 CSR           MOVE '1'   @MK,07
1537.00 CSR           SETON   4893
1538.00 CSR           END
1539.00 CSR           END
1540.00 C*
1541.00 C*   Edit from User Defined Codes - Item Category Code 001
1542.00 C*
1543.00 CSR           R@X001   IFNE *BLANK
1544.00 CSR           CLEAR10005U
1545.00 CSR           MOVELS@X001   #USY
1546.00 CSR           MOVE R@X001   #URT
1547.00 CSR           MOVE QXX001   #UKY
1548.00 CSR           CALL 'X0005'   81
1549.00 C*   -----

```

```

1550.00 CSR          PARM          I0005U
1551.00 CSR          IFEQ '1'
1552.00 CSR          #UERR        MOVE '1'          @MK,09
1553.00 CSR          SETON
1554.00 CSR          END          4893
1555.00 CSR          END
1556.00 C*-----
1557.00 C*
1558.00 C*      Scrub and edit - Item Category Code 002
1559.00 C*
1560.00 CSR          MOVELVDX002    QXX002
1561.00 C*
1562.00 C*      Set default value - Item Category Code 002
1563.00 C*
1564.00 CSR          QXX002        IFEQ *BLANK
1565.00 CSR          D@X002        IFNE *BLANK
1566.00 CSR          MOVEADX002     @40
1567.00 CSR          MOVEA@40      QXX002
1568.00 CSR          @40,1        IFEQ ' '
1569.00 CSR          MOVE ' '      @40,1
1570.00 CSR          Z-ADD2        #M
1571.00 CSR          #M           DOWLE40
1572.00 CSR          @40,#M       IFEQ ' '
1573.00 CSR          MOVE ' '      @40,#M
1574.00 CSR          END
1575.00 CSR          ADD 1         #M
1576.00 CSR          END
1577.00 CSR          MOVEA@40,2    QXX002
1578.00 CSR          END
1579.00 CSR          END
1580.00 CSR          END
1581.00 C*
1582.00 C*      Edit allowed values - Item Category Code 002
1583.00 C*
1584.00 CSR          A@X002        IFNE *BLANK
1585.00 CSR          A@X002        IFEQ '*NB'
1586.00 CSR          QXX002        ANDEQ*BLANK
1587.00 CSR          MOVE '1'      @MK,03
1588.00 CSR          SETON          4993
1589.00 CSR          ELSE
1590.00 CSR          MOVEAAX002     @40
1591.00 CSR          MOVE *HIVAL    @AV
1592.00 CSR          EXSR C997
1593.00 C*      -----
1594.00 CSR          MOVE ' '      $ERTST 1
1595.00 CSR          MOVE *BLANK    $WRK10 10
1596.00 CSR          MOVELQXX002    $WRK10
1597.00 CSR          @AV,1         IFNE *HIVAL
1598.00 CSR          $WFRK10       LOKUP@AV          81
1599.00 CSR          *IN81        IFEQ '0'
1600.00 CSR          MOVE '1'      $ERTST
1601.00 CSR          END
1602.00 CSR          $ERTST        IFEQ '1'
1603.00 CSR          MOVE '1'      @MK,07
1604.00 CSR          SETON          4993
1605.00 CSR          END
1606.00 CSR          END
1607.00 CSR          END
1608.00 CSR          END
1609.00 C*
1610.00 C*      Edit upper and lower range - Item Category Code 002
1611.00 C*
1612.00 CSR          L@X002        IFNE *BLANK
1613.00 CSR          MOVE '1'      $ERTST
1614.00 CSR          QXX002        IFGE L@X002
1615.00 CSR          QXX002        ANDLEU@X002
1616.00 CSR          MOVE ' '      $ERTST
1617.00 CSR          END
1618.00 CSR          $ERTST        IFEQ '1'
1619.00 CSR          MOVE '1'      @MK,07
1620.00 CSR          SETON          4993
1621.00 CSR          END
1622.00 CSR          END
1623.00 C*
1624.00 C*      Edit from User Defined Codes - Item Category Code 002
1625.00 C*
1626.00 CSR          R@X002        IFNE *BLANK

```

1627.00	CSR		CLEARI0005U		
1628.00	CSR		MOVELS@X002	#USY	
1629.00	CSR		MOVE R@X002	#URT	
1630.00	CSR		MOVE QXX002	OUKr	
1631.00	CSR		CALL 'X0005'		81
1632.00	C*		-----		
1633.00	CSR		PARM	I0005U	
1634.00	CSR	#UERR	IFBQ '1'		
1635.00	CSR		MOVE '1'	@MK,09	
1636.00	CSR		SETON		4993
1637.00	CSR		END		
1638.00	CSR		END		
1639.00	C*		-----		
1640.00	C*				
1641.00	C*		Scrub and edit - Item Category Code 003		
1642.00	C*				
1643.00	CSR		MOVELVDX003	QXX003	
1644.00	C*				
1645.00	C*		Set default value - Item Category Code 003		
1646.00	C*				
1647.00	CSR	QXX003	IFBQ *BLANK		
1648.00	CSR	D@X003	IFNE *BLANK		
1649.00	CSR		MOVEAD@X003	@40	
1650.00	CSR		MOVEA@40	QXX003	
1651.00	CSR	@40,1	IFBQ ''''		
1652.00	CSR		MOVE ' '	@40,1	
1653.00	CSR		Z-ADD2	#M	
1654.00	CSR	#M	DOWLE40		
1655.00	CSR	@40,#M	IFBQ ''''		
1656.00	CSR		MOVE ' '	@40,#M	
1657.00	CSR		END		
1658.00	CSR		ADD 1	#M	
1659.00	CSR		END		
1660.00	CSR		MOVEA@40,2	QXX003	
1661.00	CSR		END		
1662.00	CSR		END		
1663.00	CSR		END		
1664.00	C*				
1665.00	C*		Edit allowed values - Item Category Code 003		
1666.00	C*				
1667.00	CSR	A@X003	IFNE *BLANK		
1668.00	CSR	A@X003	IFBQ '*NB'		
1669.00	CSR	QXX003	ANDEQ*BLANK		
1670.00	CSR		MOVE '1'	@MK,03	
1671.00	CSR		SETON		5093
1672.00	CSR		ELSE		
1673.00	CSR		MOVEA@003	@40	
1674.00	CSR		MOVE *HIVAL	@AV	
1675.00	CSR		EXSR C997		
1676.00	C*		-----		
1677.00	CSR		MOVE ' '	\$ERTST 1	
1678.00	CSR		MOVE *BLANK	\$WRK10 10	
1679.00	CSR		MOVELQXX003	\$WRK10	
1680.00	CSR	@AV,1	IFNE *HIVAL		
1681.00	CSR	\$WRK10	LOKUP@AV		81
1682.00	CSR	*IN81	IFBQ '0'		
1683.00	CSR		MOVE '1'	\$ERTST	
1684.00	CSR		END		
1685.00	CSR	\$ERTST	IFBQ '1'		
1686.00	CSR		MOVE '1'	@MK,07	
1687.00	CSR		SETON		5093
1688.00	CSR		END		
1689.00	CSR		END		
1690.00	CSR		END		
1691.00	CSR		END		
1692.00	C*				
1693.00	C*		Edit upper and lower range - Item Category Code 003		
1694.00	C*				
1695.00	CSR	L@X003	IFNE *BLANK		
1696.00	CSR		MOVE '1'	\$ERTST	
1697.00	CSR	QXX003	IFGE L@X003		
1698.00	CSR	QXX003	ANDLEU@X003		
1699.00	CSR		MOVE ' '	\$ERTST	
1700.00	CSR		END		
1701.00	CSR	\$ERTST	IFBQ '1'		
1702.00	CSR		MOVE '1'	@MK,07	
1703.00	CSR		SETON		5093

```

1704.00 CSR          END
1705.00 CSR          END
1706.00 C*
1707.00 C*      Edit from User Defined Codes - Item Category Code 003
1708.00 C*
1709.00 CSR          R@X003      IFNE *BLANK
1710.00 CSR          CLEARI0005U
1711.00 CSR          MOVELS@X003      #USY
1712.00 CSR          MOVE R@X003      #URT
1713.00 CSR          MOVE QXX003      #UKY
1714.00 CSR          CALL 'X0005'      81
1715.00 C*          -----
1716.00 CSR          PARM          I0005U
1717.00 CSR          #UERR      IPEQ '1'
1718.00 CSR          MOVE '1'      @MX,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      IPEQ *BLANK
1731.00 CSR          D@X004      IFNE *BLANK
1732.00 CSR          MOVEAD@X004      @40
1733.00 CSR          MOVEA@40      QXX004
1734.00 CSR          @40,1      IPEQ ' '
1735.00 CSR          MOVE ' '      @40,1
1736.00 CSR          Z-ADD2      #M
1737.00 CSR          #M          DOWLE40
1738.00 CSR          @40,#M      IPEQ ' '
1739.00 CSR          MOVE ' '      @40,#M
1740.00 CSR          END
1741.00 CSR          ADD 1      #M
1742.00 CSR          END
1743.00 CSR          MOVEA@40,2      QXX004
1744.00 CSR          END
1745.00 CSR          END
1746.00 CSR          END
1747.00 C*
1748.00 C*      Edit allowed values - Item Category Code 004
1749.00 C*
1750.00 CSR          A@X004      IFNE *BLANK
1751.00 CSR          A@X004      IPEQ '*NB'
1752.00 CSR          QXX004      ANDEQ*BLANK
1753.00 CSR          MOVE '1'      @MX,03
1754.00 CSR          SETON          5193
1755.00 CSR          ELSE
1756.00 CSR          MOVEAA@X004      @40
1757.00 CSR          MOVE *HIYAL      @AV
1758.00 CSR          EXSR C997
1759.00 C*          -----
1760.00 CSR          MOVE ' '      $ERTST
1761.00 CSR          MOVE *BLANK      $WRK10 10
1762.00 CSR          MOVELQXX004      $WRK10
1763.00 CSR          @AV,1      IFNE *HIYAL
1764.00 CSR          $WRK10      LOKUP@AV      81
1765.00 CSR          *IN81      IPEQ '0'
1766.00 CSR          MOVE '1'      $ERTST
1767.00 CSR          END
1768.00 CSR          $ERTST      IPEQ '1'
1769.00 CSR          MOVE '1'      @MX,07
1770.00 CSR          SETON          5193
1771.00 CSR          END
1772.00 CSR          END
1773.00 CSR          END
1774.00 CSR          END
1775.00 C*
1776.00 C*      Edit upper and lower range - Item Category Code 004
1777.00 C*
1778.00 CSR          L@X004      IFNE *BLANK
1779.00 CSR          MOVE '1'      $ERTST
1780.00 CSR          QXX004      IFGE L@X004

```

1781.00	CSR	QXX004	ANDLEU@X004		
1782.00	CSR		MOVE '1'	\$ERTST	
1783.00	CSR		END		
1704.00	CSR	\$ERTST	IFBQ '1'		
1785.00	CSR		MOVE '1'	@MK,07	
1786.00	CSR		SETON		5193
1797.00	CSR		END		
1788.00	CSR		END		
1789.00	C*				
1790.00	C*	Edit from User Defined Codes - Item Category Code 004			
1791.00	C*				
1792.00	CSR	R@X004	IFNE *BLANK		
1793.00	CSR		CLEARIG005U		
1794.00	CSR		MOVELS@X004	#USY	
1795.00	CSR		MOVE R@X004	#URT	
1796.00	CSR		MOVE QXX004	#UKY	
1797.00	CSR		CALL 'X0005'		81
1798.00	C*		-----		
1799.00	CSR		FARM	I0005U	
1800.00	CSR	#UERR	IFBQ '1'		
1801.00	CSR		MOVE '1'	@MK,09	
1802.00	CSR		SETON		5193
1803.00	CSR		END		
1804.00	CSR		END		
1805.00	C*		-----		
1806.00	C*				
1807.00	C*	Scrub and edit - Item Category Code 005			
1808.00	C*				
1809.00	CSR		MOVELV0X00S	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@X00S	IFNE *BLANK		
1815.00	CSR		MOVEAD@X00S	@40	
1816.00	CSR		MOVEA@40	QXX005	
1817.00	CSR	@40,1	IFBQ ''''		
1818.00	CSR		MOVE ' '	@40,1	
1819.00	CSR		Z-ADD2	#M	
1820.00	CSR	#M	DOWLE40		
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		MOVEA@40,2	QXX005	
1827.00	CSR		END		
1828.00	CSR		END		
1829.00	CSR		END		
1830.00	C*				
1931.00	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	ANDEQ*BLANK		
1836.00	CSR		MOVE '1'	@MK,03	
1837.00	CSR		SETON		5293
1838.00	CSR		ELSE		
1839.00	CSR		MOVEA@X005	@40	
1840.00	CSR		MOVE *HIVAL	@AV	
1841.00	CSR		EXSR C997		
1842.00	C*		-----		
1843.00	CSR		MOVE ' '	\$ERTST 1	
1844.00	CSR		MOVE *BLANK	\$WRK10 10	
1845.00	CSR		MOVEQXX005	\$WRK10	
1846.00	CSR	@AV,1	IFNE *HIVAL		
1847.00	CSR	\$WRK10	LOKUP@AV		81
1848.00	CSR	*IN81	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		SETON		5293
1854.00	CSR		END		
1855.00	CSR		END		
1856.00	CSR		END		
1857.00	CSR		END		

```

1858.00 C*
1859.00 C*      Edit upper and lower range - Item Category Code 005
1860.00 C*
1861.00 CSR      L@X005      IFNE *BLANK
1862.00 CSR      MOVE '1'      $ERTST
1863.00 CSR      QXX005      IFGE L@X005
1864.00 CSR      QXX005      ANDLEU@X005
1865.00 CSR      MOVE ' '      $ERTST
1866.00 CSR      END
1867.00 CSR      $ERTST      IFEQ '1'
1868.00 CSR      MOVE '1'      @MK,07
1869.00 CSR      SETON      5293
1870.00 CSR      END
1871.00 CSR      END
1872.00 C*
1873.00 C*      Edit from User Defined Codes - Item Category Code 005
1874.00 C*
1875.00 CSR      R@X005      IFNE *BLANK
1876.00 CSR      CLEARI0005U
1877.00 CSR      MOVELS@X005      #USY
1878.00 CSR      MOVE R@X005      #URT
1879.00 CSR      MOVE QXX005      #UKY
1880.00 CSR      CALL 'X0005'      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      ENDO005      ENDSR
1890.00 C*****
1891.00 C*
1892.00 C*      Copy Common Subroutine - Currency - Translate Video Fields to Data Base
1893.00 C*
1894.00 C/COPY JDECPY,C00151
1895.00 C*****
1896.00 C*
1897.00 C*      Copy Common Subroutine - Build Allowed Values Work Array
1898.00 C*
1899.00 C/COPY JDECPY,C997
1900.00 C*****
1901.00 C*
1902.00 C*      Subroutine S010 - Update Data Base
1903.00 C*      -----
1904.00 C*
1905.00 C*      Processing: 1. Update data base file based upon valid
1906.00 C*                  action codes.
1907.00 C*
1908.00 CSR      S010      BEGSR
1909.00 C*      ----      ----
1910.00 C*
1911.00 C*      If add action, add record.
1912.00 C*
1913.00 CSR      *IN21      IFEQ '1'
1914.00 CSR      WRITEI92801      99
1915.00 CSR      END
1916.00 C*
1917.00 C*      If change action, update record.
1918.00 C*
1919.00 CSR      *IN22      IFEQ '1'
1920.00 CSR      UFDATI92801      99
1921.00 CSR      END
1922.00 C*
1923.00 C*      If delete action, delete record.
1924.00 C*
1925.00 CSR      *IN23      IFEQ '1'
1926.00 CSR      DELETI92801      99
1927.00 CSR      END
1928.00 C*

```

