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 (eay@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.
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................................................................... 7-20

8 Source Code Inventory and Database

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

Understand Source Sequence..................................................................................... 8-3

Working with Program Types ...................................................................................... 8-7
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 Modules ............................................................................................ 8-13
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 Directives ................................................................................................. 8-25

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):

<table>
<thead>
<tr>
<th>Level</th>
<th>A/D Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 1</td>
<td>The Application Platform, which represents the Technical Foundation Guide.</td>
</tr>
<tr>
<td>Level 2</td>
<td>The Design Platform, which represents the Advanced Programming Concepts and Skills (APCS) Guide.</td>
</tr>
</tbody>
</table>
Overview to Computer Aided Software Engineering (CASE)

<table>
<thead>
<tr>
<th>Level</th>
<th>A/D Cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level 3</td>
<td>The Development Platform, which represents the Program Generator (CASE)</td>
</tr>
</tbody>
</table>

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
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
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.

<table>
<thead>
<tr>
<th>Alphanumeric Code</th>
<th>Type of Copy Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>Copy the member into the F specifications</td>
</tr>
<tr>
<td>Alphanumeric Code</td>
<td>Type of Copy Module</td>
</tr>
<tr>
<td>-------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>E</td>
<td>Copy the member into the E specifications</td>
</tr>
<tr>
<td>I</td>
<td>Copy the member into the I specifications</td>
</tr>
<tr>
<td>C</td>
<td>Copy the member into the C specifications</td>
</tr>
</tbody>
</table>

- 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.

<table>
<thead>
<tr>
<th>Unique Libraries</th>
<th>Common Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source = DEVSRC</td>
<td>Source = DEVLIB</td>
</tr>
<tr>
<td>Object = DEVOBJ</td>
<td>Object = DEVLIB</td>
</tr>
<tr>
<td>Data  = DEVDTA</td>
<td>Data  = DEVLIB</td>
</tr>
</tbody>
</table>

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.

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

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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source File</td>
<td>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.</td>
</tr>
<tr>
<td>Source Library</td>
<td>The default library where source will be stored. The source file specified must reside within this library.</td>
</tr>
<tr>
<td>Object Library</td>
<td>The default library where compiled objects will be stored.</td>
</tr>
<tr>
<td>CL Source File</td>
<td>The default library where source for CL programs will be stored. This file must reside within the specified source library.</td>
</tr>
<tr>
<td>Data File Library</td>
<td>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.</td>
</tr>
</tbody>
</table>
| SAR Number             | An abbreviation for software action request (SAR).  
<p>|                        | • *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. |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compile Target Release</td>
<td>Used by various System CRT commands (RPG, CLP, COBOL, C) to compile an object compatible with a specified target release.</td>
</tr>
<tr>
<td></td>
<td>- A value of *CURRENT compiles an object compatible to the release of the machine at compile time.</td>
</tr>
<tr>
<td></td>
<td>- A value of *PRV compiles an object compatible with both one release back and the current release.</td>
</tr>
<tr>
<td>Print Option</td>
<td>Used to designate whether or not a report will be generated when an object is compiled.</td>
</tr>
<tr>
<td></td>
<td>- 0 = no print</td>
</tr>
<tr>
<td></td>
<td>- 1 = print</td>
</tr>
<tr>
<td></td>
<td>- 2 = print and hold spool file</td>
</tr>
<tr>
<td></td>
<td>- 3 = print only, does not generate execution object (applies to COBOL and RPG only)</td>
</tr>
<tr>
<td></td>
<td>- 4 = print when compile or creation fails</td>
</tr>
<tr>
<td>Cross-Reference Listing</td>
<td>Enter Y or N. Specifies whether a cross-reference listing will be generated for variables and fields in a program's compile listing.</td>
</tr>
<tr>
<td>SAR File Library</td>
<td>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.</td>
</tr>
<tr>
<td>SAR Delivery Type</td>
<td>Associated with SAR logging, which tracks all modifications to JD Edwards World* software. For example, it will track when User Defined Codes are modified.</td>
</tr>
<tr>
<td></td>
<td>- *NONE = no logging.</td>
</tr>
<tr>
<td></td>
<td>- *LOG = log to SAR number 00000000 (no SAR number is used for logging).</td>
</tr>
<tr>
<td></td>
<td>- *DFT = log to a default SAR number (specified in the SAR Number field).</td>
</tr>
<tr>
<td></td>
<td>- *PROMPT = log and prompt the user for the SAR number to be used and allow the user to enter the revision notes.</td>
</tr>
<tr>
<td>Source Gen Opt (Future)</td>
<td>Specifies whether to generate source interactively or in batch for programs with this option (for example, Fast Path Application Tool).</td>
</tr>
<tr>
<td></td>
<td>- Allowed values are as follows:</td>
</tr>
<tr>
<td></td>
<td>1. generate source on-line (interactively)</td>
</tr>
<tr>
<td></td>
<td>2. generate source in batch</td>
</tr>
<tr>
<td>Helps Maint Opt (Future)</td>
<td>Enter a user defined code, 92/HL.</td>
</tr>
</tbody>
</table>
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.

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

<table>
<thead>
<tr>
<th>Specification</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Purpose and Type</td>
<td>Defines what kind of program you are designing and the status of the program generation (CAP status).</td>
</tr>
<tr>
<td>File Specifications</td>
<td>Allows the user to enter the data base files to be used by the program you are designing.</td>
</tr>
<tr>
<td>Define General Instructions</td>
<td>Allows the user to enter program-specific help instructions.</td>
</tr>
<tr>
<td>Define Option and Function Key Exits</td>
<td>Allows the user to define special program exits.</td>
</tr>
<tr>
<td>Detailed Programming Facility</td>
<td>Allows the user to specify data field definition parameters for fields included in the screen, the report, and the master files.</td>
</tr>
<tr>
<td>Define Processing Options</td>
<td>Allows the user to define processing options the program can use.</td>
</tr>
</tbody>
</table>
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 $S fields in the TOTAL formats of the report file for the 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 File 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 M 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 Pxxxxx member in the Additional Help/Modifications Master file (F93002)
- Creates a data item in the Data Item Master file (F9200)

The Program Purpose and Type screen includes:

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

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

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
Define Program Purpose and Type

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
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program ID</td>
<td>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.</td>
</tr>
<tr>
<td>Title</td>
<td>The title defaults to the description in the Software Versions Repository and should not be changed.</td>
</tr>
<tr>
<td></td>
<td>When help instructions are generated, this title appears as the Help program title.</td>
</tr>
<tr>
<td></td>
<td>Serves as the alpha description for the data item previously mentioned.</td>
</tr>
<tr>
<td>Purpose</td>
<td>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.</td>
</tr>
<tr>
<td>Product Code</td>
<td>Defaults to the system specified in the Software Versions Repository.</td>
</tr>
<tr>
<td>SAR Number</td>
<td>Defaults to the SAR entered in the Software Versions Repository.</td>
</tr>
</tbody>
</table>
## Define Program Purpose and Type

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAP Status</td>
<td>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.</td>
</tr>
<tr>
<td>Screen-specific information</td>
<td>This field also indicates whether the source code for a program can be modified using the program generator. The five additional serial number fields are still included in the source file (142 characters). When the source generation process is complete and the program has moved into a production source file (92 characters)</td>
</tr>
<tr>
<td>Program Type</td>
<td>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.</td>
</tr>
<tr>
<td>Lockout Act (action)</td>
<td>Allows the user to specify which action codes they do not want included in the program. 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. Utilizes array @NAC in the programs.</td>
</tr>
</tbody>
</table>

### 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.
<table>
<thead>
<tr>
<th>What is the general type of program?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive</td>
</tr>
<tr>
<td>Interactive form</td>
</tr>
<tr>
<td>Print a report</td>
</tr>
<tr>
<td>Conversion program</td>
</tr>
<tr>
<td>Batch update program</td>
</tr>
</tbody>
</table>
**Print a Report**

- Does the report print subheadings?
  - YES: C0020
  - NO: C0010

**Conversion Program**

- Does the conversion program update a single file and print an error report?
  - YES: Y0020
  - NO: Y0030

**Batch Update Program**

- Does the program add records to the master file?
  - YES: X0010
  - NO: X0020

- Does all the data need to be cleared after each processing cycle?
  - YES: X0030
  - NO: X0020
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.

<table>
<thead>
<tr>
<th>PROGRAM TYPE</th>
<th>DESCRIPTION</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>A0010</td>
<td>SFL (IBM Subfile) Inquiry</td>
<td>Specify the master file with an M or 1 in the Input field.</td>
</tr>
<tr>
<td>A0020</td>
<td>Single Record Inquiry</td>
<td></td>
</tr>
<tr>
<td>C0010</td>
<td>Standard Report</td>
<td></td>
</tr>
<tr>
<td>C0020</td>
<td>Standard Report –</td>
<td></td>
</tr>
<tr>
<td>C0025</td>
<td>Subheading</td>
<td></td>
</tr>
<tr>
<td>E0010</td>
<td>Standard Report – Subheading above Columns Window</td>
<td></td>
</tr>
</tbody>
</table>
### Work with File Specifications

<table>
<thead>
<tr>
<th>PROGRAM TYPE</th>
<th>DESCRIPTION</th>
<th>SPECIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B0010</td>
<td>Single Record Maintenance</td>
<td>Specify the master file with an M or 1 in the Update field</td>
</tr>
<tr>
<td>D0040</td>
<td>SFL Maintenance — KEY</td>
<td></td>
</tr>
<tr>
<td>D0045</td>
<td>SFL Maintenance — KEY, No Action Code</td>
<td></td>
</tr>
<tr>
<td>D0060</td>
<td>SFL Maintenance — KEY</td>
<td></td>
</tr>
<tr>
<td>X0010</td>
<td>Batch Update — 1 File</td>
<td></td>
</tr>
<tr>
<td>Y0020</td>
<td>File Conversion — 1 File</td>
<td></td>
</tr>
<tr>
<td>Y0030</td>
<td>File Conversion — 1 File</td>
<td></td>
</tr>
<tr>
<td>D0100</td>
<td>SFL Maintenance — KEY, 2 Update Files</td>
<td>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.</td>
</tr>
<tr>
<td>D0010</td>
<td>SFL Maintenance — RRN</td>
<td>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.</td>
</tr>
<tr>
<td>D0020</td>
<td>SFL Maintenance — RRN</td>
<td>Specify the logical file which 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.</td>
</tr>
<tr>
<td>D0030</td>
<td>No Action code</td>
<td>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.</td>
</tr>
<tr>
<td>D0070</td>
<td>SFL Maintenance — RRN</td>
<td>Specify the logical file which 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.</td>
</tr>
<tr>
<td>D0070</td>
<td>No Action code</td>
<td>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.</td>
</tr>
<tr>
<td>D0080</td>
<td>SFL Maintenance — RRN</td>
<td></td>
</tr>
<tr>
<td>D0090</td>
<td>SFL Maintenance — RRN</td>
<td></td>
</tr>
<tr>
<td>D0090</td>
<td>SFL Maintenance — RRN</td>
<td></td>
</tr>
<tr>
<td>D0090</td>
<td>SFL Maintenance — RRN, No Action code</td>
<td></td>
</tr>
<tr>
<td>D0050</td>
<td>SFL Maintenance — RRN, 2 Update Files</td>
<td>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.</td>
</tr>
<tr>
<td>X0020</td>
<td>Batch Update, 2 Files</td>
<td>Specify the input file with a 1 in the Input field. Specify the output file with a 2 in the Update field.</td>
</tr>
<tr>
<td>X0030</td>
<td>Batch Update, 2 Files</td>
<td></td>
</tr>
<tr>
<td>Y0010</td>
<td>File Conversion, 2 Files</td>
<td></td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Specifications F93102</td>
<td>The system updates this file with one record for each file in the File Specification.</td>
</tr>
<tr>
<td></td>
<td>- If the master file includes a Business Unit field, then the system adds the Business Unit Security file (F0001) to the File Specifications.</td>
</tr>
<tr>
<td></td>
<td>- 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.</td>
</tr>
<tr>
<td>Data Base Format Parameters F93103</td>
<td>The system updates this file with one record for each format in each file.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Detail Program Logic Parameters F93105</td>
<td>The system updates this file with one record for each field in each file. The system uses the records in the Detailed Programming Facility.</td>
</tr>
<tr>
<td></td>
<td>- If the file is a master file or device file, then the system includes all fields.</td>
</tr>
<tr>
<td></td>
<td>- If the file is a database file that you use only for input purposes, then the system includes only the key fields.</td>
</tr>
</tbody>
</table>

**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 Specifications

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>The member ID of the file used by the program.</td>
</tr>
<tr>
<td>Input</td>
<td>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 Primary input file. The “P” will generate the F specification as input primary S Secondary input file. The “S” designates input secondary. X Input file. Any master file designation or an “X” will generate the RPG file (F) specification as input full procedural.</td>
</tr>
</tbody>
</table>

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. |

| 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. |
## Work with File Specifications

### Field | Explanation
--- | ---
**Update** | 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:
- M or 1 thru 9 — Update master file
- P — Update primary file
- S or X — Update secondary file
- T — Update transaction file

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.

**Add** | A code of X specifies that a file will have records written to it in the program being generated.

The data file designated as the master file in all file maintenance programs must be designated as allowing file additions.

A code of X will generate an A in column 66 of the file (F) specification in RPG.

**Src Lib/File** | The Library Name field contains the name of a valid AS/400 library name. Defaults from SVR.

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.

**Keyed(Y/N)** | 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.

- If processing by RRN, the physical file that is being updated must be specified as keyed = N.

**File Info DS** | Name assigned to an RPG III file information data structure if needed for an associated data file.

- 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

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PF Src Lib/File</td>
<td>Library where the source resides for the physical file linked to the logical file.</td>
</tr>
<tr>
<td>External(Y/N)</td>
<td>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. 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.</td>
</tr>
</tbody>
</table>

### 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:

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

<table>
<thead>
<tr>
<th>Character</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| ~         | 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
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
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function key exits or subfile options.</td>
</tr>
</tbody>
</table>

**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
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Id</td>
<td>The identification, such as program number, table number, and report number, this is assigned to an element of software.</td>
</tr>
<tr>
<td></td>
<td><strong>Screen-specific information</strong></td>
</tr>
<tr>
<td></td>
<td>The name of the program that the system executes when you choose the function exit or enter a selection option value.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Key</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Screen-specific information</strong></td>
</tr>
<tr>
<td></td>
<td>You can only define function exits for #F01 through #F15 and subfile options for #S01 through #S16.</td>
</tr>
<tr>
<td>Parm 1</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 2</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 3</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 4</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 5</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 6</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 7</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Parm 8</td>
<td>The RPG field name (6 bytes) to be passed as a parameter on function exits or subfile options.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Purpose of Exit</td>
<td>A name or remark that describes an element in the JD Edwards World systems.</td>
</tr>
<tr>
<td>Screen-specific information</td>
<td>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.</td>
</tr>
<tr>
<td>Returned Key Fld</td>
<td>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.</td>
</tr>
<tr>
<td>Screen-specific information</td>
<td>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</td>
</tr>
<tr>
<td>Returned Desc Fld</td>
<td>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.</td>
</tr>
<tr>
<td>Screen-specific information</td>
<td>Causes logic generation to let a returned description pass through the local data area and loads the value to the designated description field.</td>
</tr>
<tr>
<td></td>
<td>- Only valid with the CL program J98LDAKY</td>
</tr>
<tr>
<td></td>
<td>- For more information on using the Returned Key and Returned Desc Fld, see the program level Helps for P93104</td>
</tr>
</tbody>
</table>

**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 SHxxxx**
  
  You can define the SHxxxx fields as hidden fields on their screen and then load them with the proper information using theDetailed 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 KLST 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Name</td>
<td>The member ID of the file used by the program.</td>
</tr>
<tr>
<td>O P</td>
<td>Allows for selection exits for each field.</td>
</tr>
</tbody>
</table>
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

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

**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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source of Data</td>
<td>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 “<em>SKIP”. (An entry of “</em>” 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.</td>
</tr>
<tr>
<td>Dictionary Name</td>
<td>The 4 character data item name from the data dictionary. Used extensively for field editing within the program generator.</td>
</tr>
<tr>
<td>Field Type</td>
<td>Used to designate master file field names and display/report file field names within the data field parameter records.</td>
</tr>
<tr>
<td></td>
<td>M indicates a master file field</td>
</tr>
<tr>
<td></td>
<td>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)</td>
</tr>
<tr>
<td></td>
<td>S indicates the field is in the subfile portion of a video</td>
</tr>
<tr>
<td></td>
<td>D indicates a field within a report detail format</td>
</tr>
<tr>
<td></td>
<td>H indicates a field within a report heading format</td>
</tr>
<tr>
<td></td>
<td>T indicates a field within a report total format</td>
</tr>
<tr>
<td>Data Field Use</td>
<td>To determine how a data item is used on a video screen or report as far as:</td>
</tr>
<tr>
<td></td>
<td>I input only</td>
</tr>
<tr>
<td></td>
<td>O output only</td>
</tr>
<tr>
<td></td>
<td>B both input and output</td>
</tr>
<tr>
<td></td>
<td>H hidden field</td>
</tr>
<tr>
<td>Key Position</td>
<td>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.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PLIST Sequence</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>- 01 - 32 are valid</td>
</tr>
<tr>
<td></td>
<td>- Must enter as 01 and not 1</td>
</tr>
<tr>
<td></td>
<td>- If the first parameter is passed a non-blank value, an auto-inquiry will be performed</td>
</tr>
<tr>
<td>Entry Optional</td>
<td>Used with subfile maintenance programs to identify the field that controls database updates.</td>
</tr>
<tr>
<td></td>
<td>- One field needs to be designated as Entry Optional: N</td>
</tr>
<tr>
<td></td>
<td>- Defaults to a blank</td>
</tr>
<tr>
<td>Clear After (Y/N)</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Y indicates the field will be cleared at the end of each transaction entry. The default is Y.</td>
</tr>
<tr>
<td></td>
<td>N indicates the field will not be cleared unless specified by the user by pressing the appropriate function key.</td>
</tr>
<tr>
<td>Right Adj (Y/N)</td>
<td>A code of:</td>
</tr>
<tr>
<td></td>
<td>Y indicates the field should be right adjusted.</td>
</tr>
<tr>
<td></td>
<td>N indicates the field should NOT be right adjusted.</td>
</tr>
<tr>
<td></td>
<td>C indicates the field is a business unit and should be left filled with blanks instead of zeros.</td>
</tr>
<tr>
<td></td>
<td>A indicates the field is an account number and the account number edit routine will be used for editing.</td>
</tr>
<tr>
<td></td>
<td>Can only be used when the Read From field is a video field and the Write To field is a data base field.</td>
</tr>
<tr>
<td>Center (Y/N)</td>
<td>A code of Y will center the data within the field when it is displayed.</td>
</tr>
<tr>
<td>Description File</td>
<td>Used in conjunction with loading a VC0 description field.</td>
</tr>
<tr>
<td></td>
<td>- Identifies the file that contains the description</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Descr. File Key</td>
<td>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. 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). 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).</td>
</tr>
</tbody>
</table>
| Dictionary Edit     | Controls the generation of data dictionary editing for fields in the master file.  
- 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        | Identifies a custom error message to use when errors are detected on a screen field.  
- Loads the value in array EMK of subroutine S9999 |
| Validation File     | Specifies the file name to use for validating the current data field contents. This file name is automatically provided from the data dictionary if it exists. |
| Error Indicator     | Used to designate the error controlling indicator for a data item on a video screen. This indicator controls the standard error notification attributes for video screens (reverse image, high intensity and position cursor). |
| Error Index         | The Error Message Index field is the array index where a special error message number is loaded in the error message array. Each of the data item parameters which uses external file validation can override the standard error message (0002). A new index must be entered for these types of changes. Error indexes 1 through 20 are reserved for the program generator. Error indexes 21 through 30 are reserved for file validation. Error indexes 30 through 64 can be used for anything else. |
| Data Item Type      | This defines the type of data to be stored in the field. The data item types are defined in User Defined Codes, system code ‘98’, record type ‘DT’. Note: All amount fields should be entered as 15 bytes, 0 decimals, and data item type should be P (packed). |
Work with the Detailed Programming Facility

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Item Size</td>
<td>The field size of the data item.</td>
</tr>
<tr>
<td></td>
<td>NOTE: All amount fields should be entered as 15 bytes, 0 decimals, and the data item type should be P (packed).</td>
</tr>
<tr>
<td>Decimal Pos</td>
<td>The number of positions to the right of the decimal of the data item.</td>
</tr>
<tr>
<td>Edit Code</td>
<td>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):</td>
</tr>
<tr>
<td></td>
<td>▪ Show commas - 1, 2, A, B, J, K, N, or O</td>
</tr>
<tr>
<td></td>
<td>▪ Show decimal point - 1, 2, 3, 4, A, B, C, D, J, K, L, M, N, O, P, Q</td>
</tr>
<tr>
<td></td>
<td>▪ Show sign for negative - A, B, C, D (“CR”) or J through Q (“-”)</td>
</tr>
<tr>
<td></td>
<td>▪ Suppress leading zeros - 1 through 4, A through D, J through Q, Y and Z</td>
</tr>
<tr>
<td></td>
<td>Refer to user defined codes (system 98/ type EC) for all valid codes, including additional JD Edwards World edit codes.</td>
</tr>
</tbody>
</table>

**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.  
- 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.

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).

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).
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.
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 S9999. 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.

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: MOVEL @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, \n    \ use Processing Option as default \n    begin
Define Processing Options

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 followings:

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq</td>
<td>Specifies how the processing option text lines should be ordered on the screen. Not input capable.</td>
</tr>
<tr>
<td>Text</td>
<td>The descriptive text for the processing option.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Opt Nbr</td>
<td>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. Screen-specific information 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.</td>
</tr>
<tr>
<td>Date (0/1)</td>
<td>The Date Field specifies whether or not the processing option refers to a date. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>0 Indicates that the information is not a date.</td>
</tr>
<tr>
<td></td>
<td>1 Indicates that a date is to be stored in the processing option as a Gregorian date in month, day and year format.</td>
</tr>
<tr>
<td></td>
<td>2 Indicates that a date is to be stored in the processing option as a Julian date in century, year and day format.</td>
</tr>
<tr>
<td></td>
<td>3 Indicates the same as a “2” with the exception that the display AND entry format is “YYYY/MM/DD” (full four digit year).</td>
</tr>
<tr>
<td></td>
<td>NOTE: All data entry for date information is entered in SYSTEM FORMAT with the exception of the “3”.</td>
</tr>
<tr>
<td>R J(Right Justify)</td>
<td>Determines if the entry field is right-justified. Valid values are:</td>
</tr>
<tr>
<td></td>
<td>0 Information is not right-justified</td>
</tr>
<tr>
<td></td>
<td>1 Information to be entered is numeric and should be right-justified</td>
</tr>
<tr>
<td></td>
<td>2 Information to be entered is to be right-justified and left-filled with blanks</td>
</tr>
<tr>
<td>Text Only</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>1 for text only</td>
</tr>
<tr>
<td></td>
<td>0 for a value entry line.</td>
</tr>
<tr>
<td></td>
<td>Each separate processing option can have only one input value, or “0” value.</td>
</tr>
</tbody>
</table>
Define Processing Options

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>D L (Display Level)</td>
<td>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.</td>
</tr>
</tbody>
</table>

Field Name: The internal field name assigned to each option and function key.

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

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

- This is a required field
- Use #S01 - #S15 for options
- Use #F01 - #F15 for function keys

**Program-specific information**

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.

Special characters are not allowed as part of the data item name, with the exception of #, @, $.

If you want to create protected data names without JD Edwards World interference, use $xxx and @xxx, with xxx being user-defined.

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.

Used to validate against the data dictionary.

**Screen-specific information**

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.

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.
Define 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
Overview to Program Design Language

- 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
About PDL Statements and Syntax

- 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

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| 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. \n
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. \n```
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

About Comments

Keywords and Syntax

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>\ (backslash)</td>
<td>Initiates and terminates a comment. The syntax is: \ text \</td>
</tr>
<tr>
<td></td>
<td>You must enclose all comments within a pair of backslashes.</td>
</tr>
</tbody>
</table>

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

About Assignments

Operator and Syntax in as

<table>
<thead>
<tr>
<th>Operators</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>:=</td>
<td>The assignment operator. The system assigns the first variable the value of the variable or expression following the operator. The syntax is: variable := expression;</td>
</tr>
<tr>
<td>+</td>
<td>Add</td>
</tr>
<tr>
<td>-</td>
<td>Subtract</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&gt;</td>
</tr>
<tr>
<td></td>
<td>&lt;</td>
</tr>
<tr>
<td>SST</td>
<td>Substring</td>
</tr>
<tr>
<td></td>
<td>The syntax is: variable := SST (field,n1,n2)</td>
</tr>
<tr>
<td></td>
<td>n1 = start position</td>
</tr>
<tr>
<td></td>
<td>n2 = length of string</td>
</tr>
</tbody>
</table>

Rules

You can use standard notation using parentheses for arithmetic operations.

For example:
in98 := ‘0’;
vdremk := ‘NOT DEFINED’;
About PDL Statements and Syntax

sftrdj := $$edt;

$#am1 := $#am1 + (qzqty * qzcsl);

$#wrk := 100;

abalph := vd#fnm |> vd#lnm;

$cc := SST (qxxcc,3,10)

About Database Operations

Keywords and Syntax

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chain</td>
<td>Provides for random data base processing. The syntax is: CHAIN file;</td>
</tr>
<tr>
<td>Delete</td>
<td>Provides the ability to delete the current data base record. The syntax is: DELETE file;</td>
</tr>
<tr>
<td>Poseq</td>
<td>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;</td>
</tr>
<tr>
<td>Posgt</td>
<td>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;</td>
</tr>
<tr>
<td>Read</td>
<td>Provides for sequential data base processing by reading the next record in the file you designate. The syntax is: READ file;</td>
</tr>
<tr>
<td>Readc</td>
<td>Provides for processing of workstation files to obtain the next record change in a subfile. They syntax is: READC file;</td>
</tr>
<tr>
<td>Reade</td>
<td>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;</td>
</tr>
<tr>
<td>Readp</td>
<td>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;</td>
</tr>
<tr>
<td>Update</td>
<td>Provides the ability to update the current data base record. The syntax is: UPDATE file;</td>
</tr>
<tr>
<td>Write</td>
<td>Provides the ability to add a new data base records. The syntax is: WRITE file;</td>
</tr>
</tbody>
</table>
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. \n
Begin
  aban8 := q3an8;
  chain f0101la;
  If in98 = '0' Then
    vc0003 := abalph;
  End

About Program Calls

Keywords and Syntax

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
<td>Allows you to execute another program. The syntax is: CALL variable;</td>
</tr>
<tr>
<td>Parm</td>
<td>Allows you to deliver parameters to a program that the program call statement executes. The syntax is: PARM variable;</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Until</td>
<td>Provides for loop processing where the system evaluates a condition at the bottom of the loop.</td>
</tr>
<tr>
<td></td>
<td>- Translates to DOU in the RPG code.</td>
</tr>
<tr>
<td></td>
<td>The syntax is: UNTIL (condition) DO (Statement)</td>
</tr>
<tr>
<td>While</td>
<td>Provides for loop processing where the system evaluates a condition at the top of the loop.</td>
</tr>
<tr>
<td></td>
<td>- Translates to DOW in the RPG code.</td>
</tr>
<tr>
<td></td>
<td>The syntax is: WHILE (condition) DO (Statement)</td>
</tr>
<tr>
<td>Do</td>
<td>An integral part of the loop statement.</td>
</tr>
</tbody>
</table>

#### 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
```
About PDL Statements and Syntax

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

End

End

About Conditions

Keywords and Syntax

<table>
<thead>
<tr>
<th>Keywords</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| **If**   | Provides for conditional processing.  
|          | - 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

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>=</td>
<td>Equal</td>
</tr>
<tr>
<td>≠</td>
<td>Not Equal</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater Than</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less Than</td>
</tr>
<tr>
<td>≥</td>
<td>Greater Than or Equal To</td>
</tr>
<tr>
<td>≤</td>
<td>Less Than or Equal To</td>
</tr>
</tbody>
</table>
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';
         $#nim := 0;
      End;
   Else
      If zaclst < '900' Then

About PDL Statements and Syntax

Begin

   rr nin := ‘<0’;
   $nin := 1-

End;

Else

Begin

   rr nin := ‘>0’;
   $nin := 1;

End

End

About Miscellaneous Keywords and Syntax

Keywords and Syntax

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

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:

```plaintext
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.

<table>
<thead>
<tr>
<th>Read From</th>
<th>Write To</th>
<th>Affected</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) QXXIT</td>
<td>VDXIT</td>
<td>S004 (moves database fields to video fields)</td>
</tr>
<tr>
<td>2) VDXIT</td>
<td>QXXIT</td>
<td>S005 (moves video fields to data base fields)</td>
</tr>
</tbody>
</table>
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:

<table>
<thead>
<tr>
<th>Type of PDL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>$sel := '0';</td>
</tr>
<tr>
<td>Blocks</td>
<td>begin...end</td>
</tr>
<tr>
<td>Comment</td>
<td>\ Test order number for inclusion. \</td>
</tr>
<tr>
<td>Condition</td>
<td>If q1doco &lt; $doco1 Then $sel := '0';</td>
</tr>
</tbody>
</table>

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.

![Data Item Formula](image)

This example illustrates the following types of PDL statements:
<table>
<thead>
<tr>
<th>Type of PDL Statement</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignment</td>
<td>$#ami := 0; $#xtp := 0; q2doco := gldoco;</td>
</tr>
<tr>
<td>Blocks</td>
<td>Notice the begin...end nested within the while...do</td>
</tr>
<tr>
<td>Comments</td>
<td>Notice the embedded comments as well as the heading comment</td>
</tr>
<tr>
<td>Database</td>
<td>poseq f59422; reade f59422;</td>
</tr>
<tr>
<td>Loops</td>
<td>while in98 = ‘0’ do begin...end;</td>
</tr>
</tbody>
</table>

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 in 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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program ID</td>
<td>The RPG program name defined in the Software Versions Repository Master table.</td>
</tr>
<tr>
<td>File ID</td>
<td>The member ID of the file used by the program.</td>
</tr>
<tr>
<td>Field Name</td>
<td>This specifies the field name as it is identified in the file.</td>
</tr>
<tr>
<td>Data Item Formula</td>
<td>A set of Program Design Language (PDL) statements, which are then translated into RPG code.</td>
</tr>
</tbody>
</table>

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

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

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

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


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.
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 AGSRC 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 followng:
<table>
<thead>
<tr>
<th>Possible Resolve</th>
<th>Description</th>
</tr>
</thead>
</table>
| Ensure the CAP status is set to Y on the Program Purpose and Type screen. | Any job that prevents the MPxxxxx job from completing normally will change the CAP Status to N.  
  - Allow the MPxxxxx job to complete.  
  - Do not cancel it in the job queue.  
  If you change the CAP Status 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).  
  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. |
| Ensure the Pxxxxx member exists in the Additional Help/Modifications Master file (F93002). |  
  - The Pxxxxx member must exist in order to generate a program.  
  - The system initially creates the Pxxxxx member during the Program Purpose and Type definition step. |
| Ensure the Mxxxxx member does not exist in F93002. |  
  - The Mxxxxx member must not exist in order to generate a program.  
  - Use the RMVM command to remove this member. |
| Ensure that one step of the generation process completes before you start the batch job of another step. |  
  - 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. |
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.

<table>
<thead>
<tr>
<th>Model CL Programs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J98MODEL1</td>
<td>Serves as a template for all interactive programs that do not retrieve processing options in the CL code.</td>
</tr>
<tr>
<td>J98MODEL2</td>
<td>Serves as a template for batch programs that need the DREAM Writer but have no printer file.</td>
</tr>
</tbody>
</table>
### Model CL Programs

<table>
<thead>
<tr>
<th>Model CL Programs</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>J98MODEL3</td>
<td>Serves as a template for interactive programs that need a prompt for parameters.</td>
</tr>
<tr>
<td>J98MODEL4</td>
<td>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.</td>
</tr>
<tr>
<td>J98MODEL5</td>
<td>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.</td>
</tr>
<tr>
<td>J98MODEL6</td>
<td>Serves as a template for batch CL programs that require all DREAM Writer functions.</td>
</tr>
<tr>
<td>J98MODEL7</td>
<td>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.</td>
</tr>
<tr>
<td>J98MODEL8</td>
<td>Serves as a template for batch programs that have a control file.</td>
</tr>
</tbody>
</table>

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.
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:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC1ROW</td>
<td>Prints the data dictionary row description of the level break field.</td>
</tr>
<tr>
<td></td>
<td>Default length is 30.</td>
</tr>
<tr>
<td>VC1KEY</td>
<td>Prints the value of the level break field. Default length is 12.</td>
</tr>
<tr>
<td>VC1DSC</td>
<td>Prints the description of the value of the break field. Default length is 30.</td>
</tr>
<tr>
<td></td>
<td>Only works with the following fields:</td>
</tr>
<tr>
<td></td>
<td>▪ User defined codes</td>
</tr>
<tr>
<td></td>
<td>▪ Company Number</td>
</tr>
<tr>
<td></td>
<td>▪ Address Book Number</td>
</tr>
<tr>
<td></td>
<td>▪ Business Unit</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC0ROW</td>
<td>Prints the data dictionary row description of the level break field.</td>
</tr>
<tr>
<td></td>
<td>Default length is 30.</td>
</tr>
<tr>
<td>VC0KEY</td>
<td>Prints the value of the level break field. Default length is 12.</td>
</tr>
</tbody>
</table>
### Create Report Programs

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC0DSC</td>
<td>Prints the description of the value of the break field. Default length is 30. Only works with the following fields:</td>
</tr>
<tr>
<td></td>
<td>▪ User defined codes</td>
</tr>
<tr>
<td></td>
<td>▪ Company Number</td>
</tr>
<tr>
<td></td>
<td>▪ Address Book Number</td>
</tr>
<tr>
<td></td>
<td>▪ Business Unit</td>
</tr>
</tbody>
</table>

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).
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.

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

8. Click Enter twice.

   The description for the total field replaces the asterisk (*).
In the sample report that follows, when you print the report, the field contains the descriptive text Business Unit.

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

The Field Definition screen displays.

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

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:
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.

### 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 VCI 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

<table>
<thead>
<tr>
<th>Opt</th>
<th>Format Name</th>
<th>Type</th>
<th>Data Base File</th>
<th>Start / End Lines</th>
<th>Related Record</th>
<th>No. Flds</th>
<th>Fld Pfx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HEADING1</td>
<td>REPORT</td>
<td>001 008</td>
<td></td>
<td></td>
<td>000</td>
<td>RR</td>
</tr>
<tr>
<td>2</td>
<td>DETAIL1</td>
<td>REPORT</td>
<td>009 009</td>
<td></td>
<td></td>
<td>000</td>
<td>RR</td>
</tr>
<tr>
<td>3</td>
<td>TOTAL1</td>
<td>REPORT</td>
<td>010 012</td>
<td></td>
<td></td>
<td>000</td>
<td>SS</td>
</tr>
<tr>
<td>4</td>
<td>HEADING2</td>
<td>REPORT</td>
<td>013 013</td>
<td></td>
<td></td>
<td>000</td>
<td>RR</td>
</tr>
</tbody>
</table>

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.
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.

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:

```
The Organization's Name
Program ID . . P928400
Version . . . 002
Inventory by Business Unit Report
Report Date . . 12/02/17
San Francisco Branch
Report Time . . 9:35:50
Additional Line of Text if Required
```

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

The field names for report headings are similar to those of the cover page. For report headings, the VC0CO field contains the name of the company. The DREAM Writer fields have an RR prefix. For example, RRTTTL@ 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.

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

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

To create a program using the Quick Start CL Generator

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

2. Perform one of the following:
   - Click Exit (F3) to return to the menu.
   - Click Enter to compile the program.
**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.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object Library Name</td>
<td>Enter the name of the object library where the program you are creating will reside. This is defaulted from the CASE Profiles.</td>
</tr>
</tbody>
</table>

**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).

**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
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Use this field to enter a short one-line description of the program you are creating.</td>
</tr>
<tr>
<td>Program Name</td>
<td>Type the name of the RPG program that the CL program will call. This is a required field. Do not leave it blank.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>Create Program (Y/N)?</td>
<td>Indicate in this field whether you want to create the RPG program.</td>
</tr>
<tr>
<td>Screen or Report Name</td>
<td>Type the screen or report file name associated with the program. This field is only required for program type 2.</td>
</tr>
<tr>
<td>Select a Program Type (1-4)</td>
<td>Type one of the following in this field to indicate the type of program you are creating:</td>
</tr>
<tr>
<td></td>
<td>1 Standard Interactive Program</td>
</tr>
<tr>
<td></td>
<td>2 Standard Report Program with DREAM Writer</td>
</tr>
<tr>
<td></td>
<td>3 File Processor</td>
</tr>
<tr>
<td></td>
<td>4 File Processor with DREAM Writer</td>
</tr>
<tr>
<td>Action Code</td>
<td>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.</td>
</tr>
<tr>
<td>Selection Option</td>
<td>Enter Y if you are creating a screen with a selection option.</td>
</tr>
<tr>
<td>Report Detail Subheadings</td>
<td>Enter Y if you are creating a report with detail subheadings.</td>
</tr>
<tr>
<td>Report Total Subheadings</td>
<td>Enter Y if you are creating a report with total subheadings.</td>
</tr>
<tr>
<td>Report Totals</td>
<td>Enter Y if you are creating a report with totals.</td>
</tr>
<tr>
<td>Master File</td>
<td>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.</td>
</tr>
<tr>
<td>Library Name</td>
<td>Type in the name of the library your master file is in.</td>
</tr>
<tr>
<td>Detail File (optional)</td>
<td>Type in the name of an optional secondary file from which you want to select data.</td>
</tr>
<tr>
<td>Library Name</td>
<td>Type in the name of the library your secondary file is in.</td>
</tr>
<tr>
<td>Source File Name</td>
<td>The member ID of the file used by the program.</td>
</tr>
<tr>
<td>Source Library Name</td>
<td>File and library that contains the file source.</td>
</tr>
<tr>
<td>Object Library Name</td>
<td>Enter the name of the object library where the program you are creating will reside. This is defaulted from the CASE Profiles.</td>
</tr>
</tbody>
</table>
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

From Action Diagramming (G9363), choose Build Action Diagrams

On Build Action Diagram, choose a version.
Work with Action Diagramming

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.

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lvl/Sbr</td>
<td>Specifies the logic level and subroutine.</td>
</tr>
<tr>
<td>Program ID</td>
<td>The program name for the action diagram.</td>
</tr>
<tr>
<td>Scan</td>
<td>Allows the user to search for specific information.</td>
</tr>
</tbody>
</table>

The use of colors, arrows, indentation, and connecting vertical lines indicates the hierarchy and relationships of the code within the program.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>===&gt;</td>
<td>Signals the beginning or ending of a loop.</td>
</tr>
<tr>
<td>--&gt;&gt;</td>
<td>Signals an IF or WHEN statement or their associated end statement.</td>
</tr>
<tr>
<td>Blank</td>
<td>Labels are in reverse image.</td>
</tr>
</tbody>
</table>

**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.

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

**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:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary Key</td>
<td>The primary key represents source code lines that come from a Primary Logic Module. The primary key begins in column 80.</td>
</tr>
<tr>
<td>Secondary Key</td>
<td>The secondary key represents the source code lines that come from a Detail Logic module. The secondary key begins in column 90.</td>
</tr>
<tr>
<td>Serial Number</td>
<td>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.</td>
</tr>
<tr>
<td>User ID</td>
<td>When the program generator creates a program, it places the User ID of the program’s creator within the source sequence line.</td>
</tr>
<tr>
<td>SAR Number</td>
<td>When the program generator creates a program, it places the SAR Number, if available, within the source sequence line.</td>
</tr>
<tr>
<td>Element</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Date Last Change</td>
<td>When the program generator creates a program, it places the date you add or change the code within the source sequence line.</td>
</tr>
</tbody>
</table>

The following illustrates the parts of the source sequence line.

<table>
<thead>
<tr>
<th>Primary Key</th>
<th>Secondary Key</th>
<th>Serial Number</th>
<th>User ID</th>
<th>SAR Number</th>
<th>Date Last Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070007000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070008000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070009000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070010000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070011000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070012000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
<tr>
<td>S999-4</td>
<td>RR#BEN</td>
<td>0070013000000QUARLES</td>
<td>721561</td>
<td>000000</td>
<td></td>
</tr>
</tbody>
</table>

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.
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.

```markdown
From Model Program Design Menu (G9361), under PROGRAM TYPES, choose Index
```
Working with Program Types

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.
Working with Program Types

- 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.
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**

---

![Logic Module Cross Reference Window]

**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 accidently 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**

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

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

On Parameter Copy/Move complete the following fields:

- Program Generator Specs
- Data Dictionary
- Processing Option
- From Program ID
- To Program ID
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.
Work with Logic Modules

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:
Work with Logic Modules

- 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:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>*ACTN</td>
<td>None</td>
<td>S999</td>
<td>Load action code lock out array</td>
</tr>
<tr>
<td>*ATOT</td>
<td>XADDTOT1</td>
<td>S010</td>
<td>Accumulate report total logic</td>
</tr>
<tr>
<td>AUTHR</td>
<td>None</td>
<td>F spec</td>
<td>Program author</td>
</tr>
<tr>
<td>*AUTOI</td>
<td>X*ENTRYI</td>
<td>S999</td>
<td>Automatic inquiry at execution test logic</td>
</tr>
<tr>
<td>*CLRN</td>
<td>None</td>
<td>S001</td>
<td>Clear user requested fields</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>CLRY</td>
<td>None</td>
<td>S001</td>
<td>Clear all data fields for next transaction</td>
</tr>
<tr>
<td>CLSFL</td>
<td>None</td>
<td>S003</td>
<td>Clear all subfile fields</td>
</tr>
<tr>
<td>COPY</td>
<td>XCOPY-SUB</td>
<td>Various</td>
<td>RPGIII copy function for common subroutines</td>
</tr>
<tr>
<td>CTOT</td>
<td>XCLRTOT1</td>
<td>S010</td>
<td>Clear report totals</td>
</tr>
<tr>
<td>*DATES</td>
<td>XDSDATE</td>
<td>I Spec</td>
<td>Data structures for Gregorian dates (not using record buffer)</td>
</tr>
<tr>
<td>*DATER</td>
<td>None</td>
<td>I Spec</td>
<td>Data structures for Gregorian dates in the record buffer #BUFIN</td>
</tr>
<tr>
<td>DESC</td>
<td>None</td>
<td>F Spec</td>
<td>File or program description</td>
</tr>
<tr>
<td>*DPARM</td>
<td>XFIELDVAL</td>
<td>S998</td>
<td>Retrieve all Data Dictionary values for videos</td>
</tr>
<tr>
<td>*DPRMS</td>
<td>XFIELDVAL2</td>
<td>S998</td>
<td>Retrieve Data Dictionary values for detail subheading reports</td>
</tr>
<tr>
<td>*DPRMR</td>
<td>XFIELDVAL</td>
<td>S998</td>
<td>Retrieve Data Dictionary values for total subheading reports</td>
</tr>
<tr>
<td>DSPF</td>
<td>None</td>
<td>Various</td>
<td>Variable name substitution for display file(s) fields</td>
</tr>
<tr>
<td>DSP1</td>
<td>XDSPFLD1</td>
<td>S004</td>
<td>Display logic for primary video fields</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD2</td>
<td>S004</td>
<td>Format Alpha field for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD3</td>
<td>S004</td>
<td>Format Gregorian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD4</td>
<td>S004</td>
<td>Format Julian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD5</td>
<td>S004</td>
<td>Format VC0 field from VTX</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD6</td>
<td>S004</td>
<td>Format VC0 field from designated description file (field details)</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD7</td>
<td>S004</td>
<td>Format Alpha 3 or 28</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD8</td>
<td>S004</td>
<td>Repeat of XDSPFLD1</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>XDSPFLD1</td>
<td>S004</td>
<td>Display logic for primary video fields</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD2</td>
<td>S004</td>
<td>Format Alpha field for output</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD3</td>
<td>S004</td>
<td>Format Gregorian Date for output</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD4</td>
<td>S004</td>
<td>Format Julian Date for output</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD5</td>
<td>S004</td>
<td>Format VC0 field from VTX</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD6</td>
<td>S004</td>
<td>Format VC0 field from designated description file (field details)</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD7</td>
<td>S004</td>
<td>Format Alpha 3 or 28</td>
<td></td>
</tr>
<tr>
<td>XDSPFLD8</td>
<td>S004</td>
<td>Repeat of XDSPFLD1</td>
<td></td>
</tr>
<tr>
<td>XLOADEMK</td>
<td>S999</td>
<td>Load user defined error messages</td>
<td></td>
</tr>
<tr>
<td>XENTRY</td>
<td>X*ENTRYP</td>
<td>Various Load program execution passed parameters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>X*ENTRYM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XEXIT-CMD0</td>
<td>S00EX</td>
<td>Function key exit execution logic</td>
<td></td>
</tr>
<tr>
<td>XEXIT-CMD1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XEXIT-SEL0</td>
<td>S000P</td>
<td>Selection exit execution logic</td>
<td></td>
</tr>
<tr>
<td>XEXIT-SEL0</td>
<td>S000P</td>
<td>Selection exit execution logic</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT1</td>
<td>S005</td>
<td>Active Data Dictionary field validation logic</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT2</td>
<td>S005</td>
<td>Data Dictionary alpha edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT3</td>
<td>S005</td>
<td>Validation n=Master - Alpha</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT4</td>
<td>S005</td>
<td>Gregorian edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT5</td>
<td>S005</td>
<td>Julian edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT6</td>
<td>S005</td>
<td>Data Dictionary numeric edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT7</td>
<td>S005</td>
<td>Alpha field size 10</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT8</td>
<td>S005</td>
<td>User defined code edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDT9</td>
<td>S005</td>
<td>No dictionary</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTA</td>
<td>S005</td>
<td>Validation n = Master - Numeric</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTC</td>
<td>S005</td>
<td>Account ID</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTE</td>
<td>S005</td>
<td>Cost center edit</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTR</td>
<td>S005</td>
<td>Numeric field size 7</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTS</td>
<td>S005</td>
<td>Right adjust</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTT</td>
<td>S005</td>
<td>Validation = Master - Alpha</td>
<td></td>
</tr>
<tr>
<td>XFIELDEDTU</td>
<td>S005</td>
<td>Validation = Master - Alpha Rt Adj</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Validation = Master - Numeric</td>
<td></td>
</tr>
<tr>
<td>FILES</td>
<td>None</td>
<td>F spec Program file descriptions</td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>*FLDxx</td>
<td></td>
<td></td>
<td>Active Data Dictionary field validation for primary data</td>
</tr>
<tr>
<td>XFIELDEDT1</td>
<td>S005</td>
<td></td>
<td>Data Dictionary alpha edit</td>
</tr>
<tr>
<td>XFIELDEDT2</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDEDT3</td>
<td>S005</td>
<td></td>
<td>Gregorian edit</td>
</tr>
<tr>
<td>XFIELDEDT4</td>
<td>S005</td>
<td></td>
<td>Julian edit</td>
</tr>
<tr>
<td>XFIELDEDT5</td>
<td>S005</td>
<td></td>
<td>Data Dictionary numeric edit</td>
</tr>
<tr>
<td>XFIELDEDT6</td>
<td>S005</td>
<td></td>
<td>Alpha field size 10</td>
</tr>
<tr>
<td>XFIELDEDT7</td>
<td>S005</td>
<td></td>
<td>User defined code edit</td>
</tr>
<tr>
<td>XFIELDEDT8</td>
<td>S005</td>
<td></td>
<td>No dictionary</td>
</tr>
<tr>
<td>XFIELDEDT9</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Numeric</td>
</tr>
<tr>
<td>XFIELDEDTA</td>
<td>S005</td>
<td></td>
<td>Account ID</td>
</tr>
<tr>
<td>XFIELDEDTC</td>
<td>S005</td>
<td></td>
<td>Cost center edit</td>
</tr>
<tr>
<td>XFIELDEDTE</td>
<td>S005</td>
<td></td>
<td>Numeric field size 7</td>
</tr>
<tr>
<td>XFIELDEDTR</td>
<td>S005</td>
<td></td>
<td>Right adjust</td>
</tr>
<tr>
<td>XFIELDEDTS</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDEDTT</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha Rt Adj</td>
</tr>
<tr>
<td>XFIELDEDTU</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Numeric</td>
</tr>
<tr>
<td>INFDS</td>
<td></td>
<td></td>
<td>File information data structures, if specified</td>
</tr>
<tr>
<td>XINFDS1</td>
<td>I spec</td>
<td></td>
<td>Standard database file information data structure. The field prefix is incremented from $1 to $x where x = number of files</td>
</tr>
<tr>
<td>XINFDS2</td>
<td>I spec</td>
<td></td>
<td>OBSOLETE. Use SRVFDS.</td>
</tr>
<tr>
<td>KEYI</td>
<td></td>
<td></td>
<td>Load master file key fields for inquiry programs.</td>
</tr>
<tr>
<td>XFIELDLD1</td>
<td>S003</td>
<td></td>
<td>Load video input - Alpha</td>
</tr>
<tr>
<td>XFIELDLD2</td>
<td>S003</td>
<td></td>
<td>Load video input - Numeric</td>
</tr>
<tr>
<td>XFIELDLD3</td>
<td>S003</td>
<td></td>
<td>Load video input - Cost Center</td>
</tr>
<tr>
<td>XFIELDLD4</td>
<td>S003</td>
<td></td>
<td>Load video input - Julian Date</td>
</tr>
<tr>
<td>XFIELDLD5</td>
<td>S003</td>
<td></td>
<td>Load video input - Gregorian Date</td>
</tr>
<tr>
<td>KEYS</td>
<td></td>
<td></td>
<td>Load master file key fields in subfile format.</td>
</tr>
<tr>
<td>XFIELDLD1</td>
<td>S003</td>
<td></td>
<td>Load video input - Alpha</td>
</tr>
<tr>
<td>XFIELDLD2</td>
<td>S003</td>
<td></td>
<td>Load video input - Numeric</td>
</tr>
<tr>
<td>XFIELDLD3</td>
<td>S003</td>
<td></td>
<td>Load video input - Cost Center</td>
</tr>
<tr>
<td>XFIELDLD4</td>
<td>S003</td>
<td></td>
<td>Load video input - Julian Date</td>
</tr>
<tr>
<td>XFIELDLD5</td>
<td>S003</td>
<td></td>
<td>Load video input - Gregorian Date</td>
</tr>
<tr>
<td>XNEXT-NBR</td>
<td>S003</td>
<td></td>
<td>Load video input - Next Numbering</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------</td>
<td>---------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>KEYS2</td>
<td></td>
<td></td>
<td>Load master file key fields in primary video format</td>
</tr>
<tr>
<td></td>
<td>XFIELDLD1</td>
<td>S005</td>
<td>Load video input - Alpha</td>
</tr>
<tr>
<td></td>
<td>XFIELDLD2</td>
<td>S005</td>
<td>Load video input - Numeric</td>
</tr>
<tr>
<td></td>
<td>XFIELDLD3</td>
<td>S005</td>
<td>Load video input - Cost Center</td>
</tr>
<tr>
<td></td>
<td>XFIELDLD4</td>
<td>S005</td>
<td>Load video input - Julian Date</td>
</tr>
<tr>
<td></td>
<td>XFIELDLD5</td>
<td>S005</td>
<td>Load video input - Gregorian Date</td>
</tr>
<tr>
<td></td>
<td>XNEXT-NBR</td>
<td>S005</td>
<td>Load video input - Next Numbering</td>
</tr>
<tr>
<td>KLIST</td>
<td>XKEYLIST</td>
<td>S999</td>
<td>Create data file key list</td>
</tr>
<tr>
<td>*LVLS</td>
<td>XSAVVAL1</td>
<td></td>
<td>Save report level break data</td>
</tr>
<tr>
<td>MF</td>
<td>None</td>
<td>Various</td>
<td>Variable name substitution for master database files</td>
</tr>
<tr>
<td>*MCUxx</td>
<td>None</td>
<td>S003</td>
<td>Business Unit security logic where xx = master field designation 1 thru 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>S004</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>S00EX</td>
<td></td>
</tr>
<tr>
<td>*OPEN</td>
<td>XFILEOPN1</td>
<td>S999</td>
<td>Open report program data files</td>
</tr>
<tr>
<td>OPTE</td>
<td>None</td>
<td>S005</td>
<td>Subfile processing condition test based on mandatory entry fields in subfile format</td>
</tr>
<tr>
<td>*OTOT</td>
<td>XPRTTOT1</td>
<td>S010</td>
<td>Print all report level totals</td>
</tr>
<tr>
<td>PDL</td>
<td>None</td>
<td>Various</td>
<td>User defined entry point</td>
</tr>
<tr>
<td>*RKYxx</td>
<td>None</td>
<td>S999</td>
<td>Load softcoding record key for reports where xx = master file designation 1 thru 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RPTD</td>
<td>XDSPFLD1</td>
<td>S004</td>
<td>Format data for report detail format</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD2</td>
<td>S004</td>
<td>Format Alpha field for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD3</td>
<td>S004</td>
<td>Format Gregorian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD4</td>
<td>S004</td>
<td>Format Julian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD5</td>
<td>S004</td>
<td>Format VC0 field from VTX</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD6</td>
<td>S004</td>
<td>Format VC0 field from description file (field details)</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD7</td>
<td>S004</td>
<td>Format Alpha 3 or 28</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD8</td>
<td>S004</td>
<td>Repeat of XDSPFLD1</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>----------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>RPTH</td>
<td>XDSPFLD1</td>
<td>S004</td>
<td>Format data for report heading format</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD2</td>
<td>S004</td>
<td>Format Alpha field for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD3</td>
<td>S004</td>
<td>Format Gregorian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD4</td>
<td>S004</td>
<td>Format Julian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD5</td>
<td>S004</td>
<td>Format VC0 field from VTX</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD6</td>
<td>S004</td>
<td>Format VC0 field from description file (field details)</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD7</td>
<td>S004</td>
<td>Format Alpha 3 or 28</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD8</td>
<td>S004</td>
<td>Repeat of XDSPFLD1</td>
</tr>
<tr>
<td>*RPTT</td>
<td>XDSPFLD1</td>
<td>S004</td>
<td>Format data for report total format</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD2</td>
<td>S004</td>
<td>Format Alpha field for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD3</td>
<td>S004</td>
<td>Format Gregorian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD4</td>
<td>S004</td>
<td>Format Julian Date for output</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD5</td>
<td>S004</td>
<td>Format VC0 field from VTX</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD6</td>
<td>S004</td>
<td>Format VC0 field from description file (field details)</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD7</td>
<td>S004</td>
<td>Format Alpha 3 or 28</td>
</tr>
<tr>
<td></td>
<td>XDSPFLD8</td>
<td>S004</td>
<td>Repeat of XDSPFLD1</td>
</tr>
<tr>
<td>*RTA</td>
<td>XTOTARRY</td>
<td>E spec</td>
<td>Load totaling arrays</td>
</tr>
<tr>
<td>*RTS</td>
<td>None</td>
<td>I spec</td>
<td>Report softcoding array</td>
</tr>
<tr>
<td>*RTX</td>
<td>None</td>
<td>I spec</td>
<td>Report softcoding text fields</td>
</tr>
<tr>
<td>*RTXI</td>
<td>XVTIDX</td>
<td>S999</td>
<td>Set maximum VTX index to use</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------</td>
<td>---------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>*SFFLD</td>
<td></td>
<td></td>
<td>Active Data Dictionary data field validation for subfile fields.</td>
</tr>
<tr>
<td>XFIELDDEDT1</td>
<td>S005</td>
<td></td>
<td>Data Dictionary alpha edit</td>
</tr>
<tr>
<td>XFIELDDETT2</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDDETT3</td>
<td>S005</td>
<td></td>
<td>Gregorian edit</td>
</tr>
<tr>
<td>XFIELDDETT4</td>
<td>S005</td>
<td></td>
<td>Julian edit</td>
</tr>
<tr>
<td>XFIELDDETT5</td>
<td>S005</td>
<td></td>
<td>Data Dictionary numeric edit</td>
</tr>
<tr>
<td>XFIELDDETT6</td>
<td>S005</td>
<td></td>
<td>Alpha field size 10</td>
</tr>
<tr>
<td>XFIELDDETT7</td>
<td>S005</td>
<td></td>
<td>User defined code edit</td>
</tr>
<tr>
<td>XFIELDDETT8</td>
<td>S005</td>
<td></td>
<td>No dictionary</td>
</tr>
<tr>
<td>XFIELDDETT9</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Numeric</td>
</tr>
<tr>
<td>XFIELDDETA</td>
<td>S005</td>
<td></td>
<td>Account ID</td>
</tr>
<tr>
<td>XFIELDDETC</td>
<td>S005</td>
<td></td>
<td>Cost center edit</td>
</tr>
<tr>
<td>XFIELDDETE</td>
<td>S005</td>
<td></td>
<td>Numeric field size 7</td>
</tr>
<tr>
<td>XFIELDDETR</td>
<td>S005</td>
<td></td>
<td>Right adjust</td>
</tr>
<tr>
<td>XFIELDDETS</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDDETT</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha Rt Adj</td>
</tr>
<tr>
<td>XFIELDDETU</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Numeric</td>
</tr>
</tbody>
</table>