```

1929.00 C* Clear data field for next transaction
1930.00 C*
1931.00 CSR MOVE #PCLR @@AID
1932.00 CSR EXSR S001
1933.00 C* -----
1934.00 CSR END010 ENDSR
1935.00 C*****
1935.00 C*
1936.00 C* SUBROUTINE S998 - Load dictionary parameters.
1937.00 C* -----
1938.00 C*
1939.00 CSR S998 BEGSR
1940.00 C* -----
1941.00 C*
1942.00 C*
1943.00 C*
1944.00 C* Dictionary parameters for - Cost Center
1945.00 C*
1946.00 CSR MOVE *BLANK FRDTAI
1947.00 CSR MOVEL'XCC' FRDTAI
1948.00 CSR CALL 'X9800E' 81
1949.00 C* -----
1950.00 CSR PARM I9800E
1951.00 CSR FRERR IPRQ '0'
1952.00 CSR MOVE FRDSCR B@XCC 40
1953.00 CSR MOVE FRDTAT T@XCC 1
1954.00 CSR MOVE FREC E@XCC 1
1955.00 CSR MOVE FRDTAS C@XCC 50
1956.00 CSR MOVE FROTAD G@XCC 20
1957.00 CSR MOVE FRCDEC F@XCC 1
1958.00 CSR MOVE LFRSY S@XCC 4
1959.00 CSR MOVE FRRT R@XCC 2
1960.00 CSR MOVE FRDVAL D@XCC 40
1961.00 CSR MOVE FRVAL A@XCC 40
1962.00 CSR MOVE FRLVAL L@XCC 40
1963.00 CSR MOVE FRUVAL U@XCC 40
1964.00 CSR MOVE FREDWR W@XCC 30
1965.00 CSR MOVE FRLR J@XCC 1
1966.00 CSR MOVE FRNNIX N@XCC 20
1967.00 CSR Z-ADD1 #@XCC 110
1968.00 CSR MOVE F@XCC #A
1969.00 CSR DO #A
1970.00 CSR MULT 10 #@XCC
1971.00 CSR END
1972.00 CSR END
1973.00 C*-----
1974.00 C*
1975.00 C* Dictionary parameters for - Description
1976.00 C*
1977.00 CSR MOVE *BLANK FRDTAI
1978.00 CSR MOVEL'XDS' FRDTAI
1979.00 CSR CALL 'X9800E' 81
1980.00 C* -----
1981.00 CSR PARM I9800E
1982.00 CSR FRERR IPEQ '0'
1983.00 CSR MOVE FRDSCR B@XDS 40
1984.00 CSR MOVE FRDTAT T@XDS 1
1985.00 CSR MOVE FREC E@XDS 1
1986.00 CSR MOVE FRDTAS C@XDS 50
1987.00 CSR MOVE FRDTAD G@XDS 20
1988.00 CSR MOVE FRCDEC F@XDS 1
1989.00 CSR MOVE LFRSY S@XDS 4
1990.00 CSR MOVE FRRT R@XDS 2
1991.00 CSR MOVE FRDVAL D@XDS 40
1992.00 CSR MOVE FRVAL A@XDS 40
1993.00 CSR MOVE FRLVAL L@XDS 40
1994.00 CSR MOVE FRUVAL U@XDS 40
1995.00 CSR MOVE FREDWR W@XDS 30
1996.00 CSR MOVE FRLR J@XDS 1
1997.00 CSR MOVE FRNNIX N@XDS 20
1998.00 CSR Z-1DD1 #@XDS 110
1999.00 CSR MOVE F@XDS #A
2000.00 CSR DO #A
2001.00 CSR MULT 10 #@XDS
2002.00 CSR END
2003.00 CSR END
2004.00 C*-----
2005.00 C*

```

Forces clear of everything before processing next record. Simulates user pressing the *Clear Screen* function key.

Retrieves all of the Data Dictionary editing parameters for necessary data items used in the program and moves the information into constant fields

Data Dictionary file server


```

2006.00 C* Dictionary parameters for - Date Last Ship
2007.00 C*
2008.00 CSR MOVE *BLANK FRDTAI
2009.00 CSR MOVE 'XDT' FRDTAI
2010.00 CSR CALL 'X9800E' 81
2011.00 C* -----
2012.00 CSR PARM I9800E
2013.00 CSR FRERR IFRQ '0'
2014.00 CSR MOVE FRDSCR B@XDT 40
2015.00 CSR MOVE FRDTAT T@XDT 1
2016.00 CSR MOVE FREC E@XDT 1
2017.00 CSR MOVE FRDTAS C@XDT 50
2018.00 CSR MOVE PROTAD G@XDT 20
2019.00 CSR MOVE FRCDEC F@XDT 1
2020.00 CSR MOVE LFRSY S@XDT 4
2021.00 CSR MOVE FRRT R@XDT 2
2022.00 CSR MOVE FRDVAL D@XDT 40
2023.00 CSR MOVE FRVAL A@XDT 40
2024.00 CSR MOVE FRLVAL L@XDT 40
2025.00 CSR MOVE FRUVAL U@XDT 40
2026.00 CSR MOVE FREDWR W@XDT 30
2027.00 CSR MOVE FRLR J@XDT 1
2028.00 CSR MOVE FRNNIX N@XDT 20
2029.00 CSR Z-ADD1 #@XDT 110
2030.00 CSR MOVE F@XDT #A
2031.00 CSR DO #A
2032.00 CSR MULT 10 #@XDT
2033.00 CSR END
2034.00 CSR END
2035.00 C*-----
2036.00 C*
2037.00 C* Dictionary parameters for - Item ID
2038.00 C*
2039.00 CSR MOVE *BLANK FRDTAI
2040.00 CSR MOVE 'XIT' FRDTAI
2041.00 CSR CALL 'X9800E' 81
2042.00 C* -----
2043.00 CSR PARM I9800E
2044.00 CSR FRERR IFEQ '0'
2045.00 CSR MOVE FRDSCR B@XIT 40
2046.00 CSR MOVE FRDTAT T@XIT 1
2047.00 CSR MOVE FREC E@XIT 1
2048.00 CSR MOVE FRDTAS C@XIT 50
2049.00 CSR MOVE FRDTAD G@XIT 20
2050.00 CSR MOVE FRCDEC F@XIT 1
2051.00 CSR MOVE LFRSY S@XIT 4
2052.00 CSR MOVE FRRT R@XIT 2
2053.00 CSR MOVE FRDVAL ft 40
2054.00 CSR MOVE FRVAL A@XIT 40
2055.00 CSR MOVE FRLVAL L@XIT 40
2056.00 CSR MOVE FRUVAL U@XIT 40
2057.00 CSR MOVE FREDWR W@XIT 30
2058.00 CSR MOVE FRLR J@XIT 1
2059.00 CSR MOVE FRNNIX N@XIT 20
2060.00 CSR Z-1DD1 #@XIT 110
2061.00 CSR MOVE F@XIT #A
2062.00 CSR DO #A
2063.00 CSR MULT 10 #@XIT
2064.00 CSR END
2065.00 CSR END
2066.00 C*-----
2067.00 C*
2068.00 C* Dictionary parameters for - gnanity On Hand
2069.00 C*
2070.00 CSR MOVE *BLANK FRDTAI
2071.00 CSR MOVE 'XQT' FRDTAI
2072.00 CSR CALL 'X9800E' 81
2073.00 C* -----
2074.00 CSR PARM I9800E
2075.00 CSR FRERR IFEQ '0'
2076.00 CSR MOVE FRDSCR B@XQT 40
2077.00 CSR MOVE FRDTAT T@XQT 1
2078.00 CSR MOVE FREC E@XQT 1
2079.00 CSR MOVE FRDTAS C@XQT 50
2080.00 CSR MOVE FRDTAD G@XQT 20
2081.00 CSR MOVE FRCDEC F@XQT 1
2082.00 CSR MOVE LFRSY @SXQT 4

```

2083.00	CSR	MOVE PRRT	RXQT	2
2084.00	CSR	MOVE PRDVAL	D@XQT	40
2085.00	CSR	MOVE PRVAL	A@XQT	40
2086.00	CSR	MOVE PRLVAL	L@XQT	40
2087.00	CSR	MOVE PRUVAL	U@XQT	40
2088.00	CSR	MOVE PREDWR	W@XQT	30
2089.00	CSR	MOVE PRLR	J@XQT	1
2090.00	CSR	MOVE PRNNIX	N@XQT	20
2091.00	CSR	Z-ADD1	#@XQT	110
2092.00	CSR	MOVE P@XQT	#A	
2093.00	CSR	DO #A		
2094.00	CSR	MULT 10	#@XQT	
2095.00	CSR	END		
2096.00	CSR	END		
2097.00	C*	-----		
2098.00	C*			
2099.00	C*	Dictionary parameters for - Item Type		
2100.00	C*			
2101.00	CSR	MOVE *BLANK	FRDTAI	
2102.00	CSR	MOVE L'XTY'	FRDTAI	
2103.00	CSR	CALL 'X9800E'		81
2104.00	C*	-----		
2105.00	CSR	PARM	I9800E	
2106.00	CSR	FRERR IPBQ '0'		
2107.00	CSR	MOVE PRDSCR	B@XTY	40
2108.00	CSR	MOVE PRDTAT	T@XTY	1
2109.00	CSR	MOVE PREC	E@XTY	1
2110.00	CSR	MOVE PRDTAS	C@XTY	50
2111.00	CSR	MOVE PRDTAT	G@XTY	20
2112.00	CSR	MOVE PRCDEC	F@XTY	1
2113.00	CSR	MOVE LPRS	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 PREDWR	W@XTY	30
2120.00	CSR	MOVE PRLR	J@XTY	1
2121.00	CSR	MOVE PRNNIX	N@XTY	20
2122.00	CSR	Z-ADD1	#@XTY	110
2123.00	CSR	MOVE P@XTY	#A	
2124.00	CSR	DO #A		
2125.00	CSR	MULT 10	#@XTY	
2126.00	CSR	END		
2127.00	CSR	END		
2128.00	C*	-----		
2129.00	C*			
2130.00	C*	Dictionary parameters for - Item Unit of Measure		
2131.00	C*			
2132.00	CSR	MOVE *BLANK	FRDTAI	
2133.00	CSR	MOVE L'XUM'	FRDTAI	
2134.00	CSR	CALL 'X9800E'		81
2135.00	C*	-----		
2136.00	CSR	PARM	I9800E	
2137.00	CSR	FRERR IPBQ '0'		
2138.00	CSR	MOVE PRDSCR	B@XUM	40
2139.00	CSR	MOVE PRDTAT	T@XUM	1
2140.00	CSR	MOVE PREC	E@XUM	1
2141.00	CSR	MOVE PRDTAS	C@XUM	50
2142.00	CSR	MOVE PRDTAD	G@XUM	20
2143.00	CSR	MOVE PRCDEC	F@XUM	1
2144.00	CSR	MOVE LPRS	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 PREDWR	W@XUM	30
2151.00	CSR	MOVE PRLR	J@XUM	1
2152.00	CSR	MOVE PRNNIX	N@XUM	20
2153.00	CSR	Z-ADD1	#@XUM	110
2154.00	CSR	MOVE P@XUM	#A	
2155.00	CSR	DO #A		
2156.00	CSR	MULT 10	#@XUM	
2157.00	CSR	END		
2158.00	CSR	END		
2159.00	C*	-----		