**SLDxx**

Active Data Dictionary data field validation for subfile data fields. Where xx = specified master file 1 thru 9.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>XFIELDDETT1</td>
<td>S005</td>
<td></td>
<td>Data Dictionary alpha edit</td>
</tr>
<tr>
<td>XFIELDDETT2</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDDETT3</td>
<td>S005</td>
<td></td>
<td>Gregorian edit</td>
</tr>
<tr>
<td>XFIELDDETT4</td>
<td>S005</td>
<td></td>
<td>Julian edit</td>
</tr>
<tr>
<td>XFIELDDETT5</td>
<td>S005</td>
<td></td>
<td>Data Dictionary numeric edit</td>
</tr>
<tr>
<td>XFIELDDETT6</td>
<td>S005</td>
<td></td>
<td>Alpha field size 10</td>
</tr>
<tr>
<td>XFIELDDETT7</td>
<td>S005</td>
<td></td>
<td>User defined code edit</td>
</tr>
<tr>
<td>XFIELDDETT8</td>
<td>S005</td>
<td></td>
<td>No dictionary</td>
</tr>
<tr>
<td>XFIELDDETT9</td>
<td>S005</td>
<td></td>
<td>Validation n = Master - Numeric</td>
</tr>
<tr>
<td>XFIELDDETA</td>
<td>S005</td>
<td></td>
<td>Account ID</td>
</tr>
<tr>
<td>XFIELDDETC</td>
<td>S005</td>
<td></td>
<td>Cost center edit</td>
</tr>
<tr>
<td>XFIELDDETE</td>
<td>S005</td>
<td></td>
<td>Numeric field size 7</td>
</tr>
<tr>
<td>XFIELDDETR</td>
<td>S005</td>
<td></td>
<td>Right adjust</td>
</tr>
<tr>
<td>XFIELDDETS</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha</td>
</tr>
<tr>
<td>XFIELDDETT</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Alpha Rt Adj</td>
</tr>
<tr>
<td>XFIELDDETU</td>
<td>S005</td>
<td></td>
<td>Validation = Master - Numeric</td>
</tr>
</tbody>
</table>
Understand Directives

### Directive Code Detail Logic Module Source Created Functional Directive

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>*S00VL</td>
<td>None</td>
<td>I spec</td>
<td>Cursor Control, F1</td>
</tr>
<tr>
<td>TITLE</td>
<td>None</td>
<td>H spec</td>
<td>Program title</td>
</tr>
<tr>
<td>*VKYxx</td>
<td>None</td>
<td>S999</td>
<td>Load softcoding record key for display files where xx=display file designation 1 - 9.</td>
</tr>
<tr>
<td>*VTS</td>
<td>None</td>
<td>I spec</td>
<td>Display file softcoding array</td>
</tr>
<tr>
<td>*VTX</td>
<td>None</td>
<td>I spec</td>
<td>Update softcoding text field ending positions based upon size definition in display file</td>
</tr>
<tr>
<td>*VTXI</td>
<td>XVTXIDX</td>
<td>S999</td>
<td>Set maximum VTX index used</td>
</tr>
</tbody>
</table>

* 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:

<table>
<thead>
<tr>
<th>Directive</th>
<th>Column Allowed</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>@</td>
<td>x x x x</td>
<td>Four character Data Dictionary name</td>
</tr>
<tr>
<td>#</td>
<td>x</td>
<td>Primary parameter that passes for *ENTRY</td>
</tr>
<tr>
<td>A</td>
<td>x x x</td>
<td>Highest VTX field.</td>
</tr>
<tr>
<td>Directive</td>
<td>Column Allowed</td>
<td>Function</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>----------</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>Unused at this time.</td>
</tr>
<tr>
<td>C</td>
<td>x</td>
<td>Function key exit indicator test</td>
</tr>
<tr>
<td>D</td>
<td>x</td>
<td>Descriptions for fields, files, and copy modules</td>
</tr>
<tr>
<td>E</td>
<td>x</td>
<td>Error message key</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td>Validation file name</td>
</tr>
<tr>
<td>G</td>
<td></td>
<td>User defined calculation logic result field name</td>
</tr>
<tr>
<td>H</td>
<td></td>
<td>Descriptive display file name</td>
</tr>
<tr>
<td>I</td>
<td>x</td>
<td>Display field error condition attribute indicator</td>
</tr>
<tr>
<td>J</td>
<td>x</td>
<td>Data file names</td>
</tr>
<tr>
<td>K</td>
<td>x</td>
<td>Descriptive display file key field name</td>
</tr>
<tr>
<td>L</td>
<td></td>
<td>Data file key list name and optional file/format name</td>
</tr>
<tr>
<td>M</td>
<td></td>
<td>File information data structure name</td>
</tr>
<tr>
<td>N</td>
<td>x</td>
<td>Full data field name (write to)</td>
</tr>
<tr>
<td>O</td>
<td>x</td>
<td>Common subroutine name</td>
</tr>
<tr>
<td>P</td>
<td></td>
<td>Function key/selection exit program to execute</td>
</tr>
<tr>
<td>Q</td>
<td></td>
<td>Field name to receive description value</td>
</tr>
<tr>
<td>R</td>
<td></td>
<td>Field name to receive key value</td>
</tr>
<tr>
<td>S</td>
<td></td>
<td>Selection exit value test</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>Function key/selection exit</td>
</tr>
<tr>
<td>U</td>
<td></td>
<td>File information data structure subfield prefix</td>
</tr>
<tr>
<td>V</td>
<td>x</td>
<td>Source of data (Read From) field name</td>
</tr>
<tr>
<td>W</td>
<td></td>
<td>Data file key list key field name</td>
</tr>
<tr>
<td>X</td>
<td>x</td>
<td>Error message array index</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td>Function key/selection exit parameter field name</td>
</tr>
<tr>
<td>Z</td>
<td></td>
<td>Numeric field size definition (right adj alpha)</td>
</tr>
</tbody>
</table>
Understand Directives

<table>
<thead>
<tr>
<th>Directive</th>
<th>Column Allowed</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>x</td>
<td>Gregorian date Data Structure numeric 6 byte date</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>Gregorian date Data Structure numeric 2 byte month</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>Gregorian date Data Structure numeric 2 byte day</td>
</tr>
<tr>
<td>3</td>
<td>x</td>
<td>Gregorian date Data Structure numeric 2 byte year</td>
</tr>
<tr>
<td>4</td>
<td>x x x x</td>
<td>Parameter 1 from *PROC calculations</td>
</tr>
<tr>
<td>5</td>
<td>x x x x</td>
<td>Parameter 2 from *PROC calculations</td>
</tr>
<tr>
<td>6</td>
<td>x x x x</td>
<td>Parameter 3 from *PROC calculations</td>
</tr>
<tr>
<td>7</td>
<td>x x x x</td>
<td>Parameter 4 from *PROC calculations</td>
</tr>
<tr>
<td>8</td>
<td>x x x x</td>
<td>Parameter 5 from *PROC calculations</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Result</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td></td>
<td></td>
<td>&amp;xxFILE</td>
<td>Master/video/report file name</td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
<td>&amp;xx(FILE)</td>
<td>File name in single quote marks</td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
<td>&amp;xxFORMAT</td>
<td>Master/video format name</td>
</tr>
<tr>
<td>x</td>
<td></td>
<td></td>
<td>&amp;xxFORMAT1</td>
<td>Subfile line 24 format name</td>
</tr>
</tbody>
</table>
Understand Directives

<table>
<thead>
<tr>
<th>Factor 1</th>
<th>Factor 2</th>
<th>Result</th>
<th>Keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>&amp;xxFORMATC</td>
<td></td>
<td></td>
<td>Subfile control record format name</td>
</tr>
<tr>
<td>x</td>
<td>&amp;xxFORMATS</td>
<td></td>
<td></td>
<td>Subfile record format name</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>&amp;xxKEYFLD</td>
<td></td>
<td>Master file primary key field name</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>&amp;xxPGCTL</td>
<td></td>
<td>Number of subfile records in 1 page</td>
</tr>
<tr>
<td>x</td>
<td>%</td>
<td></td>
<td></td>
<td>Factor 1 intentionally left blank</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>x</td>
<td>=</td>
<td>User defined calculation logic result</td>
</tr>
</tbody>
</table>

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 SFSEL.C exists, include code for selection exits.

You can combine conditional directives.

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

<table>
<thead>
<tr>
<th>Position/Factor/Result</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pos 1</td>
<td>+ Include detail logic module if true</td>
</tr>
<tr>
<td></td>
<td>- Include detail logic module if false</td>
</tr>
<tr>
<td>Pos 2-5</td>
<td>FLDN Test existence of data field</td>
</tr>
<tr>
<td></td>
<td>DTAI Test existence of data item</td>
</tr>
<tr>
<td></td>
<td>FILE Test existence of file</td>
</tr>
<tr>
<td></td>
<td>FMT Test existence of file</td>
</tr>
<tr>
<td>Factor 1</td>
<td>Name of field, item, file or format to test. Can also contain *ANY{x for file test which you can use to test for types of files in a program where x might optionally designate number of files.</td>
</tr>
<tr>
<td>Oper (file test only)</td>
<td>DSPF Display file</td>
</tr>
<tr>
<td></td>
<td>PF Physical file only</td>
</tr>
<tr>
<td>Position/Factor/Result</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>LF</td>
<td>Logical file only</td>
</tr>
<tr>
<td>PRTF</td>
<td>Printer file only</td>
</tr>
<tr>
<td>DB</td>
<td>Database file</td>
</tr>
</tbody>
</table>

**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
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.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple Question and Answer</td>
<td>Access the Question Entry screen.</td>
</tr>
<tr>
<td>Add New Q &amp; A Dialogue</td>
<td>Access the Dialogue Descriptions screen.</td>
</tr>
<tr>
<td>Update Existing Q &amp; A Dialogue</td>
<td>Access the Dialogue Lists screen.</td>
</tr>
</tbody>
</table>

### 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 Number

   The 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

   ![Dialogue Descriptions Screen](image)

   The first Question Entry screen displays.

3. Complete the following fields:
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 Library
   The system renames the dialogue.

**To run a dialogue**

1. On Dialogue Lists, enter 5 in the Option field next to the dialogue you want to run.
   The Dialogue Test screen displays.
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.
Appendix A - Program Generator Checklist

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 PSxxx fields instead of VDxxx or SFxxxx fields.
    - May inadvertently change in the program the system launches.
    - You might have to define and load the PSxxx 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.

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.

<table>
<thead>
<tr>
<th>INDICATOR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Causes the Invalid Function Key Pressed message to display</td>
</tr>
<tr>
<td>02</td>
<td>Dictates the color palette to use</td>
</tr>
<tr>
<td>04</td>
<td>Controls subfile keywords SFLDROP and SFLFOLD for fold areas</td>
</tr>
<tr>
<td>20</td>
<td>Controls the clear screen action code</td>
</tr>
<tr>
<td>21</td>
<td>Controls the add action code</td>
</tr>
<tr>
<td>22</td>
<td>Controls the change action code</td>
</tr>
<tr>
<td>23</td>
<td>Controls the delete action code</td>
</tr>
<tr>
<td>24</td>
<td>Controls the inquire action code</td>
</tr>
<tr>
<td>31</td>
<td>The system uses this in conjunction with subfile processing to initiate the INVITE or SFLCLR keyword</td>
</tr>
<tr>
<td>32</td>
<td>The system uses this in conjunction with subfile processing initiating the keyword SFLNXTCHG</td>
</tr>
<tr>
<td>37</td>
<td>The system uses this in conjunction with subfile processing to highlight the last record in the display (used only with inquiry subfiles)</td>
</tr>
<tr>
<td>38</td>
<td>The system uses this in conjunction with subfile processing to control the display keyword SFLDSP</td>
</tr>
<tr>
<td>42-79</td>
<td>The system uses this for error processing to indicate which fields are in error</td>
</tr>
<tr>
<td>40</td>
<td>The system reserves this for errors in the Action Code field</td>
</tr>
<tr>
<td>41</td>
<td>The system reserves this for errors in the key fields</td>
</tr>
<tr>
<td>80-89</td>
<td>Indicates a general reusable one-time indicators</td>
</tr>
<tr>
<td>93</td>
<td>Indicates global error indicator that highlights line 24</td>
</tr>
<tr>
<td>98</td>
<td>Indicates a chain or read failure</td>
</tr>
<tr>
<td>99</td>
<td>Indicates a record is in use</td>
</tr>
<tr>
<td>OF</td>
<td>Indicates overflow for report processing</td>
</tr>
<tr>
<td>LR</td>
<td>Indicates that the last record has been read and the program should end normally</td>
</tr>
</tbody>
</table>
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
MOVE ABANS $ANS,
```

You then define $ANS as follows:

```like
*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 Database 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 an 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 #SSELC 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 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 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 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. 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.
Apppendix 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

<table>
<thead>
<tr>
<th>Seq No.</th>
<th>Mod Date</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>08.02.85</td>
<td>I***************************************************************** 00003</td>
</tr>
<tr>
<td>2</td>
<td>08.02.85</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>08.02.85</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>08.02.85</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>08.02.85</td>
<td></td>
</tr>
</tbody>
</table>
| 6       | 08.02.85 | Portion of this data structure are loaded at the time the program is loaded. Other portions of this data structure are loaded as you perform I/O. 
| 7       | 08.02.85 | 
| 8       | 08.02.85 | 
| 9       | 08.02.85 | 
| 10      | 08.02.85 | PURPOSE 
| 11      | 08.02.85 | 
| 12      | 08.02.85 | 
| 13      | 08.02.85 | 
| 14      | 08.02.85 | 
| 15      | 08.02.85 | 
| 16      | 08.02.85 | 
| 17      | 08.02.85 | 
| 18      | 08.02.85 | 
| 19      | 08.02.85 | 
| 20      | 08.02.85 | 
| 21      | 08.02.85 | 
| 22      | 08.02.85 | 
| 23      | 08.02.85 | 
| 24      | 08.02.85 | 
| 25      | 08.02.85 | 
| 26      | 08.02.85 | 
| 27      | 08.02.85 | 
| 28      | 08.02.85 | 
| 29      | 08.02.85 | 
| 30      | 08.02.85 | 
| 31      | 08.02.85 | 
| 32      | 08.02.85 | 
| 33      | 08.02.85 | 
| 34      | 08.02.85 | 
| 35      | 08.02.85 | 
| 36      | 08.02.85 | 
| 37      | 08.02.85 | 
| 38      | 08.02.85 | 
| 39      | 08.02.85 | 
| 40      | 08.02.85 | 
| 41      | 08.02.85 | 
| 42      | 08.02.85 | 
| 43      | 08.02.85 | 
| 44      | 08.02.85 | 
| 45      | 08.02.85 | 
| 46      | 08.02.85 | 
| 47      | 08.02.85 | 
| 48      | 08.02.85 | 
| 49      | 08.02.85 | 
| 50      | 08.02.85 | 
| 51      | 08.02.85 | 
| 52      | 08.02.85 | 
| 53      | 08.02.85 | 
| 54      | 08.02.85 | 
| 55      | 08.02.85 | 
| 56      | 08.02.85 | 

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-39
Copy Module - Retrieve Soft Coding Data Structure — I00SC

<table>
<thead>
<tr>
<th>Seq No.</th>
<th>JD Edwards World</th>
<th>Print Source Code</th>
<th>Date - 27.01.17</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>I*</td>
<td>1</td>
<td>12.02.88</td>
</tr>
<tr>
<td>2.00</td>
<td>I*</td>
<td>2</td>
<td>12.02.88</td>
</tr>
<tr>
<td>3.00</td>
<td>I*</td>
<td>3</td>
<td>12.02.88</td>
</tr>
<tr>
<td>4.00</td>
<td>I*</td>
<td>4</td>
<td>12.02.88</td>
</tr>
<tr>
<td>5.00</td>
<td>I*</td>
<td>5</td>
<td>12.02.88</td>
</tr>
<tr>
<td>6.00</td>
<td>I*</td>
<td>6</td>
<td>12.02.88</td>
</tr>
<tr>
<td>7.00</td>
<td>I*</td>
<td>7</td>
<td>25.04.88</td>
</tr>
<tr>
<td>8.00</td>
<td>I*</td>
<td>8</td>
<td>25.04.88</td>
</tr>
<tr>
<td>9.00</td>
<td>I*</td>
<td>9</td>
<td>25.04.88</td>
</tr>
<tr>
<td>10.00</td>
<td>I*</td>
<td>10</td>
<td>25.04.88</td>
</tr>
<tr>
<td>11.00</td>
<td>I*</td>
<td>11</td>
<td>12.02.88</td>
</tr>
<tr>
<td>12.00</td>
<td>I*</td>
<td>12</td>
<td>12.02.88</td>
</tr>
<tr>
<td>13.00</td>
<td>I*</td>
<td>13</td>
<td>12.02.88</td>
</tr>
<tr>
<td>14.00</td>
<td>I*</td>
<td>14</td>
<td>12.02.88</td>
</tr>
<tr>
<td>15.00</td>
<td>I*</td>
<td>15</td>
<td>07.01.91</td>
</tr>
<tr>
<td>16.00</td>
<td>I*</td>
<td>16</td>
<td>12.02.88</td>
</tr>
<tr>
<td>17.00</td>
<td>I*</td>
<td>17</td>
<td>17.02.88</td>
</tr>
<tr>
<td>18.00</td>
<td>I*</td>
<td>18</td>
<td>12.02.88</td>
</tr>
<tr>
<td>19.00</td>
<td>I*</td>
<td>19</td>
<td>32 I00SCF</td>
</tr>
<tr>
<td>20.00</td>
<td>I</td>
<td>20</td>
<td>17.02.88</td>
</tr>
<tr>
<td>21.00</td>
<td>I</td>
<td>21</td>
<td>25.04.88</td>
</tr>
<tr>
<td>22.00</td>
<td>I</td>
<td>22</td>
<td>17.02.88</td>
</tr>
<tr>
<td>23.00</td>
<td>I</td>
<td>23</td>
<td>25.04.88</td>
</tr>
<tr>
<td>24.00</td>
<td>I</td>
<td>24</td>
<td>17.02.88</td>
</tr>
<tr>
<td>25.00</td>
<td>I</td>
<td>25</td>
<td>25.04.88</td>
</tr>
<tr>
<td>26.00</td>
<td>I</td>
<td>26</td>
<td>25.04.88</td>
</tr>
<tr>
<td>27.00</td>
<td>I</td>
<td>27</td>
<td>25.04.88</td>
</tr>
<tr>
<td>28.00</td>
<td>I</td>
<td>28</td>
<td>17.02.88</td>
</tr>
<tr>
<td>29.00</td>
<td>I</td>
<td>29</td>
<td>25.04.88</td>
</tr>
<tr>
<td>30.00</td>
<td>I</td>
<td>30</td>
<td>25.04.88</td>
</tr>
<tr>
<td>31.00</td>
<td>I</td>
<td>31</td>
<td>25.04.88</td>
</tr>
<tr>
<td>32.00</td>
<td>I</td>
<td>32</td>
<td>25.04.88</td>
</tr>
<tr>
<td>33.00</td>
<td>I</td>
<td>33</td>
<td>25.04.88</td>
</tr>
<tr>
<td>34.00</td>
<td>I</td>
<td>34</td>
<td>17.02.88</td>
</tr>
<tr>
<td>35.00</td>
<td>I</td>
<td>35</td>
<td>25.04.88</td>
</tr>
<tr>
<td>36.00</td>
<td>I</td>
<td>36</td>
<td>25.04.88</td>
</tr>
<tr>
<td>37.00</td>
<td>I</td>
<td>37</td>
<td>25.04.88</td>
</tr>
<tr>
<td>38.00</td>
<td>I</td>
<td>38</td>
<td>25.04.88</td>
</tr>
<tr>
<td>39.00</td>
<td>I</td>
<td>39</td>
<td>25.04.88</td>
</tr>
<tr>
<td>40.00</td>
<td>I</td>
<td>40</td>
<td>17.02.88</td>
</tr>
<tr>
<td>41.00</td>
<td>I</td>
<td>41</td>
<td>25.04.88</td>
</tr>
<tr>
<td>42.00</td>
<td>I</td>
<td>42</td>
<td>25.04.88</td>
</tr>
<tr>
<td>43.00</td>
<td>I</td>
<td>43</td>
<td>17.02.88</td>
</tr>
<tr>
<td>44.00</td>
<td>I</td>
<td>44</td>
<td>25.04.88</td>
</tr>
<tr>
<td>45.00</td>
<td>I</td>
<td>45</td>
<td>25.04.88</td>
</tr>
<tr>
<td>46.00</td>
<td>I</td>
<td>46</td>
<td>17.02.88</td>
</tr>
<tr>
<td>47.00</td>
<td>I</td>
<td>47</td>
<td>25.04.88</td>
</tr>
<tr>
<td>48.00</td>
<td>I</td>
<td>48</td>
<td>25.04.88</td>
</tr>
<tr>
<td>49.00</td>
<td>I</td>
<td>49</td>
<td>17.02.88</td>
</tr>
<tr>
<td>50.00</td>
<td>I</td>
<td>50</td>
<td>25.04.88</td>
</tr>
<tr>
<td>51.00</td>
<td>I</td>
<td>51</td>
<td>25.04.88</td>
</tr>
<tr>
<td>52.00</td>
<td>I</td>
<td>52</td>
<td>17.02.88</td>
</tr>
<tr>
<td>53.00</td>
<td>I</td>
<td>53</td>
<td>25.04.88</td>
</tr>
<tr>
<td>Seq No.</td>
<td>Mod Date</td>
<td>Description</td>
<td>Date</td>
</tr>
<tr>
<td>--------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------------------------------------</td>
<td>------------</td>
</tr>
<tr>
<td>54.00</td>
<td>25.04.88</td>
<td>I* Function - Exit to Name Search</td>
<td></td>
</tr>
<tr>
<td>55.00</td>
<td>17.02.88</td>
<td>I 12 #FNS</td>
<td>25.04.88</td>
</tr>
<tr>
<td>56.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>57.00</td>
<td>25.04.88</td>
<td>I* Function - Return to Previous Panel/Menu</td>
<td></td>
</tr>
<tr>
<td>58.00</td>
<td>25.04.88</td>
<td>I 13 #FPRV</td>
<td></td>
</tr>
<tr>
<td>59.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>60.00</td>
<td>25.04.88</td>
<td>I* Function - Display Alternate Panel</td>
<td></td>
</tr>
<tr>
<td>61.00</td>
<td>25.04.88</td>
<td>I 14 #FALT</td>
<td></td>
</tr>
<tr>
<td>62.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>63.00</td>
<td>19.09.89</td>
<td>I Function - Exit to Display Valid Function Keys</td>
<td></td>
</tr>
<tr>
<td>64.00</td>
<td>19.09.89</td>
<td>I 15 #FKEYS</td>
<td></td>
</tr>
<tr>
<td>65.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>66.00</td>
<td>25.04.88</td>
<td>I* Function - Return to Primary Menu</td>
<td></td>
</tr>
<tr>
<td>67.00</td>
<td>17.02.88</td>
<td>I 16 #FMM</td>
<td></td>
</tr>
<tr>
<td>68.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>69.00</td>
<td>25.04.88</td>
<td>I* Function - Hard Copy Print</td>
<td></td>
</tr>
<tr>
<td>70.00</td>
<td>21.04.88</td>
<td>I 17 #FPRT</td>
<td></td>
</tr>
<tr>
<td>71.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>72.00</td>
<td>25.04.88</td>
<td>I* Function - Variable by Program (1 thru 15)</td>
<td></td>
</tr>
<tr>
<td>73.00</td>
<td>21.04.88</td>
<td>I 18 #F01</td>
<td></td>
</tr>
<tr>
<td>74.00</td>
<td>21.04.88</td>
<td>I 19 #F02</td>
<td></td>
</tr>
<tr>
<td>75.00</td>
<td>21.04.88</td>
<td>I 20 #F03</td>
<td></td>
</tr>
<tr>
<td>76.00</td>
<td>21.04.88</td>
<td>I 21 #F04</td>
<td></td>
</tr>
<tr>
<td>77.00</td>
<td>21.04.88</td>
<td>I 22 #F05</td>
<td></td>
</tr>
<tr>
<td>78.00</td>
<td>21.04.88</td>
<td>I 23 #F06</td>
<td></td>
</tr>
<tr>
<td>79.00</td>
<td>21.04.88</td>
<td>I 24 #F07</td>
<td></td>
</tr>
<tr>
<td>80.00</td>
<td>21.04.88</td>
<td>I 25 #F08</td>
<td></td>
</tr>
<tr>
<td>81.00</td>
<td>21.04.88</td>
<td>I 26 #F09</td>
<td></td>
</tr>
<tr>
<td>82.00</td>
<td>21.04.88</td>
<td>I 27 #F10</td>
<td></td>
</tr>
<tr>
<td>83.00</td>
<td>21.04.88</td>
<td>I 28 #F11</td>
<td></td>
</tr>
<tr>
<td>84.00</td>
<td>21.04.88</td>
<td>I 29 #F12</td>
<td></td>
</tr>
<tr>
<td>85.00</td>
<td>21.04.88</td>
<td>I 30 #F13</td>
<td></td>
</tr>
<tr>
<td>86.00</td>
<td>21.04.88</td>
<td>I 31 #F14</td>
<td></td>
</tr>
<tr>
<td>87.00</td>
<td>21.04.88</td>
<td>I 32 #F15</td>
<td></td>
</tr>
<tr>
<td>88.00</td>
<td>17.02.88</td>
<td>I* Selections 1 thru 24.</td>
<td></td>
</tr>
<tr>
<td>89.00</td>
<td>17.02.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>90.00</td>
<td>17.02.88</td>
<td>I 33 80 I00SCS</td>
<td></td>
</tr>
<tr>
<td>91.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>92.00</td>
<td>25.04.88</td>
<td>I* Selection - Select/Work With</td>
<td></td>
</tr>
<tr>
<td>93.00</td>
<td>25.04.88</td>
<td>I 33 340#SELC</td>
<td></td>
</tr>
<tr>
<td>94.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>95.00</td>
<td>25.04.88</td>
<td>I* Selection - Change/Revise</td>
<td></td>
</tr>
<tr>
<td>96.00</td>
<td>25.04.88</td>
<td>I 35 360#CHANGE</td>
<td></td>
</tr>
<tr>
<td>97.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>98.00</td>
<td>25.04.88</td>
<td>I* Selection - Copy/Hold</td>
<td></td>
</tr>
<tr>
<td>99.00</td>
<td>25.04.88</td>
<td>I 37 380#COPY</td>
<td></td>
</tr>
<tr>
<td>100.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>101.00</td>
<td>25.04.88</td>
<td>I* Selection - Delete/Cancel</td>
<td></td>
</tr>
<tr>
<td>102.00</td>
<td>25.04.88</td>
<td>I 39 400#DELETE</td>
<td></td>
</tr>
<tr>
<td>103.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>104.00</td>
<td>25.04.88</td>
<td>I* Selection - Display/View</td>
<td></td>
</tr>
<tr>
<td>105.00</td>
<td>25.04.88</td>
<td>I 41 420#DISPLAY</td>
<td></td>
</tr>
<tr>
<td>106.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>107.00</td>
<td>25.04.88</td>
<td>I* Selection - Print/Release</td>
<td></td>
</tr>
<tr>
<td>108.00</td>
<td>25.04.88</td>
<td>I 43 440#PRINT</td>
<td></td>
</tr>
<tr>
<td>109.00</td>
<td>25.04.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>110.00</td>
<td>25.04.88</td>
<td>I* Selection - Rename</td>
<td></td>
</tr>
<tr>
<td>111.00</td>
<td>25.04.88</td>
<td>I 45 460#RENUM</td>
<td></td>
</tr>
<tr>
<td>112.00</td>
<td>07.06.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>113.00</td>
<td>07.06.88</td>
<td>I* Selection - Display Attributes</td>
<td></td>
</tr>
<tr>
<td>114.00</td>
<td>07.06.88</td>
<td>I 47 480#DATR</td>
<td></td>
</tr>
<tr>
<td>115.00</td>
<td>07.06.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>116.00</td>
<td>07.06.88</td>
<td>I* Selection - Variable by Program (1 thru 16)</td>
<td></td>
</tr>
<tr>
<td>117.00</td>
<td>07.06.88</td>
<td>I 49 500#E01</td>
<td></td>
</tr>
<tr>
<td>118.00</td>
<td>07.06.88</td>
<td>I 51 520#E02</td>
<td></td>
</tr>
<tr>
<td>119.00</td>
<td>07.06.88</td>
<td>I 53 540#E03</td>
<td></td>
</tr>
<tr>
<td>120.00</td>
<td>07.06.88</td>
<td>I 55 560#E04</td>
<td></td>
</tr>
<tr>
<td>121.00</td>
<td>07.06.88</td>
<td>I 57 580#E05</td>
<td></td>
</tr>
<tr>
<td>122.00</td>
<td>07.06.88</td>
<td>I 59 600#E06</td>
<td></td>
</tr>
<tr>
<td>123.00</td>
<td>07.06.88</td>
<td>I 61 620#E07</td>
<td></td>
</tr>
<tr>
<td>124.00</td>
<td>07.06.88</td>
<td>I 63 640#E08</td>
<td></td>
</tr>
<tr>
<td>125.00</td>
<td>07.06.88</td>
<td>I 65 660#E09</td>
<td></td>
</tr>
<tr>
<td>126.00</td>
<td>07.06.88</td>
<td>I 67 680#E10</td>
<td></td>
</tr>
<tr>
<td>127.00</td>
<td>07.06.88</td>
<td>I 69 700#E11</td>
<td></td>
</tr>
<tr>
<td>128.00</td>
<td>07.06.88</td>
<td>I 71 720#E12</td>
<td></td>
</tr>
<tr>
<td>129.00</td>
<td>07.06.88</td>
<td>I 73 740#E13</td>
<td></td>
</tr>
<tr>
<td>130.00</td>
<td>07.06.88</td>
<td>I 75 760#E14</td>
<td></td>
</tr>
<tr>
<td>131.00</td>
<td>07.06.88</td>
<td>I 77 780#E15</td>
<td></td>
</tr>
<tr>
<td>132.00</td>
<td>07.06.88</td>
<td>I 79 800#E16</td>
<td></td>
</tr>
<tr>
<td>133.00</td>
<td>07.06.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>134.00</td>
<td>22.02.88</td>
<td>I* Global JD Edwards World Variables</td>
<td></td>
</tr>
<tr>
<td>135.00</td>
<td>07.01.91</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>136.00</td>
<td>22.02.88</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>137.00</td>
<td>07.01.91</td>
<td>I*</td>
<td></td>
</tr>
<tr>
<td>138.00</td>
<td>25.04.88</td>
<td>I* Future use space, room to grow</td>
<td></td>
</tr>
</tbody>
</table>
### Appendix D - Source Listings

<table>
<thead>
<tr>
<th>Seq No.</th>
<th>JD Edwards World I00SCC .ZDFERC61 Print Source Code Date - 27.01.17 Mod Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>166.00</td>
<td>I* Record format being processed (External file) 22.02.88</td>
</tr>
<tr>
<td>170.00</td>
<td>I* Record ID (Left justified for internal file) 22.02.88</td>
</tr>
<tr>
<td>171.00</td>
<td>I 38 45 $FRMT 22.02.88</td>
</tr>
<tr>
<td>172.00</td>
<td>I* Machine OR CPF message number 22.02.88</td>
</tr>
<tr>
<td>173.00</td>
<td>I 46 52 $EXNO 22.02.88</td>
</tr>
<tr>
<td>174.00</td>
<td>I* Machine instruction/Object definition template number 22.02.88</td>
</tr>
<tr>
<td>175.00</td>
<td>I 53 56 $MI 22.02.88</td>
</tr>
<tr>
<td>176.00</td>
<td>I* Unused 22.02.88</td>
</tr>
<tr>
<td>177.00</td>
<td>I 88 89 $MBR 22.02.88</td>
</tr>
<tr>
<td>178.00</td>
<td>I* Open data path type (DS-Device DB-Data Base SP-Spool) 22.02.88</td>
</tr>
<tr>
<td>179.00</td>
<td>I 81 82 $OPCD 22.02.88</td>
</tr>
<tr>
<td>180.00</td>
<td>I* Name of file actually opened 22.02.88</td>
</tr>
<tr>
<td>181.00</td>
<td>I 83 92 $FILE 22.02.88</td>
</tr>
<tr>
<td>182.00</td>
<td>I* Name of library containing file (Blank if spool file) 22.02.88</td>
</tr>
<tr>
<td>183.00</td>
<td>I 93 102 $LIBR 22.02.88</td>
</tr>
<tr>
<td>184.00</td>
<td>I* Name of spooled file (set only on spool files) 22.02.88</td>
</tr>
<tr>
<td>185.00</td>
<td>I 103 112 $SPNL 22.02.88</td>
</tr>
<tr>
<td>186.00</td>
<td>I* Name of library where spooled file is located 22.02.88</td>
</tr>
<tr>
<td>187.00</td>
<td>I 113 122 $SPLB 22.02.88</td>
</tr>
<tr>
<td>188.00</td>
<td>I* Spooled file number (set only on spool files) 22.02.88</td>
</tr>
<tr>
<td>189.00</td>
<td>I B 123 1240@BPR 22.02.88</td>
</tr>
<tr>
<td>190.00</td>
<td>I* Primary record length (bytes transferred at a time) 22.02.88</td>
</tr>
<tr>
<td>191.00</td>
<td>I B 125 1260@BRL 22.02.88</td>
</tr>
<tr>
<td>192.00</td>
<td>I* Secondary record length (bytes transferred at a time) 22.02.88</td>
</tr>
<tr>
<td>193.00</td>
<td>I B 127 1280@BSRL 22.02.88</td>
</tr>
<tr>
<td>194.00</td>
<td>I* Member Name: 22.02.88</td>
</tr>
<tr>
<td>195.00</td>
<td>I* If ODP type is DB, this entry is the 22.02.88</td>
</tr>
<tr>
<td>200.00</td>
<td>I* member name in file named in position 22.02.88</td>
</tr>
<tr>
<td>201.00</td>
<td>I* 83 through 92. 22.02.88</td>
</tr>
<tr>
<td>202.00</td>
<td>I* If ODP type is SP, this entry is the 22.02.88</td>
</tr>
<tr>
<td>203.00</td>
<td>I* member name in the file named in 22.02.88</td>
</tr>
<tr>
<td>204.00</td>
<td>I* positions 103 through 112. 22.02.88</td>
</tr>
<tr>
<td>205.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>206.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>207.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>208.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>209.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>210.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>211.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>212.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>213.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>214.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>215.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>216.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>217.00</td>
<td>B 139 1420@BRL 22.02.88</td>
</tr>
<tr>
<td>218.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>219.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>220.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>221.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>222.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>223.00</td>
<td>I* 22.02.88</td>
</tr>
<tr>
<td>224.00</td>
<td>I* 22.02.88</td>
</tr>
</tbody>
</table>

---

98330 JD Edwards World I00SCC .ZDFERC61 Print Source Code Date - 27.01.17 Mod Date
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>225</td>
<td>3 = Card</td>
</tr>
<tr>
<td>226</td>
<td>4 = Diskette</td>
</tr>
<tr>
<td>227</td>
<td>5 = Tape</td>
</tr>
<tr>
<td>228</td>
<td>B 147 1480@DVCL</td>
</tr>
<tr>
<td>229</td>
<td>Diskette location (value from 1 to 23 = slot location)</td>
</tr>
<tr>
<td>230</td>
<td>Number of rows on display screen or lines on a page</td>
</tr>
<tr>
<td>231</td>
<td>B 152 1530@VDRW</td>
</tr>
<tr>
<td>232</td>
<td>Number of columns on display screen or printed line</td>
</tr>
<tr>
<td>233</td>
<td>Number of records in file at time of open</td>
</tr>
<tr>
<td>234</td>
<td>B 156 1590@RCONT</td>
</tr>
<tr>
<td>235</td>
<td>Access type (only supplied if ODP type is DB)</td>
</tr>
<tr>
<td>236</td>
<td>KU = Keyed, Unique</td>
</tr>
<tr>
<td>237</td>
<td>RF = Keyed, FIFO W/Duplicate keys</td>
</tr>
<tr>
<td>238</td>
<td>KI = Keyed, LIPO W/Duplicate keys</td>
</tr>
<tr>
<td>239</td>
<td>AR = Arrival sequence</td>
</tr>
<tr>
<td>240</td>
<td>B 160 161 @RACTY</td>
</tr>
<tr>
<td>241</td>
<td>Duplicate key indication (D=Allowed U=Not allowed)</td>
</tr>
<tr>
<td>242</td>
<td>Source file indication (Y=Source file)</td>
</tr>
<tr>
<td>243</td>
<td>B 184 1850@RGLF</td>
</tr>
<tr>
<td>244</td>
<td>Number of records to be transferred on file open</td>
</tr>
<tr>
<td>245</td>
<td>Overflow line number (printer files only)</td>
</tr>
<tr>
<td>246</td>
<td>Name of record format just processed</td>
</tr>
<tr>
<td>247</td>
<td>Device Class</td>
</tr>
<tr>
<td>248</td>
<td>Get operation count</td>
</tr>
<tr>
<td>249</td>
<td>Non-I/O operation count (update of subfile records)</td>
</tr>
<tr>
<td>250</td>
<td>B 243 2460@PUTC</td>
</tr>
<tr>
<td>251</td>
<td>Get operation count</td>
</tr>
<tr>
<td>252</td>
<td>B 247 2500@GETC</td>
</tr>
<tr>
<td>253</td>
<td>UNUSED</td>
</tr>
<tr>
<td>254</td>
<td>Offset to volume label fields of open feedback</td>
</tr>
<tr>
<td>255</td>
<td>Offset to device dependent feedback information</td>
</tr>
<tr>
<td>256</td>
<td>(See Appendix D of the CPF Programmer’s Guide for specific devices)</td>
</tr>
<tr>
<td>257</td>
<td>Non-I/O operation count (update of subfile records)</td>
</tr>
<tr>
<td>258</td>
<td>B 251 2540@PGC</td>
</tr>
<tr>
<td>259</td>
<td>X’01’ = Get</td>
</tr>
<tr>
<td>260</td>
<td>X’02’ = Get key</td>
</tr>
<tr>
<td>261</td>
<td>X’03’ = Put</td>
</tr>
<tr>
<td>262</td>
<td>X’04’ = PutGet</td>
</tr>
<tr>
<td>263</td>
<td>X’05’ = Update</td>
</tr>
<tr>
<td>264</td>
<td>X’06’ = Release</td>
</tr>
<tr>
<td>265</td>
<td>X’07’ = Force End of Data</td>
</tr>
<tr>
<td>266</td>
<td>X’08’ = Release</td>
</tr>
<tr>
<td>267</td>
<td>X’09’ = Force End of Data</td>
</tr>
<tr>
<td>268</td>
<td>X’0A’ = Release</td>
</tr>
<tr>
<td>269</td>
<td>X’0B’ = Release</td>
</tr>
<tr>
<td>270</td>
<td>X’0C’ = Release</td>
</tr>
<tr>
<td>271</td>
<td>X’0D’ = Release</td>
</tr>
<tr>
<td>272</td>
<td>X’0E’ = Release</td>
</tr>
<tr>
<td>273</td>
<td>X’0F’ = Release</td>
</tr>
</tbody>
</table>