```

2160.00 C*
2161.00 C* Dictionary parameters for - Item Category Code 001
2162.00 C*
2163.00 CSR MOVE *BLANK FRDTAI
2164.00 CSR MOVE 'X001' FRDTAI
2165.00 CSR CALL 'X9800E' 81
2166.00 C*
2167.00 CSR PARM I9800E
2168.00 CSR FRERR IFEQ '0'
2169.00 CSR MOVE FRDSCR B@X001 40
2170.00 CSR MOVE FRDTAT T@X001 1
2171.00 CSR MOVE FREC E@X001 1
2172.00 CSR MOVE FRDTAS C@X001 50
2173.00 CSR MOVE FRDTAD G@X001 20
2174.00 CSR MOVE FRCDEC F@X001 1
2175.00 CSR MOVE LFRST S@X001 4
2176.00 CSR MOVE FRRT R@X001 2
2177.00 CSR MOVE FRDVAL D@X001 40
2178.00 CSR MOVE FRVAL A@X001 40
2179.00 CSR MOVE FRLVAL L@X001 40
2180.00 CSR MOVE FRDVAL U@X001 40
2181.00 CSR MOVE FREDWR W@X001 30
2182.00 CSR MOVE FRLR J@X001 1
2183.00 CSR MOVE FRNNIX N@X001 20
2184.00 CSR Z-ADD1 #@X001 110
2185.00 CSR MOVE F@X001 #A
2186.00 CSR DO #A
2187.00 CSR MULT 10 #@X001
2188.00 CSR END
2189.00 CSR END
2190.00 C*
2191.00 C*
2192.00 C* Dictionary parameters for - Item Category Code 002
2193.00 C*
2194.00 CSR MOVE *BLANK FRDTAI
2195.00 CSR MOVE 'X002' FRDTAI
2196.00 CSR CALL 'X9800E' 81
2197.00 C*
2198.00 CSR PARM I9800E
2199.00 CSR FRERR IFEQ '0'
2200.00 CSR MOVE FRDSCR B@X002 40
2201.00 CSR MOVE FRDTAT T@X002 1
2202.00 CSR MOVE FREC E@X002 1
2203.00 CSR MOVE FRDTAS C@X002 50
2204.00 CSR MOVE FRDTAD G@X002 20
2205.00 CSR MOVE FRCDEC F@X002 1
2206.00 CSR MOVE LFRST S@X002 4
2207.00 CSR MOVE FRRT R@X002 2
2208.00 CSR MOVE FRDVAL D@X002 40
2209.00 CSR MOVE FRVAL A@X002 40
2210.00 CSR MOVE FRLVAL L@X002 40
2211.00 CSR MOVE FRDVAL U@X002 40
2212.00 CSR MOVE FREDWR W@X002 30
2213.00 CSR MOVE FRLR J@X002 1
2214.00 CSR MOVE FRNNIX N@X002 20
2215.00 CSR Z-ADD1 #@X002 110
2216.00 CSR MOVE F@X002 #A
2217.00 CSR DO #A
2218.00 CSR MULT 10 #@X002
2219.00 CSR END
2220.00 CSR END
2221.00 C*
2222.00 C*
2223.00 C* Dictionary parameters for - Item Category Code 003
2224.00 C*
2225.00 CSR MOVE *BLANK FRDTAI
2226.00 CSR MOVE 'X003' FRDTAI
2227.00 CSR CALL 'X9800E' 81
2228.00 C*
2229.00 CSR PARM I9800E
2230.00 CSR FRERR IFEQ '0'
2231.00 CSR MOVE FRDSCR B@X003 40
2232.00 CSR MOVE FRDTAT T@X003 1
2233.00 CSR MOVE FREC E@X003 1
2234.00 CSR MOVE FRDTAS C@X003 50
2235.00 CSR MOVE FRDTAD G@X003 20
2236.00 CSR MOVE FRCDEC F@X003 1

```

2237.00	CSR	MOVELPRS	S@X003	4
2238.00	CSR	MOVE FRRT	R@X003	2
2239.00	CSR	MOVE PRDVAL	D@X003	40
2240.00	CSR	MOVE PRVAL	A@X003	40
2241.00	CSR	MOVE PRLVAL	L@X003	40
2242.00	CSR	MOVE PRDVAL	U@X003	40
2243.00	CSR	MOVE PREDWR	W@X003	30
2244.00	CSR	MOVE PRLR	J@X003	1
2245.00	CSR	MOVE PRNNIX	N@X003	20
2246.00	CSR	Z-ADD1	#@X003	110
2247.00	CSR	MOVE P@X003	#A	
2248.00	CSR	DO #A		
2249.00	CSR	MULT 10	#@X003	
2250.00	CSR	END		
2251.00	CSR	END		
2252.00	C*	-----		
2253.00	C*			
2254.00	C*	Dictionary parameters for - Item Category Code 004		
2255.00	C*			
2256.00	CSR	MOVE *BLANK	FRDTAI	
2257.00	CSR	MOVE 'X004'	FRDTAI	
2258.00	CSR	CALL 'X9800E'		81
2259.00	C*	-----		
2260.00	CSR	FRERR PARM	I9800E	
2261.00	CSR	IFBQ '0'		
2262.00	CSR	MOVE PRDSCR	B@X004	40
2263.00	CSR	MOVE PRDTAT	T@X004	1
2264.00	CSR	MOVE PREC	E@X004	1
2265.00	CSR	MOVE PRDTAS	C@X004	50
2266.00	CSR	MOVE PRDTAD	G@X004	20
2267.00	CSR	MOVE PRCDEC	F@X004	1
2268.00	CSR	MOVELPRS	S@X004	4
2269.00	CSR	MOVE FRRT	R@X004	2
2270.00	CSR	MOVE PRDVAL	D@X004	40
2271.00	CSR	MOVE PRVAL	A@X004	40
2272.00	CSR	MOVE PRLVAL	L@X004	40
2273.00	CSR	MOVE PRDVAL	U@X004	40
2274.00	CSR	MOVE PREDWR	W@X004	30
2275.00	CSR	MOVE PRLR	J@X004	1
2276.00	CSR	MOVE PRNNIX	N@X004	20
2277.00	CSR	Z-ADD1	#@X004	110
2278.00	CSR	MOVE P@X004	#A	
2279.00	CSR	DO #A		
2280.00	CSR	MULT 10	#@X004	
2281.00	CSR	END		
2282.00	CSR	END		
2283.00	C*	-----		
2284.00	C*			
2285.00	C*	Dictionary parameters for - Item Category Code 005		
2286.00	C*			
2287.00	CSR	MOVE *BLANK	FRDTAI	
2288.00	CSR	MOVE 'X005'	FRDTAI	
2289.00	CSR	CALL 'X9800E'		81
2290.00	C*	-----		
2291.00	CSR	FRERR PARM	I9800E	
2292.00	CSR	IFBQ '0'		
2293.00	CSR	MOVE PRDSCR	B@X005	40
2294.00	CSR	MOVE PRDTAT	T@X005	1
2295.00	CSR	MOVE PREC	E@X005	1
2296.00	CSR	MOVE PRDTAS	C@X005	50
2297.00	CSR	MOVE PRDTAD	G@X005	20
2298.00	CSR	MOVE PRCDEC	F@X005	1
2299.00	CSR	MOVELPRS	S@X005	4
2300.00	CSR	MOVE FRRT	R@X005	2
2301.00	CSR	MOVE PRDVAL	D@X005	40
2302.00	CSR	MOVE PRVAL	A@X005	40
2303.00	CSR	MOVE PRLVAL	L@X005	40
2304.00	CSR	MOVE PRDVAL	U@X005	40
2305.00	CSR	MOVE PREDWR	W@X005	30
2306.00	CSR	MOVE PRLR	J@X005	1
2307.00	CSR	MOVE PRNNIX	N@X005	20
2308.00	CSR	Z-ADD1	#@X005	110
2309.00	CSR	MOVE P@X005	#A	
2310.00	CSR	DO #A		
2311.00	CSR	MULT 10	#@X005	
2312.00	CSR	END		
2313.00	CSR	END		

```

2314.00 C*
2315.00 C*
2316.00 C* Set subroutine execution flag.
2317.00 C*
2318.00 CSR MOVE '1' $998 1
2319.00 C*
2320.00 CSR END998 ENDSR
2321.00 C*****
2322.00 C*
2323.00 C* SUBROUTINE S999- Housekeeping
2324.00 C*
2325.00 C*
2326.00 C* Processing: 1. Load video screen text.
2327.00 C* 2. Retrieve screen title data area, test
2328.00 C* for unauthorized access, center video
2329.00 C* title and save to video screen.
2330.00 C* 3. Initialize key list.
2331.00 C* 4. Load roll keys.
2332.00 C* 5. Passed parameters.
2333.00 C* 6. Load error message array.
2334.00 C*
2335.00 CSR S999 BEGSR
2336.00 C* ----
2337.00 C*
2338.00 C* Required program parameters.
2339.00 C*
2340.00 CSR *ENTRY PLIST
2341.00 C*
2342.00 C* Passed Parameter - Item ID
2343.00 C*
2344.00 CSR PARM $$XIT 8
2345.00 C*
2346.00 C* Move to internal reference - Item ID
2347.00 C*
2348.00 CSR MOVE ##XIT VDXIT
2349.00 C*
2350.00 C* Test for auto inquiry function.
2351.00 C*
2352.00 CSR VDXIT IFNE *BLANK
2353.00 CSR MOVE '1' $AUTO 1
2354.00 CSR END
2355.00 C*-----
2356.00 C*
2357.00 C* Load video screen text
2358.00 C*
2359.00 CSR MOVE@@FILE PSKEY 10
2360.00 CSR Z-ADD025 P$VTX# 30
2361.00 C/COPY JDECPY,C008C
2362.00 C*-----
2363.00 C*
2364.00 C* Key list for - Cost Center Security
2365.00 C*
2366.00 CSR MSKY01 KLIST
2367.00 CSR KFLD MSUSER
2368.00 CSR KFLD MSFILE
2369.00 CSR KFLD MSMCUT
2370.00 C*-----
2371.00 C*
2372.00 C* Key list for -SDM Item Master File
2373.00 C*
2374.00 CSR ZXKY01 KLIST
2375.00 CSR KFLD QXXIT
2376.00 C*-----
2377.00 C*
2378.00 C* Load roll key uppr and lower key values.
2379.00 C*
2380.00 CSR *LIKE DEFM QXXIT $RUKEY
2381.00 CSR *LIKE DEFN $RUKEY $RDKEY
2382.00 CSR MOVE *LOVAL $RUKEY
2383.00 CSR MOVE *ALL'9' $RDKEY
2384.00 C*-----
2385.00 C*

```

Assures S998 will only be executed once

Parameters passed to program

Set auto-inquiry if information is passed

Retrieves vocabulary overrides

Only loads these VTX fields displayed on the video instead of all 144

Composite keys are defined here

Using *LIKE more and more, especially for work fields.