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-43
<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>314</td>
<td>I* X'03' = Card</td>
<td>22.02.88</td>
</tr>
<tr>
<td>315</td>
<td>I* X'04' = Diskette</td>
<td>22.02.88</td>
</tr>
<tr>
<td>316</td>
<td>I* X'05' = Tape</td>
<td>22.02.88</td>
</tr>
<tr>
<td>317</td>
<td>I* Position 272 (if position 271 contains X'00')</td>
<td>22.02.88</td>
</tr>
<tr>
<td>318</td>
<td>I* X'01' = Keyed file</td>
<td>22.02.88</td>
</tr>
<tr>
<td>319</td>
<td>I* Position 272 (if position 271 not X'00')</td>
<td>22.02.88</td>
</tr>
<tr>
<td>320</td>
<td>I* X'00' = Nonkeyed file</td>
<td>22.02.88</td>
</tr>
<tr>
<td>321</td>
<td>I* X'01' = Keyed file</td>
<td>22.02.88</td>
</tr>
<tr>
<td>322</td>
<td>I* Position 272 (if position 271 not X'00')</td>
<td>22.02.88</td>
</tr>
<tr>
<td>323</td>
<td>I* X'00' = 5250 Display station, 960 characters</td>
<td>22.02.88</td>
</tr>
<tr>
<td>324</td>
<td>I* X'01' = System console, 1024 characters</td>
<td>22.02.88</td>
</tr>
<tr>
<td>325</td>
<td>I* X'02' = 5256 Printer</td>
<td>22.02.88</td>
</tr>
<tr>
<td>326</td>
<td>I* X'03' = 5211/3262 Printer</td>
<td>22.02.88</td>
</tr>
<tr>
<td>327</td>
<td>I* X'04' = MFCU</td>
<td>22.02.88</td>
</tr>
<tr>
<td>328</td>
<td>I* X'05' = 3411/3410 Tape</td>
<td>22.02.88</td>
</tr>
<tr>
<td>329</td>
<td>I* X'06' = 72M Diskette</td>
<td>22.02.88</td>
</tr>
<tr>
<td>330</td>
<td>I* X'07' = 5250 Display station, 1920 characters</td>
<td>22.02.88</td>
</tr>
<tr>
<td>331</td>
<td>I* X'08' = Spoolied</td>
<td>22.02.88</td>
</tr>
<tr>
<td>332</td>
<td>I 271 272 @@DCLS</td>
<td>22.02.88</td>
</tr>
<tr>
<td>333</td>
<td>I* 22.02.88</td>
<td></td>
</tr>
<tr>
<td>334</td>
<td>I* Device name (Last completed operation)</td>
<td>22.02.88</td>
</tr>
<tr>
<td>335</td>
<td>I 287 366 @@RDTA</td>
<td>22.02.88</td>
</tr>
<tr>
<td>336</td>
<td>I* 22.02.88</td>
<td></td>
</tr>
<tr>
<td>337</td>
<td>I* Routing data information</td>
<td>22.02.88</td>
</tr>
<tr>
<td>338</td>
<td>I* 287 366 @@RDTA</td>
<td>22.02.88</td>
</tr>
<tr>
<td>339</td>
<td>I* Current line number within a printer page</td>
<td>22.02.88</td>
</tr>
<tr>
<td>340</td>
<td>I B 367 3680@@CLNO</td>
<td>22.02.88</td>
</tr>
<tr>
<td>341</td>
<td>I* 22.02.88</td>
<td></td>
</tr>
<tr>
<td>342</td>
<td>I* Length of last I/O record processed</td>
<td>22.02.88</td>
</tr>
<tr>
<td>343</td>
<td>I* 22.02.88</td>
<td></td>
</tr>
<tr>
<td>344</td>
<td>I* AID character indication:</td>
<td>22.02.88</td>
</tr>
<tr>
<td>345</td>
<td>I* X'F1' = Enter/Rec Adv</td>
<td>22.02.88</td>
</tr>
<tr>
<td>346</td>
<td>I* X'F5' = Roll up</td>
<td>22.02.88</td>
</tr>
<tr>
<td>347</td>
<td>I* X'F4' = Roll down</td>
<td>22.02.88</td>
</tr>
<tr>
<td>348</td>
<td>I* X'F6' = Print</td>
<td>22.02.88</td>
</tr>
<tr>
<td>349</td>
<td>I* X'F8' = Home</td>
<td>22.02.88</td>
</tr>
<tr>
<td>350</td>
<td>I* X'B0' = Clear</td>
<td>22.02.88</td>
</tr>
<tr>
<td>351</td>
<td>I* X'B3' = Help</td>
<td>22.02.88</td>
</tr>
<tr>
<td>352</td>
<td>I* X'B4' = Auto Enter</td>
<td>22.02.88</td>
</tr>
<tr>
<td>353</td>
<td>I* X'31' = Command Key 01</td>
<td>09.08.91</td>
</tr>
<tr>
<td>354</td>
<td>I* X'32' = Command Key 02</td>
<td>22.02.88</td>
</tr>
<tr>
<td>355</td>
<td>I* X'33' = Command Key 03</td>
<td>22.02.88</td>
</tr>
<tr>
<td>356</td>
<td>I* X'34' = Command Key 04</td>
<td>22.02.88</td>
</tr>
<tr>
<td>357</td>
<td>I* X'35' = Command Key 05</td>
<td>22.02.88</td>
</tr>
<tr>
<td>358</td>
<td>I* X'36' = Command Key 06</td>
<td>22.02.88</td>
</tr>
<tr>
<td>359</td>
<td>I* X'37' = Command Key 07</td>
<td>22.02.88</td>
</tr>
<tr>
<td>360</td>
<td>I* X'38' = Command Key 08</td>
<td>22.02.88</td>
</tr>
<tr>
<td>361</td>
<td>I* X'39' = Command Key 09</td>
<td>22.02.88</td>
</tr>
<tr>
<td>362</td>
<td>I* X'3A' = Command Key 10</td>
<td>22.02.88</td>
</tr>
<tr>
<td>363</td>
<td>I* X'3B' = Command Key 11</td>
<td>22.02.88</td>
</tr>
<tr>
<td>364</td>
<td>I* X'3C' = Command Key 12</td>
<td>22.02.88</td>
</tr>
<tr>
<td>365</td>
<td>I* X'3D' = Command Key 13</td>
<td>22.02.88</td>
</tr>
<tr>
<td>366</td>
<td>I* X'82' = Command Key 14</td>
<td>22.02.88</td>
</tr>
<tr>
<td>367</td>
<td>I* X'83' = Command Key 15</td>
<td>22.02.88</td>
</tr>
<tr>
<td>368</td>
<td>I* X'84' = Command Key 16</td>
<td>22.02.88</td>
</tr>
<tr>
<td>369</td>
<td>I* X'85' = Command Key 17</td>
<td>22.02.88</td>
</tr>
<tr>
<td>370</td>
<td>I* X'86' = Command Key 18</td>
<td>22.02.88</td>
</tr>
<tr>
<td>371</td>
<td>I* X'87' = Command Key 19</td>
<td>22.02.88</td>
</tr>
<tr>
<td>372</td>
<td>I* X'88' = Command Key 20</td>
<td>22.02.88</td>
</tr>
<tr>
<td>373</td>
<td>I* X'89' = Command Key 21</td>
<td>22.02.88</td>
</tr>
<tr>
<td>374</td>
<td>I* X'8A' = Command Key 22</td>
<td>22.02.88</td>
</tr>
<tr>
<td>375</td>
<td>I* X'8B' = Command Key 23</td>
<td>22.02.88</td>
</tr>
<tr>
<td>376</td>
<td>I* X'8C' = Command Key 24</td>
<td>22.02.88</td>
</tr>
<tr>
<td>377</td>
<td>I 369 369 @@AID</td>
<td>22.02.88</td>
</tr>
<tr>
<td>378</td>
<td>I 369 369 @@AID</td>
<td>22.02.88</td>
</tr>
<tr>
<td>379</td>
<td>I* Cursor line in hex (display files only)</td>
<td>22.02.88</td>
</tr>
<tr>
<td>380</td>
<td>I 370 370 @@CURL</td>
<td>22.02.88</td>
</tr>
<tr>
<td>381</td>
<td>I* Cursor position in hex (display files only)</td>
<td>22.02.88</td>
</tr>
<tr>
<td>382</td>
<td>I 371 371 @@CURP</td>
<td>22.02.88</td>
</tr>
<tr>
<td>383</td>
<td>I* Note: By simply defining a 2 byte binary field</td>
<td>22.02.88</td>
</tr>
<tr>
<td>384</td>
<td>I* and moving the cursor line/position field</td>
<td>22.02.88</td>
</tr>
<tr>
<td>385</td>
<td>I* into it right justified you will have the</td>
<td>22.02.88</td>
</tr>
<tr>
<td>386</td>
<td>I* numerical value of the line/position field.</td>
<td>22.02.88</td>
</tr>
<tr>
<td>387</td>
<td>I* Remember the binary field must be set to</td>
<td>22.02.88</td>
</tr>
<tr>
<td>388</td>
<td>I* zero prior to the move.</td>
<td>22.02.88</td>
</tr>
<tr>
<td>389</td>
<td>I* Number of records transmitted</td>
<td>22.02.88</td>
</tr>
<tr>
<td>390</td>
<td>I 98330 JD Edwards World</td>
<td>22.02.88</td>
</tr>
<tr>
<td>391</td>
<td>I 98330 JD Edwards World</td>
<td>22.02.88</td>
</tr>
<tr>
<td>392</td>
<td>I 98330 JD Edwards World</td>
<td>22.02.88</td>
</tr>
<tr>
<td>Seq No.</td>
<td>Field Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>SHIN01</td>
<td>Returned field name.</td>
</tr>
<tr>
<td>1</td>
<td>SHIN02</td>
<td>Returned value.</td>
</tr>
<tr>
<td>1</td>
<td>SHIN03</td>
<td>Returned description.</td>
</tr>
<tr>
<td>1</td>
<td>SHIN04</td>
<td>Returned location: Row.</td>
</tr>
<tr>
<td>1</td>
<td>SHIN05</td>
<td>Returned location: Column.</td>
</tr>
<tr>
<td>1</td>
<td>SHIN06</td>
<td>Dictionary Field Name (non-blank=override)</td>
</tr>
<tr>
<td>1</td>
<td>SHIN07</td>
<td>Returned Display File Format</td>
</tr>
<tr>
<td>1</td>
<td>SHIN08</td>
<td>RPG Indicator Array</td>
</tr>
<tr>
<td>1</td>
<td>SHIN09</td>
<td>Override Reporting System (Jargon)</td>
</tr>
<tr>
<td>1</td>
<td>SHIN10</td>
<td>Hidden Fields for Subfile Attribute Indicators</td>
</tr>
<tr>
<td>1</td>
<td>SHIN11</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN12</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN13</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN14</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN15</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN16</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN17</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN18</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN19</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN20</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN21</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN22</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN23</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN24</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN25</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN26</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN27</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN28</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN29</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN30</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN31</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN32</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN33</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN34</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN35</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN36</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN37</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN38</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN39</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN40</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN41</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN42</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN43</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN44</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN45</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN46</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN47</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>SHIN48</td>
<td></td>
</tr>
<tr>
<td>Seq No.</td>
<td>Mod Date</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>505.00</td>
<td>65 65 SHIN65</td>
<td></td>
</tr>
<tr>
<td>506.00</td>
<td>66 66 SHIN66</td>
<td></td>
</tr>
<tr>
<td>507.00</td>
<td>67 67 SHIN67</td>
<td></td>
</tr>
<tr>
<td>508.00</td>
<td>68 68 SHIN68</td>
<td></td>
</tr>
<tr>
<td>509.00</td>
<td>69 69 SHIN69</td>
<td></td>
</tr>
<tr>
<td>510.00</td>
<td>70 70 SHIN70</td>
<td></td>
</tr>
<tr>
<td>511.00</td>
<td>71 71 SHIN71</td>
<td></td>
</tr>
<tr>
<td>512.00</td>
<td>72 72 SHIN72</td>
<td></td>
</tr>
<tr>
<td>513.00</td>
<td>73 73 SHIN73</td>
<td></td>
</tr>
<tr>
<td>514.00</td>
<td>74 74 SHIN74</td>
<td></td>
</tr>
<tr>
<td>515.00</td>
<td>75 75 SHIN75</td>
<td></td>
</tr>
<tr>
<td>516.00</td>
<td>76 76 SHIN76</td>
<td></td>
</tr>
<tr>
<td>517.00</td>
<td>77 77 SHIN77</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

Appendix D - Source Listings

<table>
<thead>
<tr>
<th>Seq No.</th>
<th>Mod Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>505.00</td>
<td>65 65 SHIN65</td>
</tr>
<tr>
<td>506.00</td>
<td>66 66 SHIN66</td>
</tr>
<tr>
<td>507.00</td>
<td>67 67 SHIN67</td>
</tr>
<tr>
<td>508.00</td>
<td>68 68 SHIN68</td>
</tr>
<tr>
<td>509.00</td>
<td>69 69 SHIN69</td>
</tr>
<tr>
<td>510.00</td>
<td>70 70 SHIN70</td>
</tr>
<tr>
<td>511.00</td>
<td>71 71 SHIN71</td>
</tr>
<tr>
<td>512.00</td>
<td>72 72 SHIN72</td>
</tr>
<tr>
<td>513.00</td>
<td>73 73 SHIN73</td>
</tr>
<tr>
<td>514.00</td>
<td>74 74 SHIN74</td>
</tr>
<tr>
<td>515.00</td>
<td>75 75 SHIN75</td>
</tr>
<tr>
<td>516.00</td>
<td>76 76 SHIN76</td>
</tr>
<tr>
<td>517.00</td>
<td>77 77 SHIN77</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

98330 JD Edwards World
I00SC .JEHESC61 Print Source Code  Date - 27.01.17
# Item Master Information — P928011

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>H/TITLE P928011-Item Master Information</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>H*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>H*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>H*</td>
<td>Copyright (c) 1993</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>H*</td>
<td>J. D. Edwards &amp; Company</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>H*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>H*</td>
<td>This unpublished material is proprietary to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>H*</td>
<td>J. D. Edwards &amp; Company. All rights reserved.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>H*</td>
<td>The methods and techniques described herein are</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>H*</td>
<td>considered trade secrets and/or confidential.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.00</td>
<td>H*</td>
<td>Reproduction or distribution, in whole or in part,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>H*</td>
<td>is forbidden except by express written permission</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>H*</td>
<td>of J. D. Edwards &amp; Company.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Program Revision Log**

<table>
<thead>
<tr>
<th>Date</th>
<th>Programmer</th>
<th>Nature of Revision</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12/07/93</td>
<td>Querida</td>
<td>SAR # 241801 (AS/400) A/G</td>
<td></td>
</tr>
</tbody>
</table>

**Copy Member for Composite Common Subroutine — C0001**

```
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>31.00</td>
<td>F*</td>
<td>PP9001</td>
<td>IP</td>
<td>E</td>
<td>K</td>
</tr>
<tr>
<td>32.00</td>
<td>F*</td>
<td>PP92001</td>
<td>UP</td>
<td>E</td>
<td>K</td>
</tr>
<tr>
<td>33.00</td>
<td>F*</td>
<td>PP92801L</td>
<td>CT</td>
<td>E</td>
<td>WORKSTN</td>
</tr>
</tbody>
</table>

**Program Tables and Arrays**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>44.00</td>
<td>E</td>
<td>EME</td>
<td>64</td>
<td>4</td>
<td>Error Msg</td>
</tr>
<tr>
<td>45.00</td>
<td>E</td>
<td>aEME</td>
<td>64</td>
<td>4</td>
<td>Error Msg</td>
</tr>
<tr>
<td>46.00</td>
<td>E</td>
<td>aME</td>
<td>64</td>
<td>4</td>
<td>Error Msg</td>
</tr>
<tr>
<td>47.00</td>
<td>E</td>
<td>aDY</td>
<td>64</td>
<td>4</td>
<td>DCit Whk</td>
</tr>
<tr>
<td>48.00</td>
<td>E</td>
<td>aC</td>
<td>256</td>
<td>4</td>
<td>Literal Work</td>
</tr>
</tbody>
</table>

**Copy Member for Composite Common Subroutine — C0001**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>52.00</td>
<td>F*</td>
<td>COPY JDECPY</td>
<td>E9001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Copy Member for Composite Common Subroutine — C0012**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>57.00</td>
<td>F*</td>
<td>COPY JDECPY</td>
<td>E9012</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Copy Member for Composite Common Subroutine — C997**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>61.00</td>
<td>F*</td>
<td>COPY JDECPY</td>
<td>E997</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Program Input Specifications and Data Structures**

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>67.00</td>
<td>F*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68.00</td>
<td>F*</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix D - Source Listings

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>69</td>
<td>JD Edwards World 9-48</td>
<td>Data structure to Load Video screen Text</td>
</tr>
<tr>
<td>70</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>JD Edwards World 9-48</td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>INDSYS DB LD 1000</td>
<td>Each VXX field is 40 long but may not use all 40. Pulls text from Vocabulary Overrides.</td>
</tr>
<tr>
<td>73</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

- **Data structure for commonly used indexes**: I/COPY JREDPY, JREDPROD
- **Data structure used with file servers**: I/COPY JREDPY, JREDPROD
- **Program status data structure**: I/COPY JREDPY, JREDPROD
- **Data structure for vocabulary overrides and function keys**: I/COPY JREDPY, JREDPROD
- **Data structure for file server N0005**: I/COPY JREDPY, JREDPROD
- **Data structure for file server N0006**: I/COPY JREDPY, JREDPROD
- **Data structure for file server N0007**: I/COPY JREDPY, JREDPROD
- **Data structure for file server N0008**: I/COPY JREDPY, JREDPROD
- **Data structure for file server N0009**: I/COPY JREDPY, JREDPROD

---

- **MAINLINE PROGRAM**: C=='
- **Process housekeeping**: C=='
- **End Screen, exit program**
- **If automatic inquiry set, process inquiry**: C=='
- **If information is passed to this program, it will automatically inquire on the record**
- **Write video screen**: C=='
Appendix D - Source Listings

160.00 C
161.00 C
162.00 C
163.00 C
164.00 C
165.00 C
166.00 C
167.00 C
168.00 C
169.00 C
170.00 C
171.00 C
172.00 C
173.00 C
174.00 C
175.00 C
176.00 C
177.00 C
178.00 C
179.00 C
180.00 C
181.00 C
182.00 C
183.00 C
184.00 C
185.00 C
186.00 C
187.00 C
188.00 C
189.00 C
190.00 C
191.00 C
192.00 C
193.00 C
194.00 C
195.00 C
196.00 C
197.00 C
198.00 C
199.00 C
200.00 C
201.00 C
202.00 C
203.00 C
204.00 C
205.00 C
206.00 C
207.00 C
208.00 C
209.00 C
210.00 C
211.00 C
212.00 C
213.00 C
214.00 C
215.00 C
216.00 C
217.00 C
218.00 C
219.00 C
220.00 C
221.00 C
222.00 C
223.00 C
224.00 C
225.00 C
226.00 C
227.00 C

Load data file dictionary parameters (one cycle only).  
One time only. Pulls in Data Dictionary editing information functions

Begin video screen read processing.

Used for cursor sensitive help. Tells where the cursor is.

If video read timed out, end program.

If valid function key pressed, process and return.

All function keys are assigned indicator 15 so if 15 is on, a function key has been pressed

Edit the action code.

Checks action code security.

If clear screen requested, process and return.

If add or change, validate all video input.

If no errors and not inquiry, update file.

Sets the file pointer and calls S004 to load the video/report fields

If an error has occurred, validates and edits data

Updates file

Return for next input.
Appendix D - Source Listings

*IND: JFDX 'L'  NOVELSYS34  VOL24
                ELST
                NOVELSYS34  VOL24
                END
                END MAINLINE PROGRAM
                                ------------------------------------------
                **Copy Common Subroutine - Edit Action Code
                C/COPY JDRCPY, C00601
                **SUBROUTINE SOCKET - PROCESS FUNCTION KEYS
                1. Determine function key pressed.
                2. Process function key request.
                Contains what function key was
                pressed by the user
                If key requested, exit subroutine.
                If display keys pressed, exit to help facility and return.
                If Cursor Sensitive Help Pressed, exit to CB Help.
                External programs start with
                an X. This is the cursor
                sensitive help program
                Parameters passed identifying where
                the cursor was when FI was pressed
Appendix D - Source Listings

394.00 CSR    END
395.00 *V
396.00 *V    If ROLL DOWN key pressed, process read prior.
397.00 -------------------------------
398.00 *V
399.00 CSR    $EAI$D    IF$EQ    $FROLD
400.00 *V    Reset error indicators if roll
401.00 *V
402.00 *V    If error on read, set error.
403.00 *V
404.00 CSR    $EAI$D    IF$EQ    'L'
405.00 CSR    SETON    9341
406.00 CSR    MOVE    '0'    *IN, 40
407.00 CSR    *IN1$E1    IF$EQ    'L'
408.00 CSR    SDEST$E1    SETL$E1    9981
409.00 CSR    SSTOP
410.00 CSR    SDEST$E1    0020
411.00 CSR    SDEST$E1    0022
412.00 *V
413.00 *V    Load video screen data on roll keys.
414.00 *V    -------------------------------
415.00 *V
416.00 CSR    $EAI$D    IF$EQ    $FROLD
417.00 CSR    $EAI$D    OR$EQ    $FROLD
418.00 *V    Release record lock or report record in use.
419.00 *V
420.00 CSR    *IN9$9    IF$EQ    '0'
421.00 CSR    EXCEPT$UNELOCK
422.00 CSR    ELSE
423.00 CSR    CALL    'PROCEK'
424.00 CSR    END
425.00 *V
426.00 CSR    PARM    $#PELS
427.00 CSR    SETON    9341
428.00 CSR    MOVE    'L'    $NEK, 6
429.00 CSR    SSTOP    $NEK$E9
430.00 *V
431.00 CSR    SSTOP
432.00 *V
433.00 *V    Cost Center security edit.
434.00 *V
435.00 CSR    MOVE    $P2$9$G1    *$FILM
436.00 CSR    MOVE$E9$G2    *$FILE
437.00 CSR    *AUT$E9'1
438.00 CSR    #AUT$E9'1
439.00 CSR    $KRR$09$G9
440.00 CSR
441.00 CSR
442.00 CSR
443.00 CSR    *AUT$E9'1
444.00 CSR    #AUT$E9'1
445.00 CSR
446.00 CSR    MOVE    'L'    $SECUR
447.00 CSR
448.00 CSR    $SECUR$C$9$E9
449.00 CSR
450.00 CSR
451.00 *V
452.00 CSR
453.00 CSR
454.00 CSR
455.00 CSR    SSTOP    $NEK$E9
456.00 CSR
457.00 CSR
458.00 CSR
459.00 CSR    $EAI$D    IF$EQ    'L'
460.00 CSR    SETON    0193
461.00 CSR    SDEST$E1    SETL$E1
462.00 CSR    END
463.00 CSR    END
464.00 CSR
465.00 CSR
466.00 CSR
467.00 CSR
468.00 CSR
469.00 CSR

Could not find a match in the Function Key Definitions for the function key pressed, so program displays invalid Function Key message.
#. CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-53

Appendix D - Source Listings