```

2386.00 C*          Load error messages array.
2387.00 C*
2388.00 CSR          MOVE '0001'          EMK,01      Inv Action
2389.00 CSR          MOVE '0002'          EMK,02      Inv Key
2390.00 CSR          MOVE '0003'          EMK,03      Inv Blanks
2391.00 CSR          MOVE '0004'          EMK,04      Inv Date
2392.00 CSR          MOVE '0005'          EMK,05      Inv Next Nbr
2393.00 CSR          MOVE '0007'          EMK,06      In Use
2394.00 CSR          MOVE '0025'          EMK,07      Inv Values
2395.00 CSR          MOVE '0026'          EMK,08      Inv MCU
2396.00 CSR          MOVE '0027'          EMK,09      Inv Desc Ttl
2397.00 CSR          MOVE '0052'          EMK,10
-----
2398.00 C*
2399.00 C*
2400.00 C*          Load invalid action code array.
2401.00 C*
2402.00 CSR          MOVEA ' '          @NAC
-----
2403.00 C*
2404.00 C*
2405.00 Ct          Load system date.
2406.00 C*
2407.00 CSR          TIME          $WRK12 120
2408.00 CSR          MOVE $WRK12      $$EDT      60
2409.00 CSR          MOVE $$EDT      $SIDAT      6
2410.00 CSR          MOVEVL ' *SYSVAL ' #PFMT      7
2411.00 CSR          MOVEVL*BLANKS #EDAT      8
2412.00 CSR          MOVEVL ' *JUL ' #TFMT      7
2413.00 CSR          MOVEVL ' *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      $$UFMJ      60
-----
2424.00 C*
2425.00 CSR          END999          ENDSR
2426.00 C*****
2427.00 C*****
2428.00 C*****
0I92801 E          UNLOCK
-----

```

Error message numbers from Data Dictionary

Lockout action code function used with the Program Generator

Use the TIME feature to allow for all date formats

Method of releasing master file record locks

Appendix E - JD Edwards World Subroutines and Flows

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:

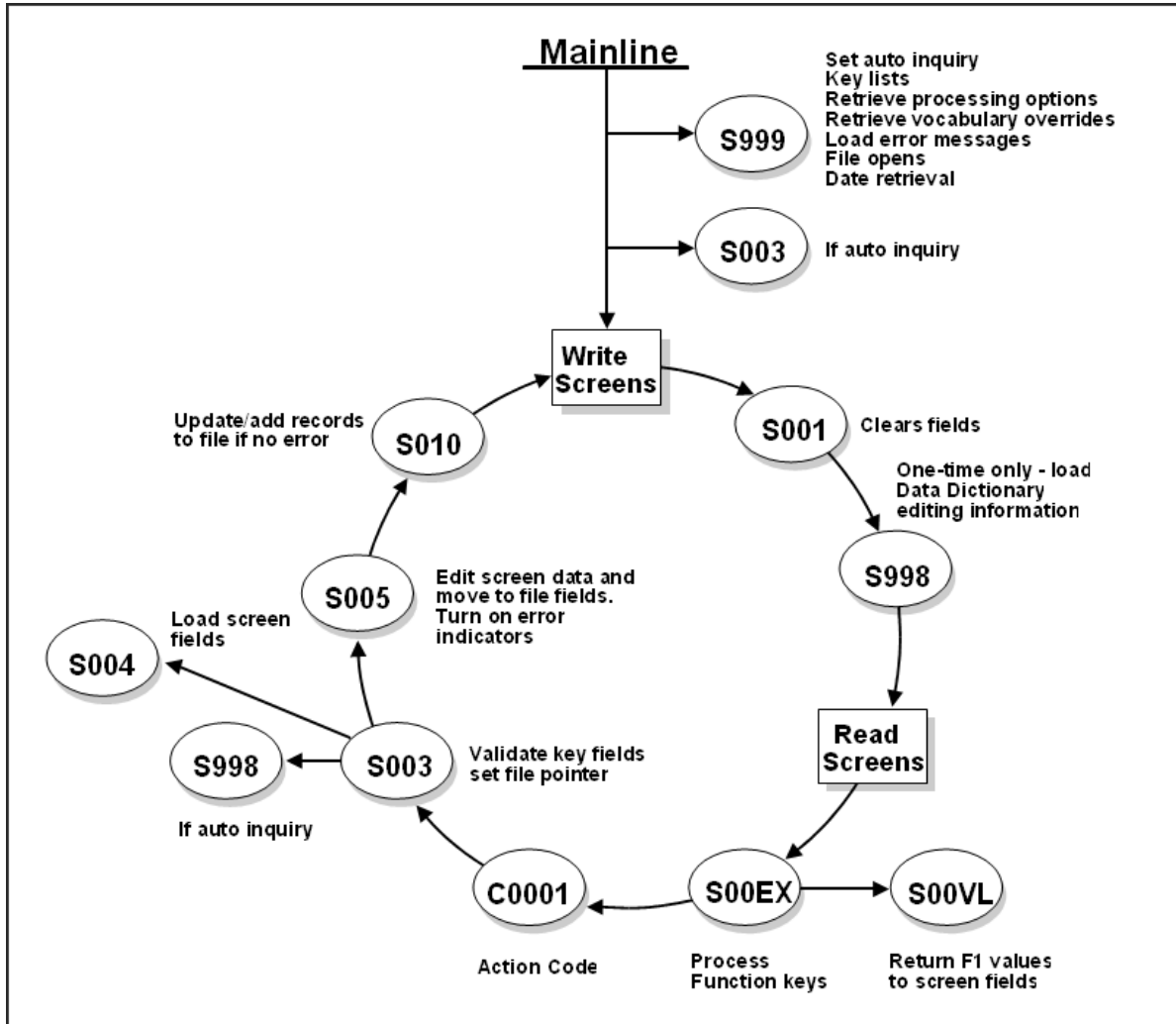
Subroutine	Description
S00EX	Processes all function exits. Choose a function exit and the system launches one of the following programs: <ul style="list-style-type: none">▪ Display Functions (F24) to launch P9601H▪ Field Sensitive Help (F1) launches X96CCX. After X96CCX launches, the system launches 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	<p>Display and load the screen or report fields.</p>
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	<p>Print Report Subheadings.</p>
S998	<p>Loads Data Dictionary values, one time only.</p> <p>Retrieves row description for level breaks and subheadings, if applicable.</p>

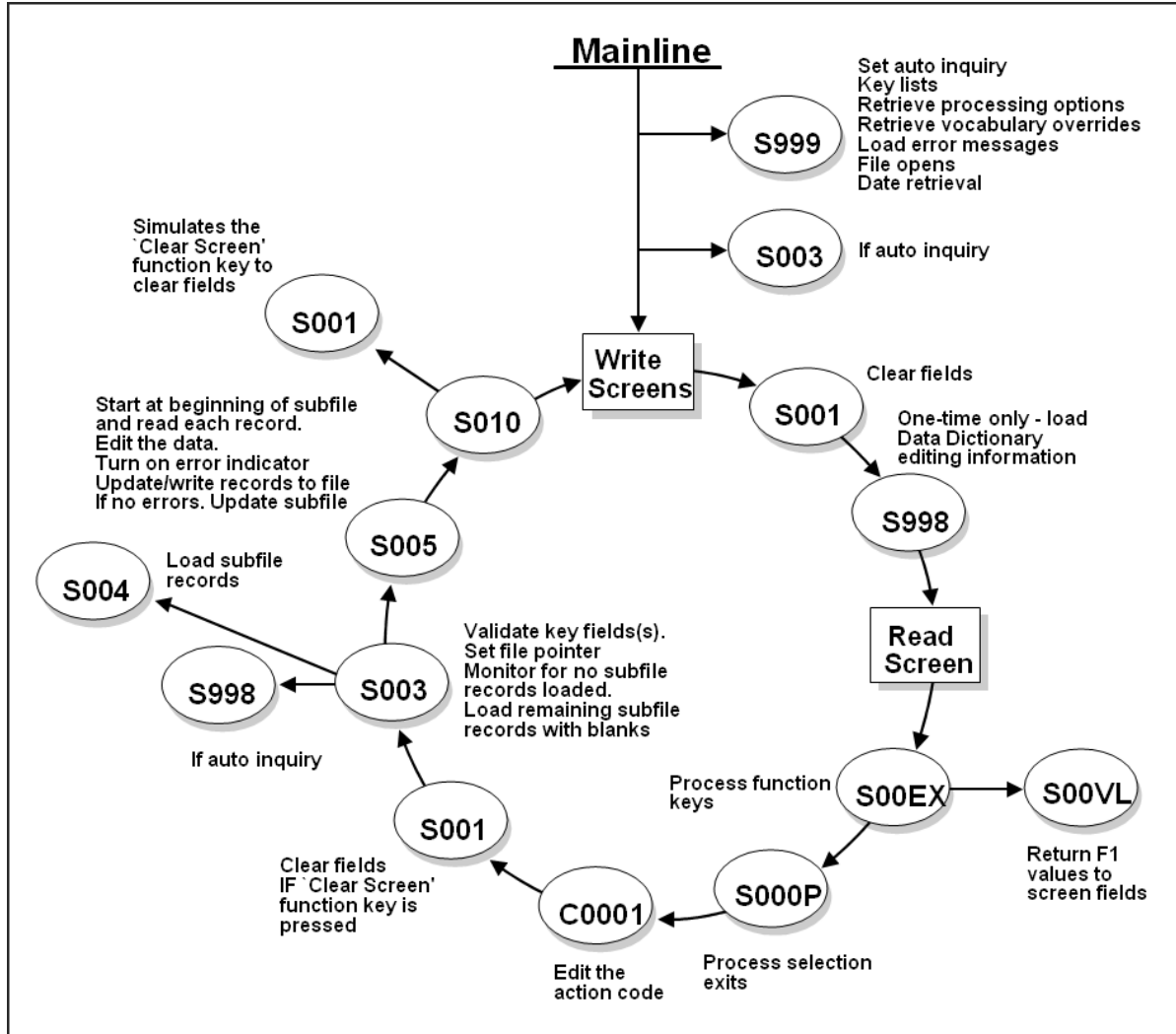
Subroutine	Description
S999	Housekeeping, one time only. <ul style="list-style-type: none">▪ Sets auto inquiry▪ Defines key lists▪ Retrieves processing options and level breaks, if applicable▪ Retrieves vocabulary overrides▪ Loads error messages▪ Opens file▪ Retrieves the current date▪ Defines work fields using *LIKE▪ Prints cover page and Helps in a report

Flows

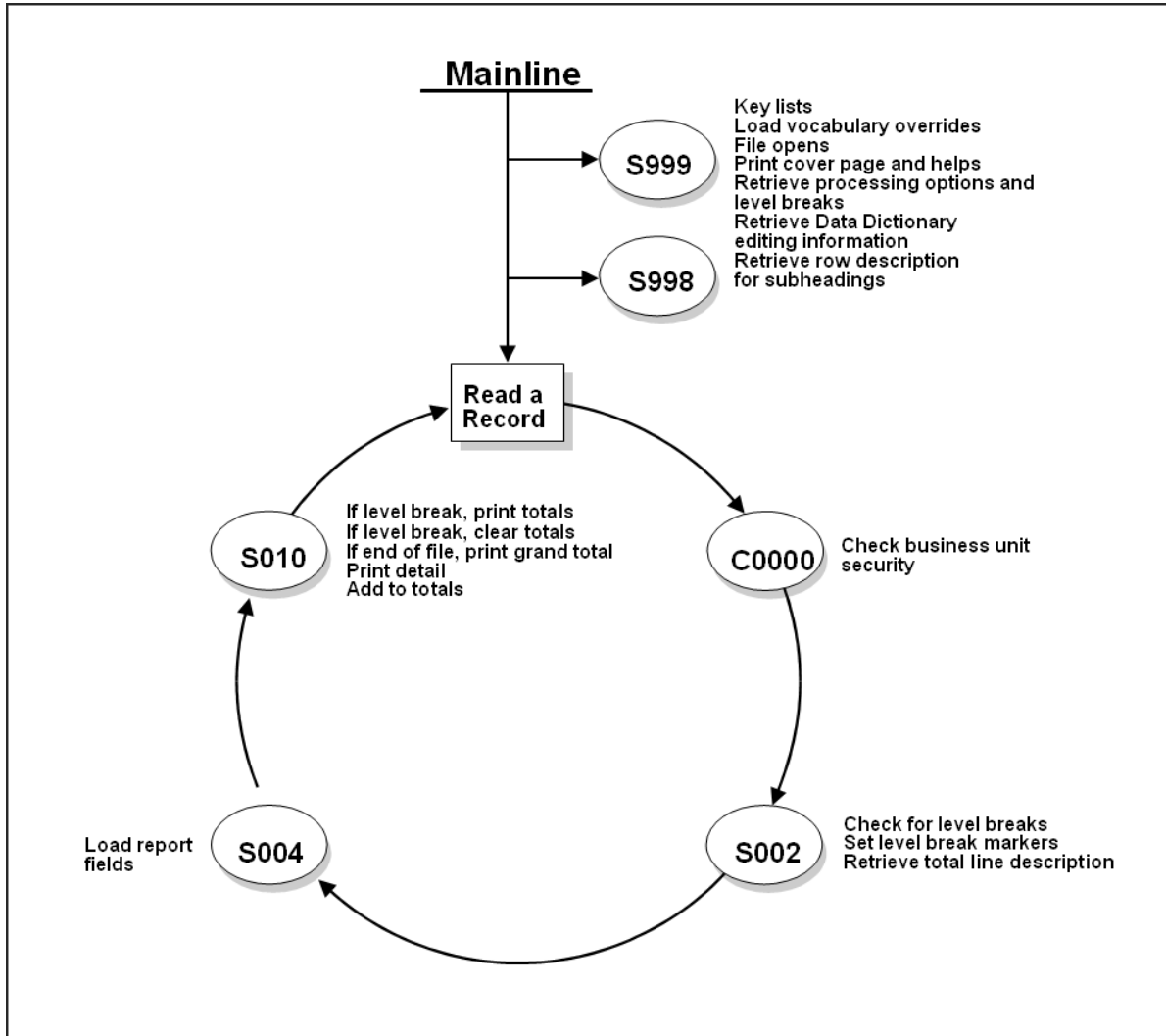
Interactive Non-Subfile Program



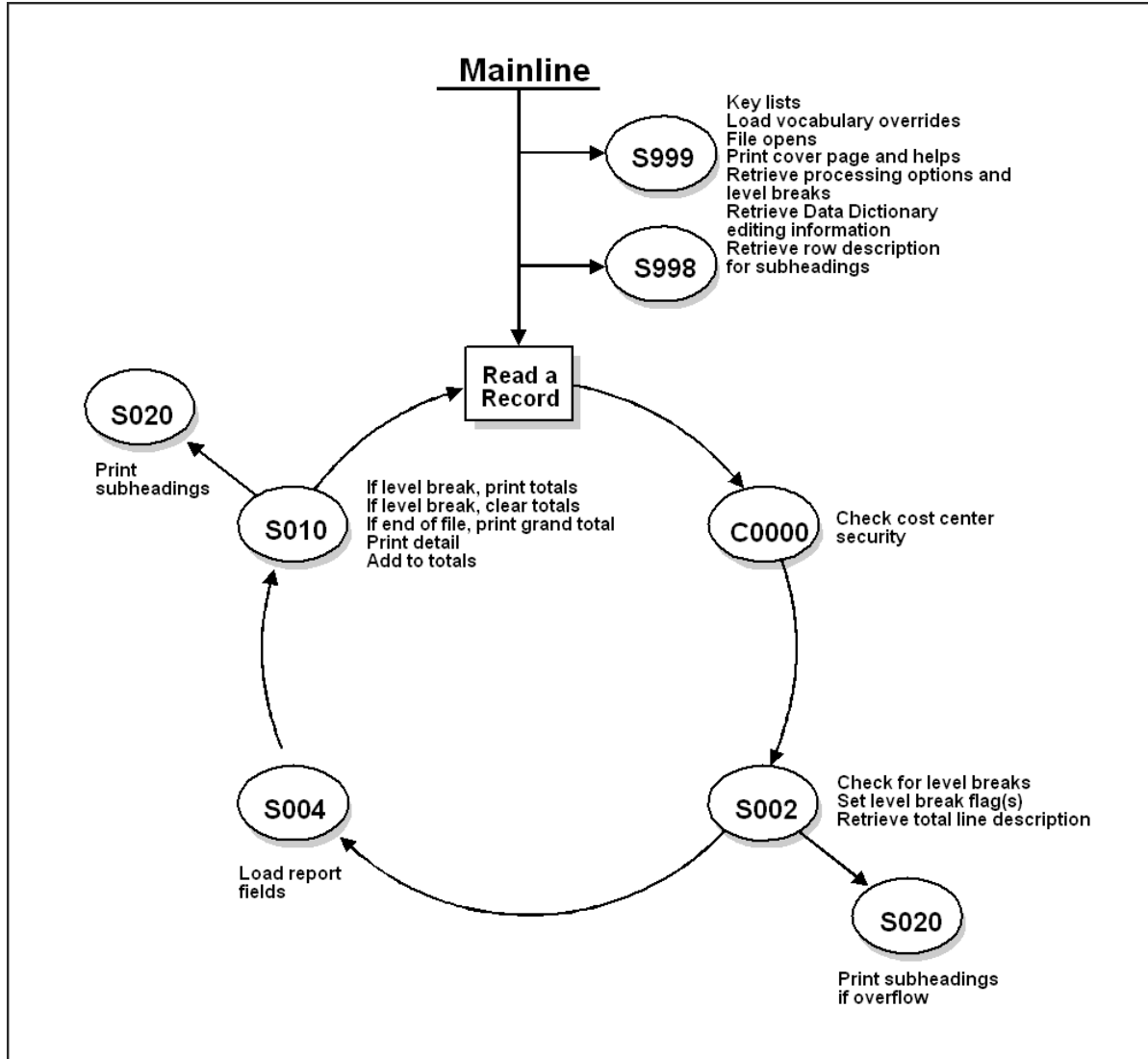
Subfile Program with Options



Report Program without Subheadings

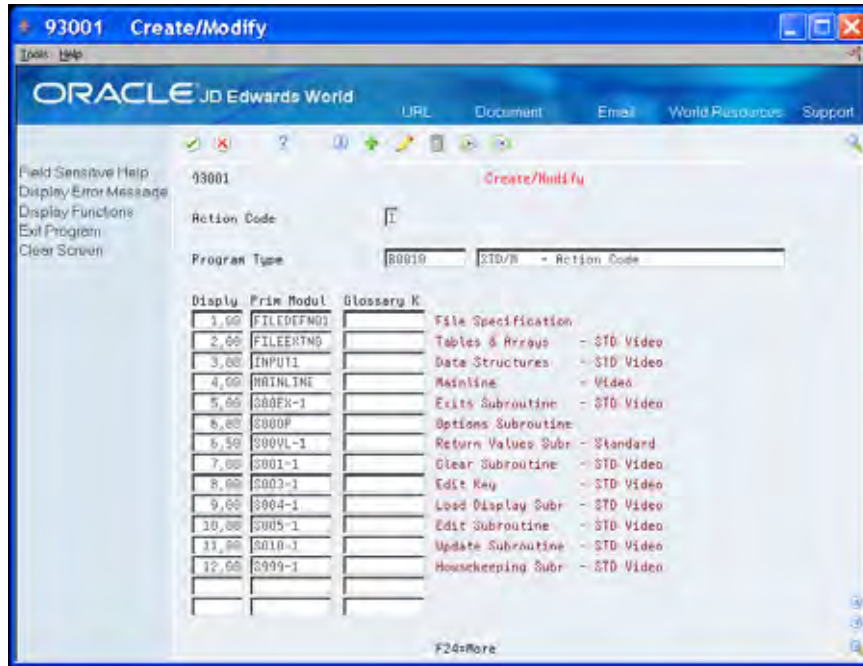


Report Program with Subheadings



Appendix F – Sample Code

Following is the code to create the basic shell for program type B0010.