```c
466.00 C*****************************************************************************
467.00 C* Copy Common Subroutine - Connect Security Check
468.00 C*
470.00 C/COPY JDECSH,C0000
472.00 C*****************************************************************************
473.00 C* SUBROUTINE SOCVL - Cursor Control Return Values
474.00 C*
475.00 C* by format, find the field to update and move in the
476.00 C* returned value. If the format is a subfile, the record
477.00 C* to change is found in SSOVR.
479.00 C*
480.00 CSR  SOCVL  $EFSR
482.00 C* ______ ______
483.00 CSR  $EVAL  IFEQ 'BLANK'
484.00 CS  MOVE *BLANK  $FVAL
485.00 CS  END
486.00 C* Return values for fields in format 9920011
487.00 C* 9920011
488.00 C* 9920011
489.00 CSR  $EINIT  IFEQ '9920011'
490.00 C*
491.00 CSR  $FNLN  IFEQ 'ACTION'
492.00 CSR  MOVE $FVAL  ACTION
493.00 CS  GOTO ENDCVL
494.00 C* ______
495.00 CDA  END
496.00 C* 9920011
497.00 CSR  $FNLN  IFEQ 'VDK1'
498.00 CS  MOVE $FVAL  VDK1
499.00 CS  GOTO ENDCVL
500.00 C* ______
501.00 CDA  END
502.00 C*
503.00 CDA  $FNLN  IFEQ 'VDK2'
504.00 CDA  MOVE $FVAL  VDK2
505.00 CS  GOTO ENDCVL
506.00 C* ______
507.00 CDA  END
508.00 C*
509.00 CSR  $FNLN  IFEQ 'VDKCC'
510.00 CDA  MOVE $FVAL  VDKCC
511.00 CS  GOTO ENDCVL
512.00 C* ______
513.00 CDA  END
514.00 C*
515.00 CDA  $FNLN  IFEQ 'VDKTY'
516.00 CDA  MOVE $FVAL  VDKTY
517.00 CS  GOTO ENDCVL
518.00 C* ______
519.00 CDA  END
520.00 C*
521.00 CDA  $FNLN  IFEQ 'VDKOT'
522.00 CDA  MOVE $FVAL  VDKOT
523.00 CS  GOTO ENDCVL
524.00 C* ______
525.00 CDA  END
526.00 C*
527.00 CDA  $FNLN  IFEQ 'VDKCT'
528.00 CDA  MOVE $FVAL  VDKCT
529.00 CS  GOTO ENDCVL
530.00 C* ______
531.00 CDA  END
532.00 C*
533.00 CDA  $FNLN  IFEQ 'VDK01'
534.00 CDA  MOVE $FVAL  VDK01
535.00 CS  GOTO ENDCVL
536.00 C* ______
537.00 CDA  END
538.00 C*
539.00 CDA  $FNLN  IFEQ 'VDK001'
540.00 CDA  MOVE $FVAL  VDK001
541.00 CS  GOTO ENDCVL
542.00 C* ______
```

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-53
Appendix D - Source Listings

541.00 CSR END
544.00 C' ***FLDN 1PEQ 'VDX001'  
546.00 CSR MOVEBL+4PLYAL VDX002
547.00 CSR GOTO END0VYL
548.00 C' ---- -----
549.00 CSR END
550.00 C'
552.00 CSR **FLDN 1PEQ 'VDX002'  
553.00 CSR GOTO END0VYL
554.00 C' ---- -----
555.00 CSR END
556.00 C'
557.00 CSR **FLDN 1PEQ 'VDX003'  
558.00 CSR MOVEBL+4PLYAL VDX004
559.00 CSR GOTO END0VYL
561.00 CSR END
562.00 C'
563.00 CSR **FLDN 1PEQ 'VDX005'  
564.00 CSR MOVEBL+4PLYAL VDX005
565.00 CSR GOTO END0VYL
566.00 C' ---- -----
567.00 CSR END
568.00 CSR END
569.00 C'
570.00 CSR END0VYL END5R
571.00 C'******Subroutine SD01 - Clear Fields******
572.00 C'
573.00 C' SUBROUTINE SD01 - Clear fields
574.00 C' -------------------------------------
575.00 C'
576.00 C' Processing: 1. Reset all video screen and data file fields
577.00 C' 2. For next transaction.
578.00 C' 3. Clear action code only if requested.
579.00 C'
580.00 CSR Z001 END5R
581.00 C'
582.00 C'
583.00 C' Reset fields for next transaction.
584.00 C'
585.00 CSR MOKEY CLEAR92801
586.00 CSR MOVE BLANE #HHLF
587.00 CSR MOVE BLANE #HHKX
588.00 CSR Z-ADD=ZERO #HSECL
589.00 CSR Z-ADD=ZERO #HSE2W
590.00 CSR MOVE BLANE VDX02
591.00 CSR MOVE BLANE VDX03
592.00 CSR MOVE BLANE VDX04
593.00 CSR MOVE BLANE VDX05
594.00 CSR MOVE BLANE VDX06
595.00 CSR MOVE BLANE VDX07
596.00 CSR MOVE BLANE VDX08
597.00 CSR MOVE BLANE VDX09
598.00 CSR MOVE BLANE VDX10
599.00 CSR MOVE BLANE VDX11
600.00 CSR MOVE BLANE VDX12
601.00 CSR MOVE BLANE VDX13
602.00 CSR MOVEBL+4L4H VDOM
603.00 CSR MOVE / ' #2D07 1
604.00 C'
605.00 C' Clear action code only if clear screen action.
606.00 C'
607.00 CSR 1PEQ #FLDR
608.00 CSR MOVE +ALL-C ' $353BT
609.00 CSR MOVEBL+4L4H +3M.41
610.00 CSR MOVE + ' ACTION 1
611.00 CSR Z-ADD=ZERO $353BT
612.00 CSR MOVE BLANE VDOM
613.00 CSR MOVE BLANE VDOM
614.00 CSR MOVE BLANE VDOM
615.00 CSR MOVE BLANE VDOM
616.00 CSR MOVE BLANE VDOM
617.00 CSR MOVE BLANE VDOM
618.00 CSR MOVE BLANE VDOM
619.00 CSR MOVE BLANE VDOM
620.00 CSR Z-ADD=ZERO #50IZ3 00
621.00 CSR END
622.00 C'
623.00 CSR ENDOVYL ENDSR

---

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 SD01 is executed.

Cleans the video fields
Appendix D - Source Listings

624.00  C***************************************************************
625.00  C*  SUBROUTINE E003 - Edit Key                               *
626.00  C*  Sets the file pointer and                            *
627.00  C*  edit the key                                         *
628.00  C*  Processing:  1. Clear error indicators and arrays.    *
629.00  C*  2. Load input keys.                                  *
630.00  C*  3. Validate Master file key.                        *
631.00  C*  4. Release master file record lock.                 *
632.00  C*  5. Load video screen output on inquiry.             *
633.00  C*  C***************************************************************
635.00  CSR  E003  B0506
636.00  C*  ----  -----                                          *
637.00  C*  Load data field dictionary parameters (one cycle only).
638.00  C*  C***************************************************************
640.00  CSR  9998  CASEQ' '  9998
641.00  C*  ----  -----                                          *
642.00  CSR  END
643.00  C*  C***************************************************************
644.00  C*  Reset error indicators and arrays.                  *
645.00  C***************************************************************
646.00  CSR  MOVE 'NIL'0'  $RSET  39
647.00  CSR  MOVE *BLANK  $RSETI  53
648.00  CSR  MOVEA$RESET  'IN, 41
649.00  CSR  MOVEA$RESETI  WRE, 2
650.00  CSR  CLEARER
651.00  C***************************************************************
652.00  C*  Load video input field for - Item ID
653.00  C***************************************************************
655.00  CSR  MOVEAEXIT  02M
656.00  CSR  ESEQ  C0312
657.00  C*  ----  -----                                          *
658.00  CSR  L-AUDNUMR  $RE08  90
659.00  CSR  MOVE  $RE08  QEXIT
660.00  C*  C***************************************************************
661.00  C*  Automatic Nest Number for - Item ID
662.00  C***************************************************************
663.00  CSR  *IN11  IPFQ '1'
664.00  CSR  VDI77  ANDEQ*BLANK
665.00  CSR  SETXY  91
666.00  CSR  *IN11  DQFQ'1'
667.00  CSR  MOVE  QEXIT  032X
668.00  CSR  CALL 'XCO10',  02
669.00  C*  ----  -----                                          *
670.00  CSR  PARM  QEXIT  RNSY  4
671.00  CSR  PARM  9565X
672.00  CSR  PARM  *SEQ  $MYTNO  90
673.00  CSR  MOVE  ENTRINO  QEXIT
674.00  CSR  MOVE  ENTRINO  VEXIT
675.00  CSR  QEXIT  $SETLIP2$001  020L
676.00  CSR  END
677.00  CSR  END
678.00  C***************************************************************
679.00  CSR  QEXIT01  CHAIN120001  2009
680.00  C*  C***************************************************************
681.00  C*  COST CENTER security edit.                          *
682.00  C***************************************************************
683.00  CSR  MOVEL'$S2001'  'FILE'
684.00  CSR  MOVEGQX000  &NOU
685.00  CSR  #A07  IPFQ '1'
686.00  CSR  #PA07  ANDEQ'1'
687.00  CSR  ESEQ  $C0000
688.00  C*  ----  -----                                          *
689.00  CSR  END
690.00  CSR  #A07  IPFQ '1'
691.00  CSR  #PA07  ANDEQ'1'
692.00  C*  #PA07  ANDEQ'1'
693.00  CSR  MOVE  '1'  $SEQCR  1
694.00  CSR  END
695.00  C*  If security violation, set error condition.
696.00  C***************************************************************
697.00  C*  C***************************************************************
698.00  CSR  $SEQCR  IPFQ '1'
699.00  CSR  MOVE  '1'  WRE, 6
700.00  CSR  SETTOP  9341
Appendix D - Source Listings

```
701.00 CSR  MOVE ' '  $SSEPCHR 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  *IN90  IPEQ '1'
709.00 CSR  *IN31  COMP '0'
710.00 C*  41 *ERROR*
711.00 CSR  ELAR
712.00 CSR  *IN31  COMP '1'
713.00 CSR  BND
714.00 C*  If indicator 41 on, invalid key for action code.
715.00 C*
716.00 CSR  *IN41  IPEQ '3'
717.00 CSR  MOVE '1'  @MV, 2
718.00 CSR  SITON   91
719.00 CSR  END
720.00 C*
721.00 C*  If indicator 60 on, record in use.
722.00 C*
723.00 CSR  *IN99  IPEQ '1'
724.00 CSR  CALL 'PDELOCK'  01
725.00 C*  41
726.00 CSR  PAIN  ----  #RFDS
727.00 CSR  MOVE '1'  @MV, 0
728.00 CSR  SITON   #31
729.00 CSR  END
730.00 C* ---------------------------------------------------------------
731.00 C*  If not inquiry, skip remainder of subroutine.
732.00 C*
733.00 CSR  *IN34  CABEQ'0'  END003
734.00 CSR  OVR4  -------
735.00 C*
736.00 C*  ***************************************************************
737.00 C*  Release record lock on master file
738.00 C*
739.00 CSR  *IN90  IPEQ '0'
740.00 CSR  (IN99  ANDEQ'0'
741.00 CSR  EXCPUNLOCK
742.00 CSR  END
743.00 C*  If errors, skip remainder of subroutine.
744.00 C*
745.00 CSR  *IN93  CABEQ'1'  END003
746.00 C*
747.00 C*
748.00 C*
749.00 C*  ***************************************************************
750.00 C*  Move data base information to video screen.
751.00 C*
752.00 C*  EKXR 2004
753.00 CSR  -------
754.00 CSR  ----
755.00 C*
756.00 CSR  END003  END8R
757.00 C*  ***************************************************************
758.00 C*
759.00 C*  Copy common subroutine - right justify numeric fields
760.00 C*
761.00 C*/COPY2_CPY2_C0012
762.00 C*  ***************************************************************
763.00 C*
764.00 C*  SUBROUINE 2004 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 are 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*  Data fields must be converted from their
774.00 C*  Internal format of month, day and year or
775.00 C*  Julian to the system format using program
776.00 C*  X0020.
```
Appendix D - Source Listings

779.00 C*    8094 BEGS
780.00 C*    ---- -----
781.00 C*    ---- -----
782.00 C*    ---- -----
783.00 C*    Move to output -Description for Cost Center
784.00 C*    ---- -----
785.00 CSE CALL 'X0005' 81
786.00 C*    ---- -----
787.00 CSE DRM 'BLANK' 81 MOD 1
788.00 CSE DRM '1' 81 MOD 1
789.00 CSE DRM 'QXCY' PSRCU 12
790.00 CSE DRM 'QXCY' PSRCM 4
791.00 CSE DRM 'QXCY' U0000
792.00 C*    ---- -----
793.00 CSE MOVE *BLANK VC0001
794.00 CSE JF EQ *BLANK
795.00 CSE MOVEMOD01 VC0001
796.00 CSE ENDC
797.00 C*    ---- -----
798.00 C*    ---- -----
799.00 C*    Description display for - Item Type
800.00 C*    ---- -----
801.00 CSE CLEARD005U
802.00 CSE NOVELLE001 USX
803.00 CSE MOVE RXCY RXT
804.00 CSE MOVE QXCY RXT
805.00 CSE CALL 'X0005' 81
806.00 C*    ---- -----
807.00 CSE DSRM *Q0005U
808.00 CSE MOVE *BLANK VC0002
809.00 CSE JF EQ 'O'
810.00 CSE NOVELLE001 VC0002
811.00 CSE ENDC
812.00 C*    ---- -----
813.00 C*    ---- -----
814.00 C*    Description display for - Item Unit of Measure
815.00 C*    ---- -----
816.00 CSE CLEARD005U
817.00 CSE NOVELLE001 USY
818.00 CSE MOVE RXCY RXT
819.00 CSE MOVE QXCY RXT
820.00 CSE CALL 'X0005' 81
821.00 C*    ---- -----
822.00 CSE DRM 'QXCY' U0000
823.00 CSE MOVE *BLANK VC0003
824.00 CSE JF EQ 'O'
825.00 CSE NOVELLE001 VC0003
826.00 CSE ENDC
827.00 C*    ---- -----
828.00 C*    ---- -----
829.00 C*    Description display for - Item category code 001
830.00 C*    ---- -----
831.00 CSE CLEARD005U
832.00 CSE NOVELLE001 U000
833.00 CSE MOVE RXCY RXT
834.00 CSE MOVE QXCY RXT
835.00 CSE CALL 'X0005' 81
836.00 C*    ---- -----
837.00 CSE DSRM U0000
838.00 CSE MOVE *BLANK VC0004
839.00 CSE JF EQ 'O'
840.00 CSE NOVELLE001 VC0004
841.00 CSE ENDC
842.00 C*    ---- -----
843.00 C*    ---- -----
844.00 C*    Description display for - Item Category Code 002
845.00 C*    ---- -----
846.00 CSE CLEARD005U
847.00 CSE NOVELLE002 USY
848.00 CSE MOVE RXCY RXT
849.00 CSE MOVE QXCY RXT
850.00 CSE CALL 'X0005' 81
851.00 C*    ---- -----
852.00 CSE DRM U0000
853.00 CSE MOVE *BLANK VC0005
854.00 CSE JF EQ 'O'

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-57
Appendix D - Source Listings

065.00  CBR       MOVEL@U1001    VCC005
066.00  CBR       END
067.00  C*    Description display for - Item Category Code 002
068.00  CBR
069.00  CBR       CLEAR@1000SU
070.00  CBR       MOVEL@S@10003  #USY
071.00  CBR       MOV@D@S@10003  #DET
072.00  CBR       MOV@Q@D@S@10003  #UXY
073.00  C*    CALL @X@Q@D@S@10003
074.00  CBR       IFEQ 'O'
075.00  CBR       MOVEL@U1001    VCC005
076.00  CBR       END

077.00  C*    Description display for - Item Category Code 004
078.00  CBR
079.00  CBR       CLEAR@1000SU
080.00  CBR       MOVEL@S@10004  #USY
081.00  CBR       MOV@D@S@10004  #DET
082.00  CBR       MOV@Q@D@S@10004  #UXY
083.00  C*    CALL @X@Q@D@S@10004
084.00  CBR       IFEQ 'O'
085.00  CBR       MOVEL@U1001    VCC007
086.00  CBR       END

087.00  C*    Description display for - Item Category Code 005
088.00  CBR
089.00  CBR       CLEAR@1000SU
090.00  CBR       MOVEL@S@10005  #USY
091.00  CBR       MOV@D@S@10005  #DET
092.00  CBR       MOV@Q@D@S@10005  #UXY
093.00  C*    CALL @X@Q@D@S@10005
094.00  CBR       IFEQ 'O'
095.00  CBR       MOVEL@U1001    VCC008
096.00  CBR       END

097.00  C*    Description display for - Cost Center
098.00  CBR
099.00  CBR       MOVE @S@1@BLANK  @S@INER
100.00  CBR       MOVEL@S@10000  #S@INER
101.00  CBR       MOVE @D@1@S@10000  #DET
102.00  CBR       MOVE @Q@D@1@S@10000  #UXQ
103.00  CBR       MOVE @D@1@S@10000  #DET
104.00  CBR       MOVE @Q@D@1@S@10000  #UXQ
105.00  CBR       MOVE ' '  #S@1@CHAR
106.00  CBR       MOVE ' '  #S@1@CHAR
107.00  CBR       MOV@D@S@10001  #DET
108.00  CBR       MOVE @Q@D@S@10000  #UXQ
109.00  CBR       MOV@D@S@10000  #DET
110.00  CBR       MOVE @Q@D@S@10000  #UXQ
111.00  CBR       MOVE ' '  #S@1@CHAR
112.00  CBR       MOV@D@S@10001  #DET
113.00  CBR       MOVE @Q@D@S@10000  #UXQ
114.00  CBR       MOV@D@S@10000  #DET
115.00  CBR       MOVE ' '  #S@1@CHAR
116.00  CBR       MOV@D@S@10001  #DET
117.00  CBR       MOVE @Q@D@S@10000  #UXQ
118.00  CBR       MOV@D@S@10000  #DET
119.00  CBR       MOVE ' '  #S@1@CHAR
120.00  CBR       MOV@D@S@10001  #DET
121.00  CBR       MOV@D@S@10000  #DET
122.00  CBR       MOV@D@S@10000  #DET
123.00  CBR       MOVE ' '  #S@1@CHAR
124.00  CBR       MOVE ' '  #S@1@CHAR
125.00  CBR       MOV@D@S@10001  #DET
126.00  CBR       MOV@D@S@10000  #DET
127.00  CBR       MOV@D@S@10000  #DET
128.00  CBR       MOVE ' '  #S@1@CHAR
129.00  CBR
130.00  CBR       MOVE ' '  #S@1@CHAR
131.00  CBR

Editing information retrieved in 9998
Copy module to edit field for use on screen/report
998.00  CBR

9-58  JD Edwards World
Appendix D - Source Listings

931.00 CSR MOVE Q0000T $6IDAT 6
932.00 CSR MOVE 'HBLANK' $EDAT 8
934.00 CSR MOVE 'SYSDATE' $FTM 7
935.00 CSR MOVE 'SYSVAL' 'TIPMT' 7
936.00 CSR MOVE 'SYSVAL' $FEP 7
937.00 CSR MOVE ' ' $FSEP 7
938.00 CSR CALL X0026 , 81
939.00 C* --- ------
940.00 CSR FARM $IDAT
941.00 CSR FARM $EDAT
942.00 CSR FARM $FTM
944.00 CSR FARM $FEP
945.00 CSR FARM $FSEP
946.00 C* NOVEL+READ VEXIT
947.00 C*-------------------------------------------------------------
948.00 C*
949.00 C* Move to output - Item ID
950.00 C*
951.00 CSR MOVE 'HBLANK' $INBR
952.00 CSR MOVE 'DXXXTY' $INBR
953.00 CSR MOVE 'XEXIT' $FMT
954.00 CSR MOVE 'MXXQT' $FMT
955.00 CSR MOVE 'MXXQT' $FRO
956.00 CSR MOVE 'MXXQT' $ENU
957.00 CSR MOVE 'XEXIT' $XDO
958.00 CSR MOVE ' ' $XDO
959.00 C*
960.00 C* Move to output - Quantity - On hand
961.00 C*
962.00 CSR MOVE 'HBLANK' $INBR
963.00 CSR MOVE 'DXXQT' $INBR
964.00 CSR MOVE 'XEXIT' $FMT
965.00 CSR MOVE 'MXXQT' $FRO
966.00 CSR MOVE 'MXXQT' $ENU
967.00 CSR MOVE 'MXXQT' $XDO
968.00 CSR MOVE ' ' $XDO
969.00 C*
970.00 C* Move to output - Item Type
971.00 C*
972.00 CSR MOVE 'DXXXTY' VEXIT
973.00 C*-------------------------------------------------------------
974.00 C*
975.00 C* Move to output - Item Unit of Measure
976.00 C*
977.00 CSR MOVE 'DXXXYM' VEXIT
978.00 C*-------------------------------------------------------------
979.00 C*
980.00 C* Move to output - Item Category Code 001
981.00 C*
982.00 CSR MOVE 'HBLANK' $INBR
983.00 CSR MOVE 'DXXXY001' $INBR
984.00 CSR MOVE 'XEXIT' $FMT
985.00 CSR MOVE 'MXXQT' $FRO
986.00 CSR MOVE 'MXXQT' $ENU
987.00 CSR MOVE 'MXXQT' $XDO
988.00 CSR MOVE ' ' $XDO
989.00 C*-------------------------------------------------------------
990.00 C*
1087.00  CSK  MOVE +BLANK  $SINER
1088.00  CSK  MOVE $LX2005  $SINER
1089.00  CSK  MOVE $LX2005  $1UTY
1090.00  CSK  MOVE $LX2005  $1UDR
1091.00  CSK  MOVE $LX2005  $EC
1092.00  CSK  MOVE $LX2005  $EDST
1093.00  CSK  MOVE $LX2005  $1UTD
1094.00  CSK  MOVE $LX2005  $DALR
1095.00  CSK  MOVE 'X'  $EDCRR
1096.00  CSK  MOVE 'Y'  $EDCRR
1097.00  CSK  EISO  C019101
1098.00  C'  ------
1099.00  CSK  TALK  IFREQ  'L'
1100.00  CSK  MOVE $SINER  VUX205
1101.00  CSK  ESEE
1102.00  CSK  MOVE $SINER  VUX205
1103.00  CSK  END
1104.00  C'-------------------------------------------------------------------------------------------------''
1105.00  CSK  END024  ENDGR
1106.00  C'-------------------------------------------------------------------------------------------------''
1107.00  C'  Copy Common Subroutine - Format Numeric Fields for Output with Override
1108.00  C'  1
1109.00  C'  2
1110.00  C/COPY JDCRVY,CO2141
1111.00  C'-------------------------------------------------------------------------------------------------''
1112.00  C'  SUBROUTINE A603 - Scrub Input
1113.00  C'                         Validates and edits data
1114.00  C'                     entered by the user
1115.00  C'  PROCESSING:  1. Validate all video input.
1116.00  C'  All numeric fields must be processed
1117.00  C'  through subroutines COOLZ and COOL5 in order
1118.00  C'  to scrub the alpha input field and correct
1119.00  C'  15 digits and 0 decimals.
1120.00  C'  Date fields must be converted from system
1121.00  C'  format to their internal format of month,
1122.00  C'  day and year or Julian using program 22026.
1123.00  C'  2. Update data record fields from video.
1124.00  C'  
1125.00  C'  
1126.00  C'  
1127.00  CSK  A603  ENDGR
1128.00  C'  ------
1129.00  C'  if not addition or change, bypass subroutine
1130.00  C'  
1131.00  C'  +INH1  IFREQ  'O'
1132.00  C'  +INH2  AMEND 'O'
1133.00  C'  GOTO END0005
1134.00  C'  ------
1135.00  C'  END
1136.00  C'  
1137.00  C'  
1138.00  C'  
1139.00  C'  
1140.00  C'  Scrub and edit - Core Center
1141.00  C'  
1142.00  CSK  CALL 'I006S'  00
1143.00  C'  ------
1144.00  CSK  PAFM 'L'  PFMX  1
1145.00  CSK  PAFM 'L'  PFMX  1
1146.00  CSK  PAFM  $20CC  PFMX  12
1147.00  CSK  PAFM +BLANK  PFSMX  4
1148.00  CSK  PAFM  IO0GC
1149.00  C'  
1150.00  CSK  PSHRM  IFREQ  'L'
1151.00  CSK  SETON  4393
1152.00  CSK  MOVE $LX2005  END,10
1153.00  CSK  MOVE 'L'  $1UTY,10
1154.00  CSK  END
1155.00  CSK  MOVE '22M'  000
1156.00  C'-------------------------------------------------------------------------------------------------''
1157.00  C'  
1158.00  C'  
1159.00  C'  
1160.00  C'  
1161.00  C'  
1162.00  C'  
1163.00  C'  Set default value - Description
Appendix D - Source Listings

```
1197.00  CSE  LD   3  QXEDS  2  QXEDS
1590.00  CSE  MOVE  'L'  QXEDS  03  GMX
1590.00  CSE  END
1592.00  C*--------------------------------------------------------------
1593.00  C*  Edit Julian Date - Data Last Ship
1594.00  C*--------------------------------------------------------------
1595.00  C*  Scrub and edit - Item ID
1596.00  C*--------------------------------------------------------------
1597.00  C*  Set default value - Item ID
1598.00  C*--------------------------------------------------------------
1599.00  C*  Set default value - Item ID
1600.00  C*--------------------------------------------------------------
1601.00  C*  Set default value - Item ID
1602.00  C*--------------------------------------------------------------
1603.00  C*  Set default value - Item ID
1604.00  C*--------------------------------------------------------------
1605.00  C*  Set default value - Item ID
1606.00  C*--------------------------------------------------------------
1607.00  C*  Set default value - Item ID
1608.00  C*--------------------------------------------------------------
1609.00  C*  Set default value - Item ID
1610.00  C*--------------------------------------------------------------
1611.00  C*  Set default value - Item ID
1612.00  C*--------------------------------------------------------------
1613.00  C*  Set default value - Item ID
1614.00  C*--------------------------------------------------------------
1615.00  C*  Set default value - Item ID
1616.00  C*--------------------------------------------------------------
1617.00  C*  Set default value - Item ID
1618.00  C*--------------------------------------------------------------
1619.00  C*  Set default value - Item ID
1620.00  C*--------------------------------------------------------------
1621.00  C*  Set default value - Item ID
1622.00  C*--------------------------------------------------------------
1623.00  C*  Set default value - Item ID
1624.00  C*--------------------------------------------------------------
1625.00  C*  Set default value - Item ID
1626.00  C*--------------------------------------------------------------
1627.00  C*  Set default value - Item ID
1628.00  C*--------------------------------------------------------------
1629.00  C*  Set default value - Item ID
1630.00  C*--------------------------------------------------------------
1631.00  C*  Set default value - Item ID
1632.00  C*--------------------------------------------------------------
1633.00  C*  Set default value - Item ID
1634.00  C*--------------------------------------------------------------
```

Appendix D - Source Listings