```

R93950          B0010      - STD/M      - Action Code          DATE - 2/02/17
TITLEH/TITLE
H* -----
H*
H*      Copyright (c) 2007
H*      JD Edwards World
H*
H*      This unpublished material is proprietary to
H*      JD Edwards World. All rights reserved.
H*      The methods and techniques described herein are
H*      considered trade secrets and/or confidential.
H*      Reproduction or distribution, in whole or in part,
H*      is forbidden except by express written permission
H*      of JD Edwards World.
H*
H* -----
F*
F*      PROGRAM REVISION LOG
F* -----
F*
F*      Date      Programmer      Nature of Revision
F* -----
AUTHRF*          SAR #          (AS/400  A/G)
F*
DESC F*
F*
F* *****
F*
FILESP*
COPY F* *****
E*
E*      PROGRAM TABLES AND ARRAYS
E* -----
E*
E*          EMK          64  4          Error Msg
E*          @MK          64  1          Error Msg
  
```

Appendix F – Sample Code

E	@ER	64	4	Error Msg	FILEEXTNO	007000000000	
E	@DV	40	1	Dflt Wrk	FILEEXTNO	008000000000	
E	@C	256	1	Literal Work	FILEEXTNO	011100000000	
COPY	E*				FILEEXTNO	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	
R93950		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	
R93950		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	

VTX I		32813320	VTX083	INPUT1	090000000000
VTX I		33213360	VTX084	INPUT1	091000000000
VTX I		33613400	VTX085	INPUT1	092000000000
VTX I		34013440	VTX086	INPUT1	093000000000
VTX I		34413480	VTX087	INPUT1	094000000000
VTX I		34813520	VTX088	INPUT1	095000000000
VTX I		35213560	VTX089	INPUT1	096000000000
VTX I		35613600	VTX090	INPUT1	097000000000
VTX I		36013640	VTX091	INPUT1	098000000000
VTX I		36413680	VTX092	INPUT1	099000000000
VTX I		36813720	VTX093	INPUT1	100000000000
VTX I		37213760	VTX094	INPUT1	101000000000
VTX I		37613800	VTX095	INPUT1	102000000000
VTX I		38013840	VTX096	INPUT1	103000000000
VTX I		38413880	VTX097	INPUT1	104000000000
VTX I		38813920	VTX098	INPUT1	105000000000
VTX I		39213960	VTX099	INPUT1	106000000000
VTX I		39614000	VTX100	INPUT1	107000000000
VTX I		40014040	VTX101	INPUT1	108000000000
VTX I		40414080	VTX102	INPUT1	109000000000
VTX I		40814120	VTX103	INPUT1	110000000000
VTX I		41214160	VTX104	INPUT1	111000000000
VTX I		41614200	VTX105	INPUT1	112000000000
VTX I		42014240	VTX106	INPUT1	113000000000
VTX I		42414280	VTX107	INPUT1	114000000000
VTX I		42814320	VTX108	INPUT1	115000000000
VTX I		43214360	VTX109	INPUT1	116000000000
VTX I		43614400	VTX110	INPUT1	117000000000
VTX I		44014440	VTX111	INPUT1	118000000000
VTX I		44414480	VTX112	INPUT1	119000000000
VTX I		44814520	VTX113	INPUT1	120000000000
VTX I		45214560	VTX114	INPUT1	121000000000
VTX I		45614600	VTX115	INPUT1	122000000000
VTX I		46014640	VTX116	INPUT1	123000000000
VTX I		46414680	VTX117	INPUT1	124000000000
VTX I		46814720	VTX118	INPUT1	125000000000
VTX I		47214760	VTX119	INPUT1	126000000000
VTX I		47614800	VTX120	INPUT1	127000000000
VTX I		48014840	VTX121	INPUT1	128000000000
VTX I		48414880	VTX122	INPUT1	129000000000
VTX I		48814920	VTX123	INPUT1	130000000000
VTX I		49214960	VTX124	INPUT1	131000000000
VTX I		49615000	VTX125	INPUT1	132000000000
VTX I		50015040	VTX126	INPUT1	133000000000
VTX I		50415080	VTX127	INPUT1	134000000000
VTX I		50815120	VTX128	INPUT1	135000000000
VTX I		51215160	VTX129	INPUT1	136000000000
VTX I		51615200	VTX130	INPUT1	137000000000
VTX I		52015240	VTX131	INPUT1	138000000000
R93950	B0010	- STD/M	- Action Code	DATE	- 2/02/17
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,I00DSINX				INPUT1	153000000000
I/COPY JDECPY,I00PS@@				INPUT1	153100000000
INFDSI/COPY JDECPY,I00DSPROG				INPUT1	154000000000
DATESI*				INPUT1	155000000000
COPY I*				INPUT1	156000000000
C*****				MAINLINE	001000000000
C* MAINLINE PROGRAM				MAINLINE	002000000000
C* -----				MAINLINE	003000000000
C*				MAINLINE	004000000000
C* Process housekeeping.				MAINLINE	005000000000
C*				MAINLINE	006000000000
C EXSR S999				MAINLINE	007000000000
C* ----				MAINLINE	008000000000
C*				MAINLINE	009000000000
C* If LR on, end program.				MAINLINE	010000000000
C*				MAINLINE	011000000000
C *INLR CABEQ'1' EOJ				MAINLINE	012000000000
C* ----				MAINLINE	013000000000
C*				MAINLINE	014000000000
C* If automatic inquiry set, process inquiry.				MAINLINE	015000000000
C*				MAINLINE	016000000000
C \$AUTO CASEQ'1' S003 24				MAINLINE	017000000000
C* ----				MAINLINE	018000000000
C				MAINLINE	019000000000
C* END				MAINLINE	020000000000
C*				MAINLINE	021000000000
C* Begin normal program processing.				MAINLINE	021000000000
C* -----				MAINLINE	022000000000
C*				MAINLINE	023000000000
C *INLR DOWEQ'0'				MAINLINE	024000000000
/*				MAINLINE	025000000000
/* If #SFRNO field, do subfile record number validation				MAINLINE	026000000000

/*					MAINLINE	027000000000
+FLDNC*	#SFRNO	ZSFLRCDNO			MAINLINE	028000000000
/*					MAINLINE	029000000000
/*	If SFLCLR is used, process *in38 accordingly				MAINLINE	030000000000
/*					MAINLINE	031000000000
+FLDNC*	?SFLCLR	ZSFLCLR			MAINLINE	032000000000
C*					MAINLINE	033000000000
C*	Write video screen.				MAINLINE	034000000000
C*					MAINLINE	035000000000
/*					MAINLINE	036000000000
/*	If not a subfile display, just write format1				MAINLINE	037000000000
/*					MAINLINE	038000000000
-FLDNC*	?SFL	ZWRITE			MAINLINE	039000000000
/*					MAINLINE	040000000000
R93950		B0010	- STD/M	- Action Code		DATE - 2/02/17
/*	If a subfile display, write format1 and formatc				MAINLINE	041000000000
/*					MAINLINE	042000000000
+FLDNC*	?SFL	ZWRITESFL			MAINLINE	043000000000
C		MOVE '1'	@@AID		MAINLINE	044000000000
C		EXSR S001			MAINLINE	045000000000
C*		----			MAINLINE	046000000000
C*					MAINLINE	047000000000
C*	Load data field dictionary parameters (one cycle only).				MAINLINE	048000000000
C*					MAINLINE	049000000000
C	\$998	CASEQ' '	S998		MAINLINE	050000000000
C*		----	----		MAINLINE	051000000000
C		END			MAINLINE	052000000000
C*					MAINLINE	053000000000
C*	Begin video screen read processing.				MAINLINE	054000000000
C*					MAINLINE	055000000000
C		SETOF		999301	MAINLINE	056000000000
DSPF C		READ &01FILE		9998	MAINLINE	057000000000
C		Z-ADD0	##RROW		MAINLINE	058000000000
C		Z-ADD0	##RCOL		MAINLINE	059000000000
C*					MAINLINE	060000000000
C*	If video read timed out, end program.				MAINLINE	061000000000
C*					MAINLINE	062000000000
C	*IN99	CABEQ'1'	EOJ	LR	MAINLINE	063000000000
C*		----	----		MAINLINE	064000000000
C	@@AID	CABEQ#FEOJ	EOJ	LR	MAINLINE	065000000000
C*		----	----		MAINLINE	066000000000
C*					MAINLINE	067000000000
C*	If valid function key pressed, process and return.				MAINLINE	068000000000
C*					MAINLINE	069000000000
C	*IN15	IFEQ '1'			MAINLINE	070000000000
C		EXSR S00EX			MAINLINE	071000000000
C*		----			MAINLINE	072000000000
C	*INLR	CABEQ'1'	EOJ		MAINLINE	073000000000
C*		----	----		MAINLINE	074000000000
C	*IN15	CABEQ'1'	END		MAINLINE	075000000000
C*		----	----		MAINLINE	076000000000
C		END			MAINLINE	077000000000
/*					MAINLINE	078000000000
/*	If any selection exits, exsr S00OP				MAINLINE	079000000000
/*					MAINLINE	080000000000
+DTAIC*	SELC	ZS00OP			MAINLINE	081000000000
/*					MAINLINE	082000000000
/*	If action code then exsr C0001				MAINLINE	083000000000
/*					MAINLINE	084000000000
+FLDNC*	ACTION	ZACTION			MAINLINE	085000000000
C*					MAINLINE	086000000000
C*	Load subfile records.				MAINLINE	087000000000
C*					MAINLINE	088000000000
C		EXSR S003			MAINLINE	089000000000
C*		----			MAINLINE	090000000000
/*					MAINLINE	091000000000
/*	If any update files then exsr S005				MAINLINE	092000000000
/*					MAINLINE	093000000000
+FILEC*	*ANY	DB	ZS005	@	MAINLINE	094000000000
/*					MAINLINE	095000000000
/*	If any update files and action code then do S010				MAINLINE	096000000000
/*					MAINLINE	097000000000
+FILEC*	*ANY	DB	*AND	@	MAINLINE	098000000000
-FILEC*	*ANY	DB	*AND	2	MAINLINE	098500000000
R93950		B0010	- STD/M	- Action Code		DATE - 2/02/17
+FLDNC*	ACTION	ZS010A			MAINLINE	099000000000
/*					MAINLINE	100000000000
/*	If a Master File 2 exists, then do S011.				MAINLINE	101000000000
/*					MAINLINE	102000000000
+FILEC*	*ANY	DB	*AND	@	MAINLINE	103000000000
+FILEC*	*ANY	DB	*AND	2	MAINLINE	103500000000
+FLDNC*	ACTION	ZS011			MAINLINE	104000000000
C*					MAINLINE	105000000000
C*	Return for next input.				MAINLINE	106000000000
C*					MAINLINE	107000000000
C	END	TAG			MAINLINE	108000000000
C*		---	---		MAINLINE	109000000000
C*					MAINLINE	110000000000
C*	Set correct message in line 24.				MAINLINE	111000000000
C*					MAINLINE	112000000000
C	*IN93	IFEQ '1'			MAINLINE	113000000000
C		MOVESVL24E	VDL24		MAINLINE	114000000000
C		ELSE			MAINLINE	115000000000
C		MOVESVL24M	VDL24		MAINLINE	116000000000
C		END			MAINLINE	117000000000