```
0116.00  CSX  45,1  TPEQ '==='
0120.00  CSX  MOVE ' '  40,1
0121.00  CSX  L-ADD  $M
0122.00  CSX  $M  DOWLD40
0123.00  CSX  $40,DM  TPEQ '==='
0124.00  CSX  MOVE ' '  40,2M
0126.00  CSX  END
0126.00  CSX  ADD  1  $M
0127.00  CSX  END
0128.00  CSX  NOVER@10,2  QXXTY
0129.00  CSX  END
0130.00  CSX  END
0131.00  CSX  END
0132.00  C'  Edit allowed values - Item Type
0134.00  C'
0135.00  CSX  AXXTY  IPHE *BLANK
0136.00  CSX  AXXTY  TPEQ 'X8P'
0137.00  CSX  QXXTY  ANDXQ*BLANK
0138.00  CSX  MOVE '1'  @MK,03
0139.00  CSX  SETUP  4493
0140.00  CSX  ELSE
0141.00  CSX  NOVERA@XY  450
0142.00  CSX  MOVE '*HVVAL'  $AV
0143.00  CSX  EXIT  3597
0144.00  C'  ----- ----- ----
0145.00  CSX  MOVE ' '  SERTST  1
0146.00  CSX  MOVE 'BLANK'  $WRDLD 10
0147.00  CSX  NOVERE@XY  $WRSL0
0148.00  CSX  HAV,1  IPHE '*HVVAL'
0149.00  CSX  LORFUP@SY  01
0150.00  CSX  'INOC'  TPEQ '0'
0151.00  CSX  MOVE '1'  SERTST
0152.00  CSX  END
0153.00  CSX  SERTST  TPEQ '1'
0154.00  C'  MOVE '1'  $*,07
0155.00  CSX  SETUP  4493
0156.00  CSX  END
0157.00  CSX  END
0158.00  CSX  END
0159.00  CSX  END
0160.00  C'  Edit upper and lower range - Item Type
0162.00  C'
0163.00  CSX  LQXXTY  IPHE 'BLANK'
0164.00  CSX  LXXTY  MOVE '1'  SERTST
0165.00  CSX  QXXTY  TPEQ 'X8P'
0166.00  CSX  QXXTY  ANDXQ*BLANK
0167.00  CSX  MOVE ' '  SERTST
0168.00  CSX  END
0169.00  CSX  SERTST  TPEQ '1'
0170.00  CSX  MOVE '1'  @MK,07
0171.00  CSX  SETUP  4493
0172.00  CSX  END
0173.00  CSX  END
0174.00  CSX  END
0175.00  C'  Edit from User Defined Codes - Item Type
0176.00  C'
0177.00  CSX  RAXTY  IPHE 'BLANK'
0178.00  CSX  CXR  CLEARX0000S
0179.00  C'  NOVERA@XY  $#SY
0180.00  CSX  MOVE RAXTY  $9AT
0181.00  CSX  MOVE QXXTY  $1FC
0182.00  CSX  CALL 'X0001'  01
0183.00  C'  CXR  100000S
0184.00  CSX  HVERA  TPEQ '1'
0186.00  CSX  MOVE '1'  @MK,09
0187.00  CSX  SETUP  4493
0188.00  CSX  END
0189.00  CSX  END
0190.00  C'  Scrub and edit - Item Unit of Measure
0192.00  C'
0194.00  CSX  NOVERA@XY  QXXTY
0195.00  CSX  END
0196.00  CSX  END
```
Appendix D - Source Listings

1386.00 C*  Set default value - item Unit of Measure
1387.00 C*
1388.00 C*  GXXM  IPEQ *BLANK
1389.00 C*  KXUM  IPEX *BLANK
1400.00 C*  NOTE#XUM  000
1401.00 C*  NOTEA40  GXXM
1402.00 C*  @A40,1  IPEQ ' ' ' '
1403.00 C*  NOTE ' ' ' '  @A40,1
1404.00 C*  S-XUM  #M
1405.00 C*  MOWLEA40
1406.00 C*  @A40,#M  IPEQ ' ' ' '
1407.00 C*  NOTE ' ' ' '  @A40,#M
1408.00 C*  END
1409.00 C*  END  1  #M
1410.00 C*  END
1411.00 C*  END  @A40,2
1412.00 C*  END
1413.00 C*  END
1414.00 C*  END
1415.00 C*
1416.00 C*  Edit allowed values - item Unit of Measure
1417.00 C*
1418.00 C*  AXXM  IPHE *BLANK
1419.00 C*  AXXM  IPHE  ' ' ' ' ' '
1420.00 C*  AXXM  EDELETE  #M
1421.00 C*  NOTE '1'  @AHE,03
1422.00 C*  SETON  4793
1423.00 C*  блок
1424.00 C*  NOTEBXUM  @A40
1425.00 C*  NOTE  ' ' ' '  MVAL  @A40
1426.00 C*  NOTE  ' ' ' '  997
1427.00 C*  END
1428.00 C*  NOTE ' ' ' '  @SETST  1
1429.00 C*  NOTE ' ' ' '  @SWEEP  1 0
1430.00 C*  NOTE@XUM  @MHE,0
1431.00 C*  AXXM  IPHE  ' ' ' ' ' '
1432.00 C*  AXXM  IPHE  ' ' ' ' ' '
1433.00 C*  AXXM  IPHE  ' ' ' ' ' '
1434.00 C*  AXXM  IPHE  ' ' ' ' ' '
1435.00 C*  AXXM  IPHE  ' ' ' ' ' '
1436.00 C*  AXXM  IPHE  ' ' ' ' ' '
1437.00 C*  AXXM  IPHE  ' ' ' ' ' '
1438.00 C*  AXXM  IPHE  ' ' ' ' ' '
1439.00 C*  AXXM  IPHE  ' ' ' ' ' '
1440.00 C*  AXXM  IPHE  ' ' ' ' ' '
1441.00 C*  AXXM  IPHE  ' ' ' ' ' '
1442.00 C*  AXXM  IPHE  ' ' ' ' ' '
1443.00 C*  AXXM  IPHE  ' ' ' ' ' '
1444.00 C*  AXXM  IPHE  ' ' ' ' ' '
1445.00 C*  AXXM  IPHE  ' ' ' ' ' '
1446.00 C*  AXXM  IPHE  ' ' ' ' ' '
1447.00 C*  AXXM  IPHE  ' ' ' ' ' '
1448.00 C*  AXXM  IPHE  ' ' ' ' ' '
1449.00 C*  AXXM  IPHE  ' ' ' ' ' '
1450.00 C*  AXXM  IPHE  ' ' ' ' ' '
1451.00 C*  AXXM  IPHE  ' ' ' ' ' '
1452.00 C*  AXXM  IPHE  ' ' ' ' ' '
1453.00 C*  AXXM  IPHE  ' ' ' ' ' '
1454.00 C*  AXXM  IPHE  ' ' ' ' ' '
1455.00 C*  AXXM  IPHE  ' ' ' ' ' '
1456.00 C*  AXXM  IPHE  ' ' ' ' ' '
1457.00 C*  AXXM  IPHE  ' ' ' ' ' '
1458.00 C*  AXXM  IPHE  ' ' ' ' ' '
1459.00 C*  AXXM  IPHE  ' ' ' ' ' '
1460.00 C*  AXXM  IPHE  ' ' ' ' ' '

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-65
Appendix D - Source Listings

2473.00  C*---------------------------------------------------------------------
2474.00  C*  scrub and edit - item category code 001
2475.00  C*
2476.00  C*  MOVELV0001  QIX001
2477.00  C*
2478.00  C*  set default value - item category code 001
2479.00  C*
2480.00  C*  QXX001  IPFX *BLANK
2481.00  C*  DXX001  IPFX *BLANK
2482.00  C*  MOVELV0002  Q40
2483.00  C*  MOVELV0002  Q20001
2484.00  C*  @40.1  IPFX '***
2485.00  C*  MOVE *'  Q40.1
2486.00  C*  Z-AUD2  *M
2487.00  C*  $M  DONLEADO
2488.00  C*  @40.0M  IPFX
2489.00  C*  NOTE '***'  Q40.0M
2490.00  C*  END
2491.00  C*  END
2492.00  C*  ZDD  1  *M
2493.00  C*  END
2494.00  C*  NOTE  *H
2495.00  C*  END
2496.00  C*  END
2497.00  C*  END
2498.00  C*
2499.00  C*  Edit allowed values - Item Category Code 001
2500.00  C*
2501.00  C*  AXX001  IPFX *BLANK
2502.00  C*  AAX001  IPFX *HBD
2503.00  C*  AQXX001  AXX001 *BLANK
2504.00  C*  NOTE '1'  @MK, 02
2505.00  C*  SETOV  45200
2506.00  C*  ELER
2507.00  C*  NOTE  AXX0001  Q40
2508.00  C*  NOTE 'HIBUL'  2A7
2509.00  C*  ELER  *C087
2510.00  C*
2511.00  C*  NOTE '***'  $EXT  1
2512.00  C*  NOTE '***'  $XK010  1D1
2513.00  C*  NOTE  MOVELV0001  @MK010
2514.00  C*  GAXV,1  IPFX 'HIBUL'
2515.00  C*  $XRK010  LOZUXV01  81
2516.00  C*  *XMI1  IPFX 'O'
2517.00  C*  NOTE '1'  $EXT
2518.00  C*  END
2519.00  C*  NOTE  '1'  $MK, 07
2520.00  C*  SETOV  1G200
2521.00  C*  END
2522.00  C*  END
2523.00  C*  END
2524.00  C*  END
2525.00  C*  END
2526.00  C*
2527.00  C*  Edit upper and lower range - Item Category Code 001
2528.00  C*
2529.00  C*  LAX001  IPFX *BLANK
2530.00  C*  MOVE '1'  $EXT
2531.00  C*  AQXX001  IPFX LAX001
2532.00  C*  AQXX001  AXX0001
2533.00  C*  MOVE '1'  $EXT
2534.00  C*  END
2535.00  C*  NOTE  '1'  @MK, 07
2536.00  C*  SETOV  4893
2537.00  C*  END
2538.00  C*  END
2539.00  C*  END
2540.00  C*
2541.00  C*  Edit from user defined codes - Item Category code 001
2542.00  C*
2543.00  C*  RAX001  IPFX *BLANK
2544.00  C*  CMAX010001
2545.00  C*  MOVELV0001  $USY
2546.00  C*  MOVE  XAX001  $USY
2547.00  C*  NOTE  XQXX001  $USY
2548.00  C*  CALL  XQ0001
2549.00  C*  ----- --------  01
Appendix D - Source Listings

1550.00 CDR  Parma  I000SU
1551.00 CDR  SUBRX  IPEQ '3'
1552.00 CDR  MOVZ '1'  @MK,00
1553.00 CDR  GETON  4693
1554.00 CDR  END
1555.00 CDR  END
1556.00 CDR  

1557.00 C*  Scrub and edit - Item Category Code 002
1558.00 C*
1559.00 CDR  MOVEXX002  QEX002
1560.00 C*
1561.00 C*  Set default value - Item Category Code 002
1562.00 C*
1563.00 C  
1564.00 CDR  QEX002  IPEQ 'AELANE'
1565.00 CDR  CEXK02  IPEQ 'AELANE'
1566.00 CDR  MOVZAXX002  @40
1567.00 CDR  MOVZAXS002  QEX002
1568.00 CDR  @40,1  IPEQ '***'
1569.00 CDR  MOVZ '/'  @40,1
1570.00 CDR  Z-ADD2  @M
1571.00 CDR  @M  DOWN240
1572.00 CDR  @40, @M  IPEQ '***'
1573.00 CDR  MOVZ '/'  @40, @M
1574.00 CDR  END
1575.00 CDR  ADD1  @M
1576.00 CDR  END
1577.00 CDR  MOVZAXS002, @QEX002
1578.00 CDR  END
1579.00 CDR  END
1580.00 CDR  END
1581.00 C*  
1582.00 C*  Edit allowed values - Item Category Code 002
1583.00 C*
1584.00 CDR  AEXK002  IPEQ 'AELANE'
1585.00 CDR  AEXK02  IPEQ 'AELANE'
1586.00 CDR  QEX002  IPEQ 'AELANE'
1587.00 CDR  MOVZ '1'  @MK,00
1588.00 CDR  GETON  4693
1589.00 CDR  EL22
1590.00 CDR  MOVZAXS002  @40
1591.00 CDR  MOVZ 'MXVAL'  @AV
1592.00 CDR  EXR  CD57
1593.00 C*  
1594.00 CDR  MOVZ '/'  $2ERTST  1
1595.00 CDR  MOVZ 'ELANE'  $2REL1 10
1596.00 CDR  MOVZAXXX002  $2REL10
1597.00 CDR  @AV,1  IPEQ 'MXVAL'
1598.00 CDR  $2REL10  LAX002A0  81
1599.00 CDR  *IS1  IPEQ '0'
1600.00 CDR  MOVZ '1'  $2ERTST
1601.00 CDR  END
1602.00 CDR  $2ERTST  IPEQ '1'
1603.00 CDR  MOVZ '1'  @MK,07
1604.00 CDR  GETON  4693
1605.00 CDR  END
1606.00 CDR  END
1607.00 CDR  END
1608.00 CDR  END
1609.00 C*  
1610.00 C*  Edit upper and lower range - Item Category Code 002
1611.00 C*
1612.00 CDR  LAX002  IPEQ 'AELANE'
1613.00 CDR  MOVZ '1'  $2ERTST
1614.00 CDR  QEX002  IPEQ LAX002
1615.00 CDR  QEX002  IPEQ '1'
1616.00 CDR  MOVZ '/'  $2ERTST
1617.00 CDR  END
1618.00 CDR  $2ERTST  IPEQ '1'
1619.00 CDR  MOVZ '1'  @MK,07
1620.00 CDR  GETON  4693
1621.00 CDR  END
1622.00 CDR  END
1623.00 C*  
1624.00 C*  Edit from User Defined Codes - Item Category Code 002
1625.00 C*
1626.00 CDR  RAX002  IPEQ 'AELANE'
Appendix D - Source Listings