```

C*                                     MAINLINE 118000000000
C                                     MAINLINE 119000000000
C*                                     MAINLINE 120000000000
C      EOJ      TAG      MAINLINE 121000000000
C*      ---      ---      MAINLINE 122000000000
C*                                     MAINLINE 123000000000
C*      END MAINLINE PROGRAM      MAINLINE 124000000000
C*      -----      MAINLINE 125000000000
COPY C*****      MAINLINE 126000000000
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      BEGSR      S00EX-1 005000000000
CSR      S00EX      BEGSR      S00EX-1 006000000000
C*      S00EX      BEGSR      S00EX-1 007000000000
+FLDNC*      #SFRNO      Z@@SRCN      S00EX-1 008000000000
CSR      T00EXA      TAG      S00EX-1 009000000000
C*      T00EXA      TAG      S00EX-1 009500000000
C*      If EOJ requested, exit subroutine.      S00EX-1 010000000000
C*      -----      S00EX-1 011000000000
CSR      @@AID      CABEQ#FEOJ      ENDEXE      LR      S00EX-1 012000000000
C*      -----      S00EX-1 013000000000
C*      If Display Keys pressed, exit to help facility and return.      S00EX-1 014000000000
C*      -----      S00EX-1 015000000000
CSR      @@AID      IFEQ #FKEYS      S00EX-1 016000000000
C*      CALL 'P9601H'      98      S00EX-1 017000000000
C*      -----      S00EX-1 018000000000
CSR      PARM      I00SC      S00EX-1 019000000000
CSR      PARM      SRVFDS      S00EX-1 020000000000
CSR      PARM      I00CSR      S00EX-1 021000000000
C*      PARM      I00CSR      S00EX-1 022000000000
CSR      @@AID      CABNE#FKEYS      T00EXA      S00EX-1 023000000000
C*      -----      S00EX-1 024000000000
R93950      B0010      - STD/M      - Action Code      S00EX-1 025000000000
CSR      GOTO ENDEXE      S00EX-1 026000000000
C*      -----      S00EX-1 027000000000
CSR      END      S00EX-1 028000000000
C*      If Cursor Sensitive Help Pressed, exit to CS Help.      S00EX-1 029000000000
C*      -----      S00EX-1 030000000000
CSR      @@AID      IFEQ #FQMRK      S00EX-1 031000000000
CSR      MOVEA*IN      ##IN      S00EX-1 032000000000
CSR      CALL 'X96CCX'      98      S00EX-1 033000000000
C*      -----      S00EX-1 034000000000
CSR      PARM      I00SC      S00EX-1 040000000000
CSR      PARM      SRVFDS      S00EX-1 041000000000
CSR      PARM      I00CSR      S00EX-1 042000000000
CSR      PARM ' '      ##CCFF 2      S00EX-1 043000000000
CSR      PARM      I00MDE      S00EX-1 044000000000
C*      PARM      I00MDE      S00EX-1 045000000000
CSR      ##FLDN      IFNE *BLANKS      S00EX-1 046000000000
CSR      EXSR S00VL      S00EX-1 047000000000
C*      -----      S00EX-1 048000000000
CSR      MOVEA##IN      *IN,1      S00EX-1 049000000000
CSR      END      S00EX-1 050000000000
CSR      MOVE#BLANKS      ##DTAI      S00EX-1 051000000000
CSR      GOTO ENDEXE      S00EX-1 052000000000
C*      -----      S00EX-1 053000000000
CSR      END      S00EX-1 054000000000
C*      If Display errors pressed, exit to error messages.      S00EX-1 055000000000
C*      -----      S00EX-1 056000000000
CSR      @@AID      IFEQ #FERRD      S00EX-1 057000000000
CSR      Z-ADD1      #G      S00EX-1 058000000000
CSR      Z-ADD1      #H      S00EX-1 059000000000
CSR      #G      DOWLE64      S00EX-1 060000000000
CSR      @MK,#G      IFEQ '1'      S00EX-1 061000000000
CSR      MOVE EMK,#G      @ER,#H      S00EX-1 062000000000
CSR      ADD 1      #H      S00EX-1 063000000000
CSR      END      S00EX-1 064000000000
CSR      ADD 1      #G      S00EX-1 065000000000
CSR      END      S00EX-1 066000000000
CSR      CALL 'P0000E'      98      S00EX-1 067000000000
C*      -----      S00EX-1 068000000000
CSR      PARM      @ER      S00EX-1 069000000000
CSR      GOTO ENDEXE      S00EX-1 070000000000
C*      -----      S00EX-1 071000000000
CSR      END      S00EX-1 072000000000
C*      If HELP key pressed, exit to help facility and return.      S00EX-1 073000000000
C*      -----      S00EX-1 074000000000
CSR      @@AID      IFEQ #FHHELP      S00EX-1 075000000000
CSR      CALL 'P00HELP'      98      S00EX-1 076000000000
C*      -----      S00EX-1 077000000000
CSR      PARM      HS@@      S00EX-1 078000000000
CSR      PARM      HE@@      S00EX-1 079000000000
CSR      PARM      HE@@      S00EX-1 080000000000
CSR      PARM      HE@@      S00EX-1 081000000000
CSR      PARM      HE@@      S00EX-1 082000000000
CSR      PARM      HE@@      S00EX-1 083000000000

```

CSR	PARM	I00SC		S00EX-1	084000000000
CSR	PARM	SRVFD5		S00EX-1	085000000000
CSR	PARM	I00CSR		S00EX-1	086000000000
CSR	GOTO ENDEXE			S00EX-1	087000000000
R93950	B0010	- STD/M	- Action Code		DATE - 2/02/17
C*	----	-----		S00EX-1	088000000000
CSR	END			S00EX-1	089000000000
C*				S00EX-1	090000000000
C*	If Clear screen pressed, clear screen and return.			S00EX-1	091000000000
C*	-----	-----		S00EX-1	092000000000
C*				S00EX-1	093000000000
CSR	@@AID	IFEQ #FCLR		S00EX-1	094000000000
CSR		EXSR S001		S00EX-1	095000000000
C*	----	-----		S00EX-1	096000000000
CSR	GOTO ENDEXE			S00EX-1	097000000000
C*	----	-----		S00EX-1	098000000000
EXITCCSR	END			S00EX-1	099000000000
C*				S00EX-1	100000000000
C*	Process roll up and down keys.			S00EX-1	101000000000
C*	-----	-----		S00EX-1	102000000000
C*				S00EX-1	103000000000
CSR	@@AID	IFEQ #FROLU		S00EX-1	104000000000
CSR	@@AID	OREQ #FROLD		S00EX-1	105000000000
CSR	\$SECUR	DOUEQ ' '		S00EX-1	107000000000
CSR		MOVE ' ' \$SECUR 1		S00EX-1	108000000000
C*				S00EX-1	109000000000
C*	If ROLL UP key pressed, process read next.			S00EX-1	110000000000
C*	-----	-----		S00EX-1	111000000000
C*				S00EX-1	112000000000
CSR	@@AID	IFEQ #FROLU		S00EX-1	113000000000
C*				S00EX-1	114000000000
C*	Reset error indicators if roll			S00EX-1	115000000000
C*				S00EX-1	116000000000
CSR		MOVE\$RESET *IN,41		S00EX-1	117000000000
CSR		MOVE '0' *IN,40		S00EX-1	118000000000
CSR		SETOF	818299	S00EX-1	119000000000
MF	%	READ &01FORMAT	9981	S00EX-1	120000000000
CSR	*IN81	IFEQ '1'		S00EX-1	121000000000
MF	\$RUKEY	SETLL&01FORMAT		S00EX-1	122000000000
CSR		SETOF	8299	S00EX-1	123000000000
MF	%	READ &01FORMAT	9982	S00EX-1	124000000000
C*				S00EX-1	125000000000
C*	If error on read, set error.			S00EX-1	126000000000
C*				S00EX-1	127000000000
CSR	*IN82	IFEQ '1'		S00EX-1	128000000000
CSR		SETON	9341	S00EX-1	129000000000
CSR		MOVE '1' @MK,2		S00EX-1	130000000000
CSR		GOTO ENDEXE		S00EX-1	131000000000
C*	----	-----		S00EX-1	132000000000
CSR	END			S00EX-1	133000000000
CSR	END			S00EX-1	134000000000
CSR	END			S00EX-1	135000000000
C*				S00EX-1	136000000000
C*	If ROLL DOWN key pressed, process read prior.			S00EX-1	137000000000
C*	-----	-----		S00EX-1	138000000000
C*				S00EX-1	139000000000
CSR	@@AID	IFEQ #FROLD		S00EX-1	140000000000
C*				S00EX-1	141000000000
C*	Reset error indicators if roll			S00EX-1	142000000000
C*				S00EX-1	143000000000
CSR		MOVE\$RESET *IN,41		S00EX-1	144000000000
CSR		MOVE '0' *IN,40		S00EX-1	145000000000
CSR		SETOF	818299	S00EX-1	146000000000
MF	%	READP&01FORMAT	9981	S00EX-1	147000000000
R93950	B0010	- STD/M	- Action Code		DATE - 2/02/17
CSR	*IN81	IFEQ '1'		S00EX-1	148000000000
MF	\$RDKEY	SETLL&01FORMAT		S00EX-1	149000000000
CSR		SETOF	8299	S00EX-1	150000000000
MF	%	READP&01FORMAT	9982	S00EX-1	151000000000
C*				S00EX-1	152000000000
C*	If error on read, set error.			S00EX-1	153000000000
C*				S00EX-1	154000000000
CSR	*IN82	IFEQ '1'		S00EX-1	155000000000
CSR		SETON	9341	S00EX-1	156000000000
CSR		MOVE '1' @MK,2		S00EX-1	157000000000
CSR		GOTO ENDEXE		S00EX-1	158000000000
C*	----	-----		S00EX-1	159000000000
CSR	END			S00EX-1	160000000000
CSR	END			S00EX-1	161000000000
CSR	END			S00EX-1	162000000000
C*				S00EX-1	163000000000
C*	Load video screen data on roll keys.			S00EX-1	164000000000
C*	-----	-----		S00EX-1	165000000000
C*				S00EX-1	166000000000
CSR	@@AID	IFEQ #FROLU		S00EX-1	167000000000
CSR	@@AID	OREQ #FROLD		S00EX-1	168000000000
/*				S00EX-1	169000000000
/*	Include record lock logic if update files exist.			S00EX-1	169100000000
/*				S00EX-1	169200000000
+FILEC*	*ANY	DB ZUNLOCK @		S00EX-1	169300000000
C*				S00EX-1	169400000000
MCU01C*				S00EX-1	169900000000
MCU01C*	Cost Center security edit.			S00EX-1	170000000000
MCU01C*				S00EX-1	171000000000
MCU01CSR		MOVE&01(FILE)#FILE		S00EX-1	172000000000

MCU01CSR		MOVE&01KEY	#MCU	S00EX-1	173000000000
MCU01CSR	#AUT	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	#MAUT	ANDNE'1'		S00EX-1	181000000000
MCU01CSR		MOVE '1'	\$SECUR	S00EX-1	182000000000
MCU01CSR		END		S00EX-1	183000000000
CSR	\$SECUR	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	@@AID	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	001000000000
/*				S00OP	000200000000
/*			If the display file has the selection option field,	S00OP	000300000000
/*			include the S00OP subroutine to process selection options.	S00OP	000400000000
/*				S00OP	001000000000
+FLDNC**	VDSELC	*AND		S00OP	001100000000
-FLDNC**	SFSELC	S00OP-1		S00OP	001200000000
/*				S00OP	001300000000
+FLDNC**	SFSELC	S00OP-2		S00OP	001400000000
C*				S00VL-1	001500000000
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 @@RRN.	S00VL-1	007000000000
C*				S00VL-1	008000000000
CSR	S00VL	BEGSR		S00VL-1	009000000000
C*		----		S00VL-1	010000000000
C*				S00VL-1	011000000000
CSR	##RVAL	IFEQ '*BLANK'		S00VL-1	012000000000
CSR		MOVE *BLANK	##RVAL	S00VL-1	013000000000
CSR		END		S00VL-1	014000000000
S00VLC*				S00VL-1	015000000000
C*				S00VL-1	016000000000
CSR	ENDOVL	ENDSR		S00VL-1	017000000000
COPY C*****				S00VL-1	018000000000
C*				S001-1	001000000000
C*			SUBROUTINE S001 - Clear Fields	S001-1	002000000000
C*			-----	S001-1	003000000000
C*				S001-1	004000000000
C*			Processing: 1. Reset all video screen and data file fields	S001-1	005000000000
C*			for next transaction.	S001-1	006000000000
C*			2. Clear action code only if requested.	S001-1	007000000000
C*				S001-1	008000000000
CSR	S001	BEGSR		S001-1	009000000000
C*		----		S001-1	010000000000
C*				S001-1	011000000000
C*			Reset fields for next transaction.	S001-1	012000000000
C*				S001-1	013000000000
MF CSR	*NOKEY	CLEAR&01FORMAT		S001-1	013100000000
CLRY C*				S001-1	014000000000
CSR		MOVESVL24M	VDL24	S001-1	015000000000
CSR		MOVE ' '	@IN37 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	@@AID	IFEQ #FCLR		S001-1	020000000000
CSR		MOVE *ALL'0'	\$RESET	S001-1	021000000000
CSR		MOVEA\$RESET	*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

C*	5. Load video screen output on inquiry.	S003-1	009000000000
C*		S003-1	010000000000
CSR	S003 BEGSR	S003-1	011000000000
C*	----	S003-1	012000000000
C*		S003-1	012100000000
C*	Load data field dictionary parameters (one cycle only).	S003-1	012200000000
C*		S003-1	012300000000
CSR	\$998 CASEQ' ' S998	S003-1	012400000000
C*	-----	S003-1	012500000000
CSR	END	S003-1	012600000000
C*		S003-1	013000000000
C*	Reset error indicators and arrays.	S003-1	014000000000
C*		S003-1	015000000000
CSR	MOVE *ALL'0' \$RESET 39	S003-1	016000000000
CSR	MOVE *BLANK \$REST1 63	S003-1	016100000000
CSR	MOVEA\$RESET *IN,41	S003-1	017000000000
CSR	MOVEA\$REST1 @MK,2	S003-1	018000000000
CSR	CLEAR@ER	S003-1	019000000000
C*	-----	S003-1	020000000000
KEYS C*		S003-1	021000000000
C*	-----	S003-1	022000000000
MF CSR	CHAIN&01FORMAT 9899	S003-1	023000000000
MCU01C*		S003-1	024000000000
MCU01C*	Cost Center security edit.	S003-1	025000000000
MCU01C*		S003-1	026000000000
MCU01CSR	MOVEL&01(FILE)#FILE	S003-1	027000000000
MCU01CSR	MOVEL&01KEY #MCU	S003-1	028000000000
MCU01CSR	#AUT IFNE '1'	S003-1	029000000000
MCU01CSR	#FAUT ANDNE'1'	S003-1	030000000000
MCU01CSR	EXSR C0000	S003-1	031000000000
MCU01C*	-----	S003-1	032000000000
MCU01CSR	END	S003-1	033000000000
MCU01CSR	#AUT IFNE '1'	S003-1	034000000000
MCU01CSR	#FAUT ANDNE'1'	S003-1	035000000000
MCU01CSR	#MAUT ANDNE'1'	S003-1	036000000000
MCU01CSR	MOVE '1' \$\$SECR 1	S003-1	037000000000
MCU01CSR	END	S003-1	038000000000
C*		S003-1	039000000000
C*	If security violation, set error condition.	S003-1	040000000000
C*		S003-1	041000000000
CSR	\$\$SECR IFEQ '1'	S003-1	042000000000
CSR	MOVE '1' @MK,8	S003-1	043000000000
CSR	SETON 9341	S003-1	044000000000
CSR	MOVE ' ' \$\$SECR 1	S003-1	045000000000
CSR	GOTO END003	S003-1	046000000000
C*	-----	S003-1	047000000000
CSR	END	S003-1	048000000000
C*		S003-1	049000000000
C*	Edit result of read and action code.	S003-1	050000000000
C*		S003-1	051000000000
CSR	*IN98 IFEQ '1'	S003-1	052000000000
CSR	*IN21 COMP '0' 41 *error*	S003-1	053000000000
CSR	ELSE	S003-1	054000000000
CSR	*IN21 COMP '1' 41 *error*	S003-1	055000000000
R93950	B0010 - STD/M - Action Code	S003-1	DATE - 2/02/17
CSR	END	S003-1	056000000000
C*		S003-1	057000000000
C*	If indicator 41 on, invalid key for action code.	S003-1	058000000000
C*		S003-1	059000000000
CSR	*IN41 IFEQ '1'	S003-1	060000000000
CSR	MOVE '1' @MK,2	S003-1	061000000000
CSR	SETON 93	S003-1	062000000000
CSR	END	S003-1	063000000000
C*		S003-1	064000000000
C*	If indicator 99 on, record in use.	S003-1	065000000000
C*		S003-1	066000000000
CSR	*IN99 IFEQ '1'	S003-1	067000000000
CSR	CALL 'P98RLCK' 81	S003-1	067100000000
C*	-----	S003-1	067200000000
CSR	PARM ##PSDS	S003-1	067300000000
CSR	MOVE '1' @MK,6	S003-1	068000000000
CSR	SETON 9341	S003-1	069000000000
CSR	END	S003-1	070000000000
C*	-----	S003-1	071000000000
C*		S003-1	072000000000
C*	If not inquiry, skip remainder of subroutine.	S003-1	073000000000
C*		S003-1	074000000000
CSR	*IN24 CABEQ'0' END003	S003-1	075000000000
C*	-----	S003-1	076000000000
C*	-----	S003-1	077000000000
C*		S003-1	078000000000
C*	Release record lock on master file.	S003-1	079000000000
C*		S003-1	079100000000
CSR	*IN98 IFEQ '0'	S003-1	079200000000
CSR	*IN99 ANDEQ'0'	S003-1	080000000000
CSR	EXCPTUNLOCK	S003-1	081000000000
CSR	END	S003-1	081100000000
C*		S003-1	082000000000
C*	If errors, skip remainder of subroutine.	S003-1	083000000000
C*		S003-1	084000000000
CSR	*IN93 CABEQ'1' END003	S003-1	085000000000
C*	-----	S003-1	086000000000
C*	-----	S003-1	087000000000
C*		S003-1	088000000000
C*	Move data base information to video screen.	S003-1	089000000000

C*			S003-1	090000000000
CSR	EXSR S004		S003-1	091000000000
C*	----		S003-1	092000000000
C*	-----		S003-1	093000000000
CSR	END003 ENDSR		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	B0010 - STD/M - Action Code			DATE - 2/02/17
C*	julian to the system format using program		S004-1	014000000000
C*	X0028.		S004-1	015000000000
C*			S004-1	016000000000
CSR	S004 BEGSR		S004-1	017000000000
C*	----		S004-1	018000000000
DSP1	C*		S004-1	025000000000
CSR	END004 ENDSR		S004-1	026000000000
COPY	C*****		S004-1	027000000000
C*			S005-1	001000000000
C*	SUBROUTINE S005 - Scrub Input		S005-1	002000000000
C*	-----		S005-1	003000000000
C*			S005-1	004000000000
C*	Processing: 1. Validate all video input.		S005-1	005000000000
C*	All numeric fields must be processed		S005-1	006000000000
C*	thru subroutines C0012 and C0015 in order		S005-1	007000000000
C*	to scrub the alpha input field and convert		S005-1	008000000000
C*	back to internal numeric representation of		S005-1	009000000000
C*	15 digits and 0 decimals.		S005-1	010000000000
C*			S005-1	011000000000
C*	Date fields must be converted from system		S005-1	012000000000
C*	format to their internal format of month,		S005-1	013000000000
C*	day and year or julian using program X0028.		S005-1	014000000000
C*			S005-1	015000000000
C*	2. Update data record fields from video.		S005-1	016000000000
C*			S005-1	017000000000
CSR	S005 BEGSR		S005-1	018000000000
C*	----		S005-1	019000000000
C*			S005-1	020000000000
C*	If not addition or change, bypass subroutine		S005-1	021000000000
C*			S005-1	022000000000
CSR	*IN21 IFEQ '0'		S005-1	023000000000
CSR	*IN22 ANDEQ '0'		S005-1	024000000000
CSR	GOTO END005		S005-1	025000000000
C*	-----		S005-1	026000000000
CSR	END		S005-1	028000000000
C*			S005-1	029000000000
FIELD	C*		S005-1	030000000000
CSR	END005 ENDSR		S005-1	031000000000
COPY	C*****		S010-1	001000000000
C*			S010-1	002000000000
C*	SUBROUTINE S010 - Update Data Base		S010-1	003000000000
C*	-----		S010-1	004000000000
C*			S010-1	005000000000
C*	Processing: 1. Update data base file based upon valid		S010-1	006000000000
C*	action codes.		S010-1	007000000000
C*			S010-1	008000000000
CSR	S010 BEGSR		S010-1	009000000000
C*	----		S010-1	010000000000
AC*			S010-1	011000000000
AC*	If add action, add record.		S010-1	012000000000
AC*			S010-1	013000000000
ACSR	*IN21 IFEQ '1'		S010-1	014000000000
MF	ACSR % WRITE&01FORMAT 99		S010-1	015000000000
ACSR	END		S010-1	016000000000
CC*			S010-1	017000000000
CC*	If change action, update record.		S010-1	018000000000
CC*			S010-1	019000000000
CCSR	*IN22 IFEQ '1'		S010-1	020000000000
MF	CCSR % UPDAT&01FORMAT 99		S010-1	021000000000
CCSR	END		S010-1	022000000000
R93950	B0010 - STD/M - Action Code			DATE - 2/02/17
DC*			S010-1	023000000000
DC*	If delete action, delete record.		S010-1	024000000000
DC*			S010-1	025000000000
DCSR	*IN23 IFEQ '1'		S010-1	026000000000
MF	DCSR % DELET&01FORMAT 99		S010-1	027000000000
DCSR	END		S010-1	028000000000
C*			S010-1	029000000000
C*	Clear data field for next transaction		S010-1	030000000000
C*			S010-1	031000000000
CSR	MOVE #FCLR @@AID		S010-1	032000000000
CSR	EXSR S001		S010-1	033000000000
C*	----		S010-1	034000000000
CSR	END010 ENDSR		S010-1	035000000000
COPY	C*****		S999-1	001000000000
C*				

C*****	S999-1	09700000000
/*	S999-1	09800000000
/* If processing options exist, include copy module	S999-1	09900000000
/*	S999-1	10000000000
+FLDNC* *OPTION ZOPTIONC	S999-1	10100000000
COPY C*****	S999-1	10200000000
MF O&01FMT E UNLOCK	S999-1	10300000000

Appendix G - Functional Servers

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:

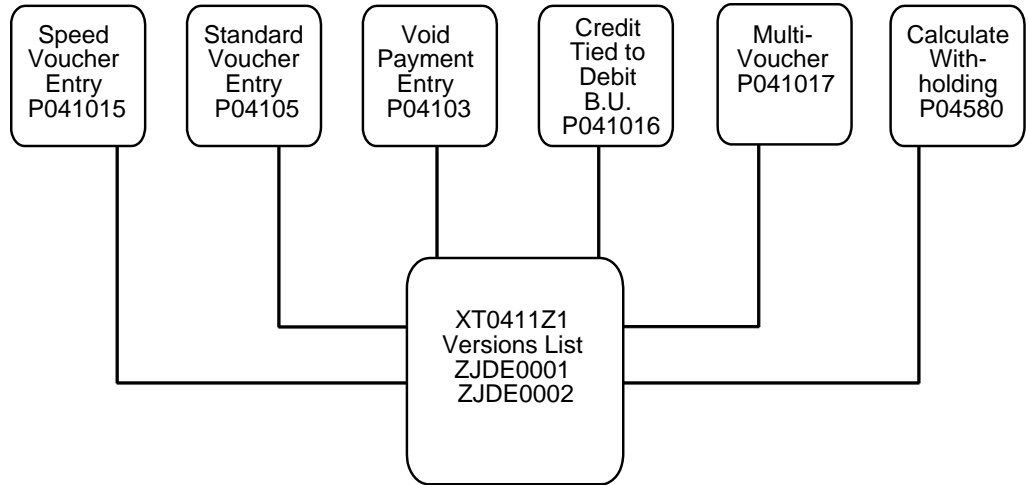
1. Create a DREAM Writer version for a specific functional server program (for example, XT0411Z1 for voucher entry).
2. Set the processing options within the version according to your company requirements.
3. Specify the version you want the entry program to use in the processing options for that entry 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 *Technical Tools Guide*.

Example: Voucher Processing Functional Server

The following graphic includes the programs that use the voucher processing functional server. JD Edwards World provides two demo versions of the functional server, ZJDE0001 and ZJDE0002.



Miscellaneous

*ENTRY PLIST entries, 3-52

A

A0010 – Interactive Subfile Inquiry program, 9-9

A0020 – Interactive Single Record Inquiry program, 9-10

Abbreviations for the program types index, 8-7

About

- abbreviations for the program types index, 8-7

- action diagramming, 7-19

- additional tools, 7-1

- changing generated source, 5-3

- clone status all/only active toggle, 8-10

- creating or modifying program types, 8-7

- edit screen, 3-30

- foundation information, 2-1

- glossary K, 8-10

- logic modules, 8-13

- master source code, 1-6

- option and function exits, 3-33

- options for program types, 8-8

- program design language, 4-1

- program generator, 1-6

- program specifications, 1-6

- program types, 1-6

- program types cross reference, 8-9

- program types index, 8-8

- quick start application tool, 7-7

- quick start CL generator, 7-3

- source modifications, 5-1

- special characters, 3-27

- the detailed programming facility screen, 3-39

- user defined PDL, 8-51

- using the source code inventory and database, 8-1

Accessing

- CASE profiles, 2-13

- data item formula revisions, 4-21

- logic translation feature, 7-24

- program generator, 3-3

- program generator options, 3-5

- program generator specifications, 3-3

- quick start CL generator, 7-3

- the logic module index, 8-15

Action diagram

- building, 7-19

- overview, 7-19

- viewing, 7-20

Adding

- new Q & A dialogue, 8-39

- PDL to a field, 4-21

Additional tools, 7-1

All help instructions, 8-14

Answer Entry screen (P98552), 8-39, 8-40

Application development cycle, 1-1

Arrays for EMK, @MK and @ER, 9-5

Assignments in PDL, 4-7

Authorities for objects, 2-17

B

B0010 – Interactive Single Record Maintenance program, 9-11

Blocks of statements, 4-5

Browse for screens or reports using quick start, 7-12

Browse screen, 3-52, 3-59

Build Action Diagrams screen (P98300), 7-19

Building an action diagram, 7-19

C

C0010 – Batch Report with Totals program, 9-12

C0020 – Batch Report with Totals and Subheadings program, 9-13

C0025 – Batch Report with Totals and Subheadings program, 9-14

CAD, 1-1

CAP

- overview, 1-1

- status

 - changing, 5-8

 - to correct invalid error, 5-10

CASE

- benefits, 1-4

- menus, 1-8

- profiles

 - accessing, 2-13

 - understanding, 2-12

- program types, 9-9

- specifications inquiry overview, 8-23

CASE Profiles screen (P98009), 2-14

CASE Specifications Inquiry screen (P93130), 8-24

- Changing
 - a dialogue, 8-44
 - CAP status, 5-8
 - program specifications, 7-13
- Checklist
 - data file design aid, 9-1
 - program generator, 9-2
 - report design aid, 9-2
 - screen design aid, 9-1
- CL
 - overview, 5-11
 - program compile using quick start, 7-5
- Comments in PDL, 4-6
- Compiling
 - CL programs using quick start, 7-5
 - using quick start, 7-12, 7-14
- Computer Assisted Design, 1-1
- Computer Assisted Programming, 1-1
- Concepts and terms, 1-4
- Conditional directives, 8-35
- Conditions in PDL, 4-11
- Control language, 5-11
- Copy File screen (CPYF), 2-10
- Copying
 - a dialogue, 8-44
 - model control language, 5-11
 - program specifications, 8-20
- Create/Modify Logic Modules screen (P93001SEU), 8-18
- Create/Modify Program Types screen (P93001), 8-11, 8-51
- Creating
 - *ENTRY PLIST entries, 3-52
 - a partial KLIST for a file, 3-56
 - formula library entry, 8-19
 - JDESRC file for use with Program Generator, 2-10
 - logic modules, 8-18
 - report programs, 6-7
 - subheadings, 6-12
 - total formats, 6-9
 - user defined PDL, 8-51
- Customizing model control language, 5-12
- D**
- D0010 – Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number program, 9-15
- D0020 – Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number program, 9-16
- D0030 – Interactive Subfile Maintenance without Action Code, without Options, by Relative Record Number with Read Next Modified Record program, 9-18
- D0040 – Interactive Subfile Maintenance with Action Code, with Options, by Key program, 9-19
- D0050 – Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Relative Record Number program, 9-20
- D0060 - Interactive Subfile Maintenance with Action Code, without Options, by Key program, 9-22
- D0070 – Interactive Subfile Maintenance with Action Code, with Options, by Relative Record Number program, 9-23
- D0080 – Interactive Subfile Maintenance without Action Code, with Options, by Relative Record Number program, 9-24
- D0090 – Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, Balance program, 9-26
- D0100 – Interactive Subfile Maintenance with Two Master Files, with Action Code, with Options, by Key program, 9-27
- Data dictionary
 - editing disabled, 3-55
 - glossary update, 7-15
- Data fields to select using quick start, 7-11
- Data file design aid checklist, 9-1
- Data Item Formula Revisions screen (P93109), 4-14, 4-19, 4-21, 8-55
- Data structure
 - I00DSPROG source, 9-39
 - I00SC source, 9-40
- Database operations in PDL, 4-8
- Database update function for subfiles, 3-51
- Date and time standards, 9-8
- Define Generator Specification screen (P93100M), 3-9, 3-20, 3-29, 3-33, 3-39, 3-60, 5-8, 5-9, 8-53
- Define Program Generator Specifications screen (P93100M), 3-5
- Defining
 - applications using quick start, 7-9
 - general instructions, 3-27
 - option and function exits, 3-34
 - processing options, 3-57
 - program purpose and type, 3-9
- Deleting a dialogue, 8-48
- Detailed information, 1-5
 - CAD, 1-5
 - CAP, 1-5
 - CASE profiles, 1-5
- Detailed Programming Facility, 3-39, 6-4, 6-6
- Detailed Programming Facility screen (P93105), 3-40, 3-56, 4-21, 8-54
- Development libraries, 2-9
- Dialogue
 - changing, 8-44
 - copying, 8-44

- deleting, 8-48
- existing, 8-42
- rename, 8-45
- reviewing the flow, 8-43
- running, 8-46
- Dialogue Copy screen (P98536), 8-44, 8-45
- Dialogue Descriptions screen (P98541), 8-39
- Dialogue Flow Revisions screen (P98531), 8-43
- Dialogue Lists screen (P98530), 8-43, 8-44, 8-45, 8-46
- Dialogue Selection screen (P98533), 3-13, 8-47
- Dialogue Test screen (P98535), 8-46, 8-48
- Dialogue Test screen (P98537), 8-49
- Directives
 - conditional, 8-35
 - exception, 8-34
 - functional, 8-25
 - substitution, 8-32
 - understanding, 8-25
- Disable data dictionary editing, 3-55
- Display Action Diagram screen (P92705), 7-21, 7-23
- DREAM Writer considerations for report formats, 6-14

E

- E0010 – Interactive Window program, 9-28
- Edit screen, 3-28, 3-30, 8-22, 8-52
- Editing, parsing, and source generation of PDL, 4-17
- Enabling database update function for subfiles, 3-51
- Error handling using arrays, 9-5
- Exception directives, 8-34

F

- Features, 1-3
- Field Definition screen, 6-10
- Field protection, 3-54
- File specifications, 3-17
- File Specifications screen (P93102), 3-20, 3-24
- Flows, 9-79
- Formula library entry for creating or modifying, 8-19
- Formula Library Entry screen (P93109), 4-13, 8-19
- Full data field parameters
 - accessing, 3-42
 - understanding, 3-42
- Full Data Field Parameters screen (P93125), 3-43, 3-48, 3-49, 3-51, 3-53, 3-54, 3-55
- Function exits
 - previous profile, 2-17
- Function exits set up, 3-33
- Functional directives, 8-25
- Functional servers
 - example, 9-100
 - overview, 9-99

G

- General instructions for help text, 3-27
- Generated source code
 - changing, 5-4
 - from file specifications, 3-24
 - regenerating, 5-7
 - resolving errors, 5-9
- Generator Updates screen (P9366), 8-23
- Global program regeneration, 8-14
- Glossary K, 8-10
- Glossary revisions screen (P92001), 7-16
- Glossary updates, 7-15
- Guidelines for program types, 9-9

H

- Help file updates, 3-30
- Help instructions
 - rebuild for a single program, 8-14
 - rebuild for all programs, 8-14
- Help Instructions Master file (F98HELP), 3-28

I

- I00DSPROG data structure source, 9-39
- I00SC data structure source, 9-40
- Interactive non-subfile program flow, 9-82

J

- Job queues, 2-11

K

- Keywords
 - in blocks of statements, 4-5
 - in comments, 4-6
- KLIST standards, 9-7

L

- Line structure, 8-3
- Loading VCO description fields, 3-47
- Locating
 - a dialogue flow, 8-43
 - screens or reports using quick start, 7-12
- Logic Module Cross Reference screen (P93952), 8-15
- Logic modules, 8-13
 - accessing the index, 8-14
 - creating or modifying, 8-18
 - detail, 8-13
 - maintaining, 8-17
 - viewing cross reference, 8-15
 - viewing op codes, 8-16
- Logic translation feature, 7-24

Loops in PDL, 4-10

M

Maintaining logic modules

- remove logic module, 8-17
- resequence logic module, 8-17

Master dialogue questions, 8-37

Menus, 1-8

Miscellaneous keywords and syntax in PDL, 4-13

Model Control Language Programs

- customizing, 5-12
- overview, 5-11
- provided by JD Edwards World, 5-12

Modifying

- formula library entry, 8-19
- logic modules, 8-18
- program specifications, 7-13

Moving program specifications, 8-20

Multi-member source file, 2-9

N

Naming convention standards, 9-7

New Q & A dialogue, 8-39

O

Object authorities, 2-17

- job control, 2-17
- job queues, 2-17
- source file, 2-17
- source library, 2-17

Op Codes screen (P93108), 8-16

Operators in assignments, 4-7

Option & Function exits screen (P93104), 3-34

Options

- defining, 3-33
- overview, 3-33

P

Parameter Copy/Move screen (P93890), 8-20

Partial KLIST created for a file, 3-56

PDL

editing, parsing, and source generation

- editing, 4-17
- parsing, 4-17
- source code generation, 4-18

statements, 4-3

- constants, 4-5
- database files, 4-4
- keywords, 4-4
- operations, 4-4
- operators, 4-5
- punctuation, 4-5
- variables, 4-4

user defined

creating, 8-51

Prerequisites, 2-3

- common user defined codes, 2-4
- Program Generator files, 2-3
- source code for copy modules, 2-7
- source code for JD Edwards World files, 2-7

Primary module, 8-10

Print Program Specification screen (P98300), 8-21

Printing program generator specifications, 8-21

Process for quick start, 7-7

Processing options defined, 3-57

Processing Options Setup screen (P98304), 3-60

Program calls in PDL, 4-9

Program code sample, 9-87

Program Design Language, *See* PDL

Program flows, 9-82

Program Generator

- accessing, 3-3
- checklist, 9-2
- files, 2-3
 - program design language, 2-4
- Q & A dialogue, 2-4
- source modifications/helps, 2-3
- specifications, 2-3
- merging updates, 8-23
- printing specifications, 8-21
- reviewing options, 3-5
- specifications
 - accessing, 3-3

Program purpose and type definition, 3-9

Program Purpose and Type screen (P93100), 3-10, 5-8

Program specifications

- modifications using quick start, 7-13

Program specifications to copy or move, 8-20

Program types

- B0010 example, 9-87
- conversion, 1-7
- creating or modifying, 8-7
- cross reference, 8-9
- index, 8-8
- interactive, 1-7
- report, 1-7
- server, 1-7
- window, 1-7

Program Types

- A0010 – Interactive Subfile Inquiry, 9-9
- A0020 – Interactive Single Record Inquiry, 9-10
- B0010 – Interactive Single Record Maintenance, 9-11
- C0010 – Batch Report with Totals, 9-12
- C0020 – Batch Report with Totals and Subheadings, 9-13
- C0025 – Batch Report with Totals and Subheadings, 9-14
- D0010 – Interactive Subfile Maintenance with Action Code, without Options, by Relative Record Number, 9-15

- D0020 – Interactive Subfile Maintenance
without Action Code, without Options, by
Relative Record Number, 9-16
- D0030 – Interactive Subfile Maintenance
without Action Code, without Options, by
Relative Record Number with Read Next
Modified Record, 9-18
- D0040 – Interactive Subfile Maintenance with
Action Code, with Options, by Key, 9-19
- D0050 – Interactive Subfile Maintenance with
Two Master Files, with Action Code, with
Options, by Relative Record Number, 9-20
- D0060 - Interactive Subfile Maintenance with
Action Code, without Options, by Key, 9-22
- D0070 – Interactive Subfile Maintenance with
Action Code, with Options, by Relative
Record Number, 9-23
- D0080 – Interactive Subfile Maintenance
without Action Code, with Options, by
Relative Record Number, 9-24
- D0090 – Interactive Subfile Maintenance with
Action Code, without Options, by Relative
Record Number, Balance, 9-26
- D0100 – Interactive Subfile Maintenance with
Two Master Files, with Action Code, with
Options, by Key, 9-27
- E0010 – Interactive Window, 9-28
guidelines, 9-9
overview, 9-9
- X0010 – Batch Update with Report, 9-30
- X0020 – Batch Update, 9-31
- X0030 – Batch Update with Subroutine S001, 9-
32
- X0040 - Batch Update with Report, 9-33
- Y0010 – Conversion, Two Files with Error
Report, 9-34
- Y0020 – Conversion, One File Update with
Error Report, 9-35
- Y0030 – Conversion, One File Write with Error
Report, 9-36
- Program Types Index screen (P93900), 8-8
- Program Types X-Reference screen (P93953), 8-9
- Programming standards, 9-5
- Programs and IDs
- CPYF (copy file), 2-10
 - edit, 8-22, 8-52
 - P00051 (user defined code revisions), 2-4
 - P2710 (translation table), 7-24
 - P92001 (glossary revisions), 7-16
 - P92705 (display action diagrams), 7-21, 7-23
 - P93001 (create/modify program types), 8-11, 8-
51
 - P93001SEU (create/modify logic modules), 8-18
 - P93100 (program purpose and type), 3-10, 5-8
 - P93100M (define generator specification), 3-9, 3-
20, 3-29, 3-33, 3-39, 3-60, 5-8, 5-9, 8-53
 - P93100M (define generator specifications), 3-5
 - P93102 (file specifications), 3-20, 3-24
 - P93104 (option & function exits), 3-33
 - P93105 (detailed programming facility), 3-40, 3-
56, 4-21, 8-54
 - P93108 (logic module op codes), 8-16
 - P93109 (data item formula revisions), 4-14, 4-19,
4-21, 8-55
 - P93109 (formula library entry), 4-13, 8-19
 - P93125 (full data field parameters), 3-43, 3-48, 3-
49, 3-51, 3-53, 3-54, 3-55
 - P93130 (CASE specifications inquiry), 8-24
 - P93513 (quick start application tool), 7-9
 - P93513J (quick start CL generator), 7-3
 - P93515V (quick start application tool), 7-11, 7-12,
7-13, 7-14, 7-15, 7-16
 - P93515V (quick start CL generator), 7-5
 - P9366 (generator updates), 8-23
 - P93890 (parameter copy/move), 8-20
 - P93900 (program types index), 8-8
 - P93952 (logic module cross reference), 8-15
 - P93953 (program types x-reference), 8-9
 - P98009 (CASE profiles), 2-14
 - P9801 (software versions repository), 2-14, 3-24
 - P98300 (build action diagrams), 7-19
 - P98300 (print program specification), 8-21
 - P98304 (processing options setup), 3-57
 - P98529 (simple question & answer), 8-38
 - P98530 (dialogue lists), 8-43, 8-44, 8-45, 8-46
 - P98531 (dialogue flow revisions), 8-43
 - P98533 (dialogue selection), 3-13, 8-47
 - P98534 (quiz answer review), 8-47
 - P98535 (dialogue test), 8-46, 8-48
 - P98536 (dialogue copy), 8-44, 8-45
 - P98537 (dialogue test), 8-49
 - P98541 (dialogue descriptions), 8-39
 - P98551 (question entry), 8-39
 - P98552 (answer entry), 8-39, 8-40
 - RM/M (remove member), 2-11
- Project management, 2-11
- Protecting fields from being cleared, 3-54
- ## Q
- Q & A dialogue, 8-39
- Question and answer system overview, 8-37
- Question Entry screen (P98551), 8-39
- Questions in a master dialogue, 8-38
- Quick reference of program types, 9-9
- Quick Start
- application tool overview, 7-7
 - CL generator for creating programs, 6-3, 6-5, 7-3
 - CL generator overview, 7-3
 - generating subfile inquiry programs, 6-4
 - generator to create subfile maintenance
programs, 6-6
 - steps for process, 7-7
- Quick Start Application Tool screen (P93513), 7-9

Quick Start Application Tool screen (P93515V), 7-11, 7-12, 7-13, 7-14, 7-15, 7-16
 Quick Start C L Generator screen (P93515V), 7-5
 Quick Start CL Generator screen (P93513J), 7-3
 Quiz Answer Review screen (P98534), 8-47
 Quiz to determine program type, 8-48

R

RDA, 6-7
 Rebuild help instructions, 8-14
 Record Formats List screen, 6-9, 6-13
 Regenerating source code, 5-7, 8-14
 Remove logic module, 8-17
 Remove Member (RM/M), 2-11
 Renaming a dialogue, 8-45
 Report design aid
 checklist, 9-2
 Report Design Aid
 creating reports, 6-7
 Report format considerations for DREAM Writer, 6-14
 Report program with subheadings flow, 9-85
 Report program without subheadings flow, 9-84
 Report programs subheadings and totals, 6-7
 Reports or screens
 browse or update using quick start, 7-11
 compiling using quick start, 7-12
 Resequence logic module, 8-17
 Reviewing
 a dialogue flow, 8-43
 questions, 8-38
 source modifications, 8-22
 RPG subroutines, 9-79
 Running a dialogue, 8-46
 Running a quiz, 8-48

S

Sample program code, 9-87
 SAR, 2-12
 Screen design aid checklist, 9-1
 Screens
 Answer Entry, 8-39, 8-40
 Browse, 3-52, 3-59
 Build Action Diagrams, 7-19
 CASE Profiles, 2-14
 CASE Specifications Inquiry, 8-24
 Copy File (CPYF), 2-10
 Create/Modify Logic Modules, 8-18
 Create/Modify Program Types, 8-11, 8-51
 Data Item Formula Revisions, 4-14, 4-19, 4-21, 8-55
 Define Generator Specification, 3-9, 3-20, 3-29, 3-33, 3-39, 3-60, 5-8, 5-9, 8-53
 Define Generator Specifications, 3-5

Detailed Programming Facility, 3-40, 3-56, 4-21, 8-54
 Dialogue Copy, 8-44, 8-45
 Dialogue Descriptions, 8-39
 Dialogue Flow Revisions, 8-43
 Dialogue Lists, 8-43, 8-44, 8-45, 8-46
 Dialogue Selection, 3-13, 8-47
 Dialogue Test, 8-46, 8-48, 8-49
 Display Action Diagram, 7-21, 7-23
 Edit, 3-28, 3-30, 8-22, 8-52
 Field Definition, 6-10
 File Specifications, 3-20, 3-24
 Formula Library Entry, 4-13
 Formula Library Entry, 8-19
 Full Data Field Parameters, 3-43, 3-48, 3-49, 3-51, 3-53, 3-54, 3-55
 Generator Updates, 8-23
 Glossary, 7-16
 Logic Module Cross Reference, 8-15
 Op Codes, 8-16
 Option & Function exits, 3-34
 Parameter Copy/Move, 8-20
 Print Program Specification, 8-21
 Processing Options Setup, 3-60
 Program Purpose and Type, 3-10, 5-8
 Program Types Cross Reference, 8-9
 Program Types Index, 8-8
 Question Entry, 8-39
 Quick Start Application Tool, 7-9, 7-11, 7-12, 7-13, 7-14, 7-15, 7-16
 Quick Start CL Generator, 7-3, 7-5
 Quiz Answer Review, 8-47
 Record Formats List, 6-9, 6-13
 Remove Member, 2-11
 Simple Question & Answer, 8-38
 Software Versions Repository, 2-14, 3-24
 Source Entry Utility, 5-11
 Translation Table, 7-24
 User Defined Code Revisions, 2-4
 Screens or reports
 browse or update using quick start, 7-11
 compiling using quick start, 7-12
 Selecting data fields using quick start, 7-11
 Serial numbers, 8-3, 8-4
 Simple Question & Answer screen (P98529), 8-38
 Software Action Request, 2-12
 Software Versions Repository screen (P9801), 2-14, 3-24, 5-11
 Solving generation problems, 5-9
 Source code
 for copy modules, 2-7
 for JD Edwards World World files, 2-7
 inventory and database, 8-1
 regenerating, 5-7
 when to regenerate, 5-7
 Source Entry Utility screen, 5-11
 Source Listings, 9-39

- Source modifications
 - about, 5-1
 - code review, 8-22
 - Source sequence line numbers, 8-3
 - Source sequence line structure, 8-3
 - Source serial numbers, 8-3
 - Special characters
 - about, 3-27
 - within the help file, 3-28
 - Standards for programming, 9-5
 - Standards using functional servers, 9-99
 - Statements in PDL, 4-3
 - Structure of serial numbers, 8-4
 - Subfile Inquiry Program
 - components, 6-3
 - overview, 6-3
 - special considerations, 6-4
 - Subfile Maintenance Program
 - components, 6-5
 - overview, 6-5
 - special considerations, 6-6
 - Subfile program with options flow, 9-83
 - Subfiles database update function, 3-51
 - Subheadings
 - formats, 6-12
 - overview, 6-7
 - Submit program to compile using quick start, 7-14
 - Subroutines for RPG, 9-79
 - Substitution directives, 8-32
 - Syntax
 - in assignments, 4-7
 - in blocks of statements, 4-5
 - in comments, 4-6
 - 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
 - formats, 6-9
 - overview, 6-7
 - Translation Table screen (P92710), 7-24
- U**
- Understanding
 - assignments, 4-7
 - operator and syntax, 4-7
 - rules, 4-7
 - blocks of statements, 4-5
 - keywords and syntax, 4-5
 - rules, 4-5
 - CASE profiles, 2-12
 - comments
 - keywords and syntax, 4-6
 - rules, 4-6
 - conditions
 - keywords and syntax, 4-11
 - rules, 4-12
 - symbols, 4-11
 - database operations
 - keywords and syntax, 4-8
 - rules, 4-9
 - directives, 8-25
 - full data field parameters, 3-42
 - loops
 - keywords and syntax, 4-10
 - rules, 4-10
 - miscellaneous keywords and syntax
 - keywords and syntax, 4-13
 - rules for include, 4-13
 - rules for return, 4-15
 - program calls
 - keywords and syntax, 4-9
 - rules, 4-9
 - source sequence line numbers, 8-3
 - source sequence line structure, 8-3
 - source serial numbers, 8-3
 - structure of the serial number, 8-4
 - Updating
 - data dictionary, 7-15
 - glossary, 7-15
 - help file, 3-30
 - screens or reports using quick start, 7-12
 - Usage indicator standards, 9-6
 - User Defined Code Revisions screen (P00051), 2-4
 - User Defined Codes, 2-4
 - User-provided prerequisites, 2-9
 - Using
 - CASE specifications inquiry, 8-23
 - program generator updates, 8-23
- V**
- VCO description fields for screens, 3-47
 - Viewing
 - an action diagram, 7-20
 - logic module cross reference, 8-15
 - logic module op codes, 8-16
- W**
- Work field standards, 9-7
 - Working with
 - file specifications, 3-17
 - the question and answer system, 8-37
 - user provided prerequisites
 - development libraries, 2-9
 - job queues, 2-11
 - multi-member source file, 2-9
 - overview, 2-9
 - project management, 2-11

X

- X0010 – Batch Update with Report program, 9-30
- X0020 – Batch Update program, 9-31
- X0030 – Batch Update with Subroutine S001 program, 9-32
- X0040 - Batch Update with Report program, 9-33

Y

- Y0010 – Conversion, Two Files with Error Report program, 9-34
- Y0020 – Conversion, One File Update with Error Report program, 9-35
- Y0030 – Conversion, One File Write with Error Report program, 9-36