```plaintext
L527.00  CSE  CLASSIFICATION
L528.00  CSE  MOVESAS0002  SUX
L529.00  CSE  MOVE  ASX002  HCMT
L530.00  CSE  MOVE  QNX002  QNX
L531.00  CSE  CALL  'QNX002'  91
L532.00  C*  --------
L533.00  CSE  DASM  900000
L534.00  CSE  83301  IP68Q '1'
L535.00  CSE  MOVE  '1'  $MN,09
L536.00  CSE  SETCN  4993
L537.00  CSE  END
L538.00  CSE  END
L539.00  C*  ---------------------------------------
L540.00  C*  
L541.00  C*  Scrub and edit - Item Category Code 003
L542.00  C*  
L543.00  CSE  MOVEVEX003  QXX003
L544.00  C*  
L545.00  C*  set default value - Item Category Code 003
L546.00  C*  
L547.00  CSE  QXX003  IP70Q *BLANK
L548.00  CSE  DXX003  IPH0 *BLANK
L549.00  CSE  MOVENVXX003  940
L550.00  CSE  MOVENV800  QXX003
L551.00  CSE  940,1  IP68Q ' *'
L552.00  CSE  MOVR ' '  940,1
L553.00  CSE  940,2  DXX003  $MN
L554.00  CSE  $MN  DONLEV20
L555.00  CSE  940,3  IP68Q ' *'
L556.00  CSE  MOVE  ' '  940,3
L557.00  CSE  END
L558.00  CSE  ADD  1  $MN
L559.00  CSE  END
L560.00  CSE  MOVENV800,2  QXX003
L561.00  CSE  END
L562.00  CSE  END
L563.00  CSE  END
L564.00  C*  
L565.00  C*  Edit allowed values - Item Category Code 003
L566.00  C*  
L567.00  CSE  ASX002  IPX0 *BLANK
L568.00  CSE  ASX002  IP68Q *H2V*
L569.00  CSE  QXX003  ANDR0 *BLANK
L570.00  CSE  MOVR '1'  $MN,03
L571.00  CSE  SETCN  5993
L572.00  CSE  BLSH
L573.00  CSE  MOVENV8003  940
L574.00  CSE  MOVE  *HEVAL  $AV
L575.00  CSE  ENGR  9997
L576.00  C*  --------
L577.00  CSE  MOVR ' '  $SPECFT  1
L578.00  CSE  MOVE  *BLANK $9RX1 10
L579.00  CSE  MOVENVXX003  $9RX10
L580.00  CSE  $AV,1  IPX0 *HEVAL
L581.00  CSE  S9RX0  LORUSAV  91
L582.00  CSE  81801  IP68Q '0'
L583.00  CSE  MOVE  '1'  $SPECFT
L584.00  CSE  END
L585.00  CSE  $SPECFT  IP68Q '1'
L586.00  CSE  MOVR '1'  $MN,07
L587.00  CSE  SETCN  5993
L588.00  CSE  END
L589.00  CSE  END
L590.00  CSE  END
L591.00  CSE  END
L592.00  C*  
L593.00  C*  Edit upper and lower range - Item Category Code 003
L594.00  C*  
L595.00  CSE  LAX002  IPX0 *BLANK
L596.00  CSE  MOVR '1'  $SPECFT
L597.00  CSE  QXX003  IPX0 LAX003
L598.00  CSE  QXX003  ANGLENX003
L599.00  CSE  MOVE  '0'  $SPECFT
L600.00  CSE  END
L601.00  CSE  $SPECFT  IP68Q '1'
L602.00  CSE  MOVR '1'  $MN,07
L603.00  CSE  SETCN  5993
```

9-68  JD Edwards World
Appendix D - Source Listings

1764.00 CSE EBD
1765.00 CSE EBD
1766.00 C* Edit from User Defined Codes - Item Category Code 003
1767.00 C* 
1768.00 C* 
1769.00 CSE R0X003 IFPHR *BLANK
1770.00 CSE CLEEX100000
1771.00 CSE MOVE R0X003 #X00
1772.00 CSE MOVE R0X003 #X00
1773.00 CSE MOVE X0X0003 #X00
1774.00 CSE CALL 'X0X0003' 01
1775.00 C* 
1776.00 CSE EBAE 20000V
1777.00 CSE $TURE IFPHQ 'L'
1778.00 CSE MOVE 'L' #X0009
1779.00 CSE EBAE 8053
1780.00 CSE EBD
1781.00 CSE EBD
1782.00 C* 
1783.00 C* Scrub and edit - Item Category Codes 004
1784.00 C* 
1785.00 CSE MOVEVX004 QRXX04
1786.00 C* 
1787.00 C* Set default value - Item Category Code 004
1788.00 C* 
1789.00 C* 
1790.00 CSE QRXX04 IFPHQ *BLANK
1791.00 CSE DX004 IFPHQ *BLANK
1792.00 CSE MOVEVDX004 #X0
1793.00 CSE MOVEVDX004 QRXX04
1794.00 CSE #XX1,1 IFPHQ ' ' #X0
1795.00 CSE MOVE ' ' #X0
1796.00 CSE #XX2 
1797.00 CSE $M DOUBL20
1798.00 CSE #XX3,1 IFPHQ ' ' #X0
1799.00 CSE MOVE ' ' #X0
1800.00 C* END
1801.00 CSE END
1802.00 CSE END
1803.00 CSE END
1804.00 C* Set allowed values - Item Category Code 004
1805.00 C* 
1806.00 C* 
1807.00 CSE A0X004 IFPHQ *BLANK
1808.00 CSE A0X004 IFPHQ 'NR'
1809.00 CSE QRXX04 #X0
1810.00 CSE MOVE '1' #X0
1811.00 CSE SETON 5193
1812.00 CSE ELSE
1813.00 CSE MOVEVDX004 #X0
1814.00 CSE MOVE 'XITAL' #X0
1815.00 CSE ELSE C997
1816.00 C* 
1817.00 CSE MOVE ' ' $ERTST7
1818.00 CSE MOVE 'BLANK' QRXX40 10
1819.00 CSE MOVEVDX004 QRXX40
1820.00 CSE #XX1,1 IFPHQ 'XITAL'
1821.00 CSE $XRX03 LCEX04XV 01
1822.00 CSE +IS1,1 IFPHQ ' ' 
1823.00 CSE MOVE '1' $ERTST7
1824.00 CSE END
1825.00 CSE END
1826.00 CSE SETON 5193
1827.00 CSE END
1828.00 CSE END
1829.00 CSE END
1830.00 C* 
1831.00 C* Edit upper and lower range - Item Category Code 004
1832.00 C* 
1833.00 C* 
1834.00 CSE L0X004 IFPHQ *BLANK
1835.00 CSE MOVE '1' $ERTST7
1836.00 CSE QRXX04 IFPHQ L0X004
Appendix D - Source Listings

1781.00  CXX  QXX004  AHEADQXX004
1782.00  CXX  MOV R'  \$F1ST
1783.00  CXX  END
1784.00  CXX  \$F1ST5  IPDG '1'
1785.00  CXX  MOV R '3'  \$MK,07
1786.00  CXX  SETON  5193
1787.00  CXX  END
1788.00  CXX  END
1789.00  C*  Edit from User Defined Codes - Item Category Code 004
1790.00  C*  Edit from User Defined Codes - Item Category Code 004
1791.00  C*
1792.00  CXX  RA004  IDMB  $ELINK
1793.00  CXX  CLZMARIC005U
1794.00  CXX  MOVRELQXX004  \$MK
1795.00  CXX  MOV R RA004  \$MK
1796.00  CXX  MOV R QXX004  \$MK
1797.00  CXX  CALL 'X0005'
1798.00  C*  --------
1799.00  CXX  \$F1ST  IO0005U
1800.00  CXX  \$F1ST  IPDG '1'
1801.00  CXX  MOV R '3'  \$MK,09
1802.00  CXX  SETON  5193
1803.00  CXX  END
1804.00  CXX  END
1805.00  C*  --------
1806.00  C*  scrub and edit - Item Category Code 005
1807.00  C*  scrub and edit - Item Category Code 005
1808.00  C*  MOVELQXX005  QXX005
1809.00  C*  Set default value - Item Category Code 005
1810.00  C*  Set default value - Item Category Code 005
1811.00  C*  QXX005  IDMB  $ELINK
1812.00  C*  MOVRELQXX005  \$MK
1813.00  C*  MOV R QXX005  \$MK
1814.00  C*  MOV R QXX005  \$MK
1815.00  C*  MOV R QXX005  \$MK
1816.00  C*  MOV R QXX005  \$MK
1817.00  C*  MOV R QXX005  \$MK
1818.00  C*  MOV R QXX005  \$MK
1819.00  C*  MOV R QXX005  \$MK
1820.00  C*  MOV R QXX005  \$MK
1821.00  C*  MOV R QXX005  \$MK
1822.00  C*  MOV R QXX005  \$MK
1823.00  C*  MOV R QXX005  \$MK
1824.00  C*  MOV R QXX005  \$MK
1825.00  C*  MOV R QXX005  \$MK
1826.00  C*  MOV R QXX005  \$MK
1827.00  C*  MOV R QXX005  \$MK
1828.00  C*  MOV R QXX005  \$MK
1829.00  C*  MOV R QXX005  \$MK
1830.00  C*  --------
1831.00  C*  Edit allowed values - Item Category Code 005
1832.00  C*  Edit allowed values - Item Category Code 005
1833.00  C*  IDMB  $ELINK
1834.00  C*  IDMB  $ELINK
1835.00  C*  IDMB  $ELINK
1836.00  C*  MOV R '1'  \$MK,03
1837.00  CXX  SETON  5193
1838.00  CXX  \$ELINK
1839.00  CXX  MOVRELQXX005  \$MK
1840.00  CXX  MOV R QXX005  \$MK
1841.00  CXX  \$ELINK
1842.00  C*  --------
1843.00  C*  MOV R '1'  \$F1ST  2
1844.00  CXX  MOV R $ELINK  QXXR10 X0
1845.00  CXX  MOV R QXX005  QXXR10
1846.00  CXX  MOV R QXX005  QXXR10
1847.00  CXX  MOV R QXX005  QXXR10
1848.00  CXX  *IN81  IPDG '0'
1849.00  CXX  MOV R '1'  \$F1ST
1850.00  CXX  END
1851.00  CXX  \$F1ST  IPDG '1'
1852.00  CXX  MOV R '1'  \$MK,07
1853.00  CXX  SETON  5193
1854.00  CXX  END
1855.00  CXX  END
1856.00  CXX  END
1857.00  CXX  END
Appendix D - Source Listings

1050.00 C* Edit upper and lower range - Item Category Code 005
1055.00 C* 1060.00 C*
1061.00 CER LAX005 IPHE 'BLANK'
1062.00 CER MOVE 'L' EQRTST
1063.00 CER QXM005 IPHE LAX005
1064.00 CER QXM005 ANELIVS2005
1065.00 CER MOVE ' ' EQRTST
1066.00 CER EN EN
1067.00 CER END IPEQ 'L'
1068.00 CER MOVE 'L' AMK,07
1069.00 CER ENP EN
1070.00 CER EN EN
1071.00 C* Edit from user defined codes - Item Category code 005
1072.00 C*
1073.00 CER MAX005 IPHE 'BLANK'
1074.00 CER CLEARX0005U
1075.00 CER MOVEXD005 $CBE
1076.00 CER MOVEXR005 X0005
1077.00 CER MOVE QXM005 $CBE
1078.00 CER OXLI 'X0005' 01
1079.00 C* -------
1080.00 CER IPEQ 'L'
1081.00 CER MOVE 'L' AMK,07
1082.00 CER EN EN
1083.00 CER EN EN
1084.00 CER EN EN
1085.00 C* ----------------------------------------------------------------------------------------------
1086.00 CER KNO055 KNOE
1087.00 C**************************************************************************************************************************************************
1088.00 C* Copy Common Subroutine - Currency - Translate Video Fields to Data Base
1089.00 C*----------------------------------------------------------------*************************************************************************
1090.00 C* C/OPT SDECYP,CD2121
1091.00 C**************************************************************************************************************************************************
1092.00 C* Copy Common Subroutine - Build Allowed Values Work Array
1093.00 C*----------------------------------------------------------------*************************************************************************
1094.00 C* C/OPT SDECYP,C097
1095.00 C**************************************************************************************************************************************************
1096.00 C* Subroutine 2016 - Update Data Base
1097.00 C* ----------------------------------------------------------------*************************************************************************
1098.00 C* Processing: 1. Update data base file based upon valid action codes.
1099.00 C* ----------------------------------------------------------------*************************************************************************
1100.00 C* 5020.00 C* $CBE ENCBE
1101.00 C* ---- ----
1102.00 C* 1030.00 C* IF add action, add record.
1103.00 C* 1104.00 C* IN21 IPEQ 'L'
1105.00 C* 1106.00 C* W31KIK2001 99
1107.00 C* 1108.00 C* W31KIK2001 99
1109.00 C* 1110.00 C* If change action, update record.
1111.00 C* 1112.00 C* IN21 IPEQ 'L'
1113.00 C* 1114.00 C* WEAT22001 99
1115.00 C* 1116.00 C* WEAT22001 99
1117.00 C* 1118.00 C* If delete action, delete record.
1119.00 C* 1120.00 C* IN22 IPEQ 'L'
1121.00 C* 1122.00 C* IXLETT22001 99
1123.00 C* 1124.00 C* IXLETT22001 99
1125.00 C* 1126.00 C* IXLETT22001 99
1127.00 C* 1128.00 C*
Appendix D - Source Listings

1529.00  C*  Clear data field for next transaction  Forces clear of everything before processing next record. Simulates user pressing the Clear Screen function key.
1530.00  C*  CCR  MOVE $PCKR $SHRDL
1532.00  CCR  ERR$H  $SHDL
1533.00  CCR  END010  END01

1535.00  C* ******************************************************************************
1536.00  C*  SUBROUTINE GCD - Load Dictionary parameters.  Retrieves all of the Data
1537.00  C*  ------------------------------------------------------------------------
1538.00  C*  Dictionary editing parameters for necessary data items used
1539.00  C*  in the program and moves the information into constant fields
1541.00  C*
1542.00  C*
1543.00  C*
1544.00  C*  Dictionary parameters for - Cost Center
1545.00  C*
1546.00  CCR  MOVE 'BLANK' PREZLI
1547.00  CCR  MOVE 'LOC' PREZLI
1548.00  CCR  CALL '299992'  Data Dictionary file server
1549.00  C* ******************************************************************************
1550.00  C*  ERROR  IERROR
1551.00  CCR  ERR01  IPREQ '0'
1552.00  CCR  MOVE PREDOR D0XCC 40
1553.00  CCR  MOVE PREDCT D0XCC 1
1554.00  CCR  MOVE PERC D0XCC 1
1555.00  CCR  MOVE PREFAS D0XCC 50
1556.00  CCR  MOVE PREPAC D0XCC 20
1557.00  CCR  MOVE PRODEC D0XCC 1
1558.00  CCR  MOVE PROFAS D0XCC 4
1559.00  CCR  MOVE PROFAC D0XCC 2
1560.00  CCR  MOVE PROFAL D0XCC 40
1561.00  CCR  MOVE PROFAL D0XCC 40
1562.00  CCR  MOVE PROFAL D0XCC 30
1563.00  CCR  MOVE PROFAL D0XCC 1
1564.00  CCR  MOVE PROFAL D0XCC 20
1565.00  CCR  E-ADDL #4XCC 110
1566.00  CCR  MOVE PREDC $A
1567.00  CCR  DO $A
1568.00  CCR  MULT 10 #4XCC
1569.00  CCR  END
1570.00  CCR  END

1571.00  C*
1572.00  C*
1573.00  C* ******************************************************************************
1574.00  C*  Dictionary parameters for - Description
1575.00  C*
1576.00  CCR  MOVE 'BLANK' PREZLI
1577.00  CCR  MOVE 'IDS' PREZLI
1578.00  CCR  CALL '299992'  Data Dictionary file server
1579.00  C* ******************************************************************************
1580.00  C*  ERROR  IERROR
1581.00  CCR  ERR01  IPREQ '0'
1582.00  CCR  MOVE PREDOR D0XDS 40
1583.00  CCR  MOVE PREDCT D0XDS 1
1584.00  CCR  MOVE PERC D0XDS 1
1585.00  CCR  MOVE PROFAS D0XDS 50
1586.00  CCR  MOVE PROFAC D0XDS 40
1587.00  CCR  MOVE PRODEC D0XDS 1
1588.00  CCR  MOVE PROFAL D0XDS 20
1589.00  CCR  MOVE PROFAS D0XDS 4
1590.00  CCR  MOVE PROFAL D0XDS 20
1591.00  CCR  MOVE PROFAL D0XDS 40
1592.00  CCR  MOVE PROFAL D0XDS 40
1593.00  CCR  MOVE PROFAL D0XDS 50
1594.00  CCR  MOVE PROFAL D0XDS 20
1595.00  CCR  MOVE PROFAL D0XDS 30
1596.00  CCR  MOVE PROFAL D0XDS 40
1597.00  CCR  E-ADDL #5XDS 110
1598.00  CCR  MOVE PREDC $A
1599.00  CCR  DO $A
1600.00  CCR  MULT 10 #5XDS
1601.00  CCR  END
1602.00  CCR  END
1603.00  CCR  END
1604.00  C* ******************************************************************************
1605.00  C*
### Appendix D - Source Listings

#### Dictionary parameters for - Date Last Shipped

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2006.00</td>
<td>C*</td>
</tr>
<tr>
<td>2007.00</td>
<td>C*</td>
</tr>
<tr>
<td>2008.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2009.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2010.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2011.00</td>
<td>C*</td>
</tr>
<tr>
<td>2012.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2013.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2014.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2015.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2016.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2017.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2018.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2019.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2020.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2021.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2022.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2023.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2024.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2025.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2026.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2027.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2028.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2029.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2030.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2031.00</td>
<td>CDR</td>
</tr>
</tbody>
</table>

#### Dictionary parameters for - Item 1D

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2032.00</td>
<td>C*</td>
</tr>
<tr>
<td>2033.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2034.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2035.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2036.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2037.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2038.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2039.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2040.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2041.00</td>
<td>CDR</td>
</tr>
</tbody>
</table>

#### Dictionary parameters for - Quantity On Hand

<table>
<thead>
<tr>
<th>Address</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>2042.00</td>
<td>C*</td>
</tr>
<tr>
<td>2043.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2044.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2045.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2046.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2047.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2048.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2049.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2050.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2051.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2052.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2053.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2054.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2055.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2056.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2057.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2058.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2059.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2060.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2061.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2062.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2063.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2064.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2065.00</td>
<td>CDR</td>
</tr>
<tr>
<td>2066.00</td>
<td>C*</td>
</tr>
</tbody>
</table>

---

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-73
Appendix D - Source Listings

2083.00  CSR  MOVZ DRTT  PRCT  3
2084.00  CSR  MOVZ DROVAL  DAQCT  40
2085.00  CSR  MOVZ DROVAL  A0XCT  40
2086.00  CSR  MOVZ DROVAL  L0XCT  40
2087.00  CSR  MOVZ DROVAL  U0XCT  40
2088.00  CSR  MOVZ FRLT  JSQCT  1
2089.00  CSR  MOVZ FRIENTX  JSQCT  10
2090.00  CSR  CALL 'XRMINX'  JSQCT  10
2091.00  CSR  Z-ADDO1  #AQXCT  110
2092.00  CSR  MOVZ DAQCT  RA
2093.00  CSR  DO RA
2094.00  CSR  MUL10  #AQXCT
2095.00  CSR  END
2096.00  CSR  END
2097.00  CSR  ENDDL

2098.00  CSR

2099.00  CSR

2100.00  CSR

2101.00  CSR  MOVE 'BLANK'  FRETAL
2102.00  CSR  MOVE 'XYT'  FRETAL
2103.00  CSR  CALL 'XRMINX'  01
2104.00  CSR  CALL 'FARM'  10000
2105.00  CSR  PRED 'IPKQ  '0'
2106.00  CSR  MOVE PRODOR  DAQTY  40
2107.00  CSR  MOVE PRODOR  JSQTY  1
2108.00  CSR  MOVE PRODOR  XSQTY  1
2109.00  CSR  MOVE PRODOR  XSQTY  50
2110.00  CSR  MOVE PRODOR  XSQTY  20
2111.00  CSR  MOVE PRODOR  XSQTY  1
2112.00  CSR  MOVE PRODOR  XSQTY  4
2113.00  CSR  MOVE PRODOR  XSQTY  2
2114.00  CSR  MOVE PRODOR  XSQTY  40
2115.00  CSR  MOVE PRODOR  XSQTY  1
2116.00  CSR  MOVE PRODOR  XSQTY  40
2117.00  CSR  MOVE PRODOR  XSQTY  1
2118.00  CSR  MOVE PRODOR  XSQTY  130
2119.00  CSR  MOVE PRODOR  XSQTY  2
2120.00  CSR  DO RA
2121.00  CSR  MUL10  #AQXCT
2122.00  CSR  END
2123.00  CSR  ENDDL

---

2132.00  CSR

2133.00  CSR

2134.00  CSR

2135.00  CSR

2136.00  CSR

2137.00  CSR

2138.00  CSR

2139.00  CSR

2140.00  CSR

2141.00  CSR

2142.00  CSR

2143.00  CSR

2144.00  CSR

2145.00  CSR

2146.00  CSR

2147.00  CSR

2148.00  CSR

2149.00  CSR

2150.00  CSR

2151.00  CSR

2152.00  CSR

2153.00  CSR

2154.00  CSR

2155.00  CSR

2156.00  CSR

2157.00  CSR

2158.00  CSR

2159.00  CSR

---

9-74  JD Edwards World
<table>
<thead>
<tr>
<th>Line</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>2180.00</td>
<td>C*</td>
</tr>
<tr>
<td>2181.00</td>
<td>C*</td>
</tr>
<tr>
<td>2182.00</td>
<td>C*</td>
</tr>
<tr>
<td>2183.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2184.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2185.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2186.00</td>
<td>C*</td>
</tr>
<tr>
<td>2187.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2188.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2189.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2190.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2191.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2192.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2193.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2194.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2195.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2196.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2197.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2198.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2199.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2200.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2201.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2202.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2203.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2204.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2205.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2206.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2207.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2208.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2209.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2210.00</td>
<td>C*</td>
</tr>
<tr>
<td>2211.00</td>
<td>C*</td>
</tr>
<tr>
<td>2212.00</td>
<td>C*</td>
</tr>
<tr>
<td>2213.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2214.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2215.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2216.00</td>
<td>C*</td>
</tr>
<tr>
<td>2217.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2218.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2219.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2220.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2221.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2222.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2223.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2224.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2225.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2226.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2227.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2228.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2229.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2230.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2231.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2232.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2233.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2234.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2235.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2236.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2237.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2238.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2239.00</td>
<td>C*</td>
</tr>
<tr>
<td>2240.00</td>
<td>C*</td>
</tr>
<tr>
<td>2241.00</td>
<td>C*</td>
</tr>
<tr>
<td>2242.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2243.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2244.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2245.00</td>
<td>C*</td>
</tr>
<tr>
<td>2246.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2247.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2248.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2249.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2250.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2251.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2252.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2253.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2254.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2255.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2256.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2257.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2258.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2259.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2260.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2261.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2262.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2263.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2264.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2265.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2266.00</td>
<td>CBR</td>
</tr>
<tr>
<td>2267.00</td>
<td>CBR</td>
</tr>
</tbody>
</table>
Appendix D - Source Listings

2277.00  CSR  MOVEFLRY  D$x0003  4
2278.00  CSR  MOVE PA  R$x0003  2
2279.00  CSR  MOVE PRED  D$x0003  40
2280.00  CSR  MOVE PREV  A$x0005  40
2281.00  CSR  MOVE PRJ  L$x0005  40
2282.00  CSR  MOVE PRED  U$x0003  40
2283.00  CSR  MOVE PREDM  M$x0005  20
2284.00  CSR  MOVE FARA  J$x0005  1
2285.00  CSR  MOVE PHNIX  N$x0003  20
2286.00  CSR  Z-ADD1  H$x0003  110
2287.00  CSR  MOVE PA0003  H
2288.00  CSR  DO  $A
2289.00  CSR  MULT 10  H$x0003
2290.00  CSR  ENO
2291.00  CSR  ENU

2292.00  C*-------------------------------------------------------------
2293.00  C*  Dictionary parameters for - Item Category Code 064
2294.00  C*-------------------------------------------------------------
2295.00  C*  MOVE 'BLANK'  PRDR1
2296.00  C*  MOVE 'EXIT'  PRDR1
2297.00  C*  CALL 'X90002'  01
2298.00  C*-------------------------------------------------------------
2299.00  C*  $EM  I90002
2300.00  C*  FENR  TP01 '0'
2301.00  C*  MOVE PRED  D$x0004  40
2302.00  C*  MOVE PRED  T$x0004  1
2303.00  C*  MOVE PRED  R$x0004  1
2304.00  C*  MOVE PRDCA  C$x0004  50
2305.00  C*  MOVE PRDCA  G$x0004  20
2306.00  C*  MOVE PRDCA  B$x0004  1
2307.00  C*  MOVE FARA  S$x0004  4
2308.00  C*  MOVE PA  R$x0004  2
2309.00  C*  MOVE PRED  D$x0004  40
2310.00  C*  MOVE PRED  M$x0004  40
2311.00  C*  MOVE PRED  L$x0004  40
2312.00  C*  MOVE PRED  G$x0004  30
2313.00  C*  MOVE FARA  J$x0004  1
2314.00  C*  MOVE PHNIX  N$x0004  20
2315.00  C*  Z-ADD1  H$x0004  110
2316.00  C*  MOVE PA0004  H
2317.00  C*  DO  $A
2318.00  C*  MULT 10  H$x0004
2319.00  C*  ENO
2320.00  C*  ENU

2321.00  C*-------------------------------------------------------------
2322.00  C*  Dictionary parameters for - Item Category Code 066
2323.00  C*-------------------------------------------------------------
2324.00  C*  MOVE 'BLANK'  PRDR1
2325.00  C*  MOVE 'EXIT'  PRDR1
2326.00  C*  CALL 'X90002'  01
2327.00  C*-------------------------------------------------------------
2328.00  C*  $EM  I90002
2329.00  C*  FENR  TP01 '0'
2330.00  C*  MOVE PRED  B$x0005  40
2331.00  C*  MOVE PRED  T$x0005  1
2332.00  C*  MOVE PRED  C$x0005  1
2333.00  C*  MOVE PRED  G$x0005  50
2334.00  C*  MOVE PRED  G$x0005  20
2335.00  C*  MOVE PRED  C$x0005  4
2336.00  C*  MOVE PRED  R$x0005  2
2337.00  C*  MOVE PRED  D$x0005  40
2338.00  C*  MOVE PRED  A$x0005  40
2339.00  C*  MOVE PRED  L$x0005  40
2340.00  C*  MOVE PRED  O$x0005  40
2341.00  C*  MOVE PRED  M$x0005  20
2342.00  C*  MOVE FARA  J$x0005  1
2343.00  C*  MOVE PHNIX  N$x0005  20
2344.00  C*  Z-ADD1  H$x0005  110
2345.00  C*  MOVE PA0005  H
2346.00  C*  DO  $A
2347.00  C*  MULT 10  H$x0005
2348.00  C*  ENO
2349.00  C*  ENU
Appendix D - Source Listings

2396.00 C* Load error messages array.
2397.00 C*                                      Error message numbers from Data Dictionary
2398.00 CBR MOVE '0001' ENK,61 Inv Action
2399.00 CBR MOVE '0002' ENK,62 Inv Key
2400.00 CBR MOVE '0003' ENK,63 Inv Blanks
2401.00 CBR MOVE '0004' ENK,64 Inv Date
2402.00 CBR MOVE '0005' ENK,65 Inv Next #br
2403.00 CBR MOVE '0006' ENK,66 Inv Use
2404.00 CBR MOVE '0025' ENK,67 Inv Values
2405.00 CBR MOVE '0026' ENK,68 Inv MCU
2406.00 CBR MOVE '0027' ENK,69 Inv Desc Ttl
2407.00 CBR MOVE '0082' ENK,10

2399.00 C* Load invalid action code array.
2400.00 C* Lockout action code function used with the Program Generator
2401.00 C*---------------------------------------------
2402.00 C*---------------------------------------------
2403.00 C*---------------------------------------------
2404.00 C*---------------------------------------------
2405.00 C* Load system date.
2406.00 C*---------------------------------------------
2407.00 CBR MOVE SWRL12 $RD7 120
2408.00 CBR MOVE $RD7 $SIDAT 60
2409.00 CBR MOVE $RD7 $SIDAT 6
2410.00 CBR MOVE 'JUL' $DFMT 7
2411.00 CBR MOVE 'SEP' $DFMT 7
2412.00 CBR MOVE 'SEP' $DFMT 1
2413.00 CBR CALL 'X0028'
2414.00 C*                                        Feature to allow for all date formats
2415.00 CBR PARM $SIDAT
2416.00 CBR PARM $SIDAT
2417.00 CBR PARM $DFMT
2418.00 CBR PARM $DFMT
2419.00 CBR PARM $DFMT
2420.00 CBR PARM $DFMT
2421.00 CBR PARM $DFMT
2422.00 CBR PARM $DFMT
2423.00 CBR MOVE $SIDAT $UPNJ 60
2424.00 C*---------------------------------------------
2425.00 CBR ENDR
2426.00 C*---------------------------------------------
2427.00 C* Method of releasing master file record locks
2428.00 CT92861 E UNLOCK

9-78 JD Edwards World
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:

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>S00EX</strong></td>
<td>Processes all function exits. Choose a function exit and the system launches one of the following programs:</td>
</tr>
<tr>
<td></td>
<td>▪ Display Functions (F24) to launch P9601H</td>
</tr>
<tr>
<td></td>
<td>▪ Field Sensitive Help (F1) launches X96CCX. After X96CCX launches, the system launches subroutine S00VL.</td>
</tr>
<tr>
<td></td>
<td>▪ Display Error Message (F7) launches P0000E</td>
</tr>
<tr>
<td></td>
<td>▪ HELP launches P00HELP</td>
</tr>
<tr>
<td></td>
<td>▪ Clear Screen (F22) launches subroutine S001</td>
</tr>
<tr>
<td></td>
<td>▪ Launches all programs to process all user defined function keys</td>
</tr>
<tr>
<td><strong>S00VL</strong></td>
<td>Retrieves values with Field Level Help. After X96CCX launches, the system launches subroutine S00VL.</td>
</tr>
<tr>
<td><strong>S00OP</strong></td>
<td>Subfile Options.</td>
</tr>
<tr>
<td><strong>S001</strong></td>
<td>Clears all database and screen fields. This usually only clears key fields and VC0 fields if you choose Clear Screen (F22).</td>
</tr>
<tr>
<td><strong>S002</strong></td>
<td>Checks for level breaks for reports.</td>
</tr>
<tr>
<td></td>
<td>▪ Activates level break markers.</td>
</tr>
<tr>
<td></td>
<td>▪ Retrieves the total line description</td>
</tr>
<tr>
<td>Subroutine</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| S003       | Validates the key fields. Launched subroutine S998 if the system invokes auto inquire. Sets the file pointer.  
- Performs a SETLL and CHAIN if the program is a single record maintenance program  
- Performs a SETLL for subfile programs  
Launches subroutine S004 to load screen and report fields  
Monitors that subfile records load if this is a subfile  
Loads subfile records that the system does not use with blanks |
| S004       | Display and load the screen or report fields. |
| S005       | Scrubs and edits screen and report fields.  
- 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       | For reports with level breaks it:  
- 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  
Launches subroutine S020 if this is a report with subheadings  
If this is not a report, S010 updates, adds, or deletes records from the database file.  
Deactivates the Clear Screen (F22) function and executes S001 to clear the buffer before reading another record. |
| S020       | Print Report Subheadings. |
| S998       | Loads Data Dictionary values, one time only.  
Retrieves row description for level breaks and subheadings, if applicable. |
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Description</th>
</tr>
</thead>
</table>
| S999       | Housekeeping, one time only.  
  - 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 |
Subfile Program with Options

Mainline
- Set auto inquiry
- Key lists
- Retrieve processing options
- Retrieve vocabulary overrides
- Load error messages
- File opens
- Date retrieval

S999

S003
If auto inquiry

S010
Start at beginning of subfile and read each record.
- Edit the data
- Turn on error indicator
- Update/write records to file if no errors, update subfile

S005
- Validate key fields(s)
- Set file pointer
- Monitor for no subfile records loaded, load remaining subfile records with blanks

S998
- Clear fields
- If auto inquiry

S003
- Clear fields if 'Clear Screen' function key is pressed

S001
- Process function keys
- Process selection exits

S00EX
- Return F1 values to screen fields

S00VL

C0001
- Edit the action code

S000P

Write Screens

Read Screen

Simulates the 'Clear Screen' function key to clear fields

Load subfile records

Appendix E - JD Edwards World Subroutines and Flows

CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-83
Report Program without Subheadings

Mainline:
- S999: Key lists
- S998: Load vocabulary overrides
- S998: File opens
- S998: Print cover page and helps
- S998: Retrieve processing options and level breaks
- S998: Retrieve Data Dictionary editing information
- S998: Retrieve row description for subheadings

Read a Record:
- S010: If level break, print totals
- S010: If level break, clear totals
- S010: If end of file, print grand total
- S010: Print detail
- S010: Add to totals

C0001: Check business unit security

S004: Load report fields

S002: Check for level breaks
- Set level break markers
- Retrieve total line description
Appendix F – Sample Code

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

```
TITLE
FILEDEFN01 001000000000
H* -----------------------------------------------------------------
H* FILEDEFN01 002000000000
H* PROGRAM REVISION LOG
H* --------------------
H* Date Programmer Nature of Revision
H* -------- ---------- ------------------------------------
AUTHR*
DESC F*
COPY F*

FILES
COPY E*
```
<table>
<thead>
<tr>
<th>VTS</th>
<th>IDSTXT</th>
<th>DS</th>
</tr>
</thead>
<tbody>
<tr>
<td>VTX</td>
<td>I</td>
<td>1 40 VTX001</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>41 80 VTX002</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>81 120 VTX003</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>121 160 VTX004</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>161 200 VTX005</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>201 240 VTX006</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>241 280 VTX007</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>281 320 VTX008</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>321 360 VTX009</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>361 400 VTX010</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>401 440 VTX011</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>441 480 VTX012</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>481 520 VTX013</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>521 560 VTX014</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>561 600 VTX015</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>601 640 VTX016</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>641 680 VTX017</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>681 720 VTX018</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>721 760 VTX019</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>761 800 VTX020</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>801 840 VTX021</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>841 880 VTX022</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>881 920 VTX023</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>921 960 VTX024</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>961 1000 VTX025</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1001 1040 VTX026</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1041 1080 VTX027</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1081 1120 VTX028</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1121 1160 VTX029</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1161 1200 VTX030</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1201 1240 VTX031</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1241 1280 VTX032</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1281 1320 VTX033</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1321 1360 VTX034</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1361 1400 VTX035</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1401 1440 VTX036</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1441 1480 VTX037</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1481 1520 VTX038</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1521 1560 VTX039</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1561 1600 VTX040</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1601 1640 VTX041</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1641 1680 VTX042</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1681 1720 VTX043</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1721 1760 VTX044</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1761 1800 VTX045</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1801 1840 VTX046</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1841 1880 VTX047</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1881 1920 VTX048</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1921 1960 VTX049</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>1961 2000 VTX050</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2001 2040 VTX051</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2041 2080 VTX052</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2081 2120 VTX053</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2121 2160 VTX054</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2161 2200 VTX055</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2201 2240 VTX056</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2241 2280 VTX057</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2281 2320 VTX058</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2321 2360 VTX059</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2361 2400 VTX060</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2401 2440 VTX061</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2441 2480 VTX062</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2481 2520 VTX063</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2521 2560 VTX064</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2561 2600 VTX065</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2601 2640 VTX066</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2641 2680 VTX067</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2681 2720 VTX068</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2721 2760 VTX069</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2761 2800 VTX070</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2801 2840 VTX071</td>
</tr>
<tr>
<td>VTX</td>
<td>I</td>
<td>2841 2880 VTX072</td>
</tr>
</tbody>
</table>

**B010**

- **STD/M - Action Code**
- **DATE - 2/02/17**
CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-89
Appendix F – Sample Code

/* MAINLINE 083000000000*/

+FLDNC* #SFRHD 2SFLRCCOD
+FLDNC* If SFLCLR is used, process 'in38 accordingly
+FLDNC* ?SFL CLR 2SFLCLR
C* Write video screen.
C* If not a subfile display just write formatl
+FLDNC* 7SFL 2WRITE

R09999 080000000000 BSO15 - STD/M - Action Code
+FLDNC* If a subfile display, write formatl and formatf
+FLDNC* 7SFL 2WRITE
C* Move '1'
C* EXSR S051
C* ---- ----
C* Load data field dictionary parameters (one cycle only).
C* $998 CASE$ ' / $998
C* ---- ----
C* END
C* Begin video screen read processing.
C* SETP 999301
DBFP C READ 401FIER 998
C Z-ADD0 ##RR0W
C Z-ADD0 ##RCOL
C* If video read timed out, end program.
C* '+'IN99 CAREQ'1' EOJ LR
C* ---- ---- ---
C* ?AID CAREQ#FD0J EOJ LR
C* ---- ---- ---
C* If valid function key pressed, process and return.
C* '+'IN15 IPEQ '1'
C* EXSR S050E
C* ---- ----
C* '+'INLR CAREQ'1' EOJ
C* ---- ---- ---
C* '+'IN15 CAREQ'1' END
C* ---- ---- ---
C* END
*/

/* If any selection exits, exsr S000P*/

+DTAC* SELC 2S000P
+FLDNC* If action code then exsr C0001
+FLDNC* ACTION ZACTION
C* Load subfile records.
C* EXSR S053
C* ---- ----
/* If any update files then exsr S005*/

+FILE* "ANY DB 2S005 0"
*/

/* If any update files and action code then do S010*/

+FILE* "ANY DB 'AMD' 0"
+FILE* "ANY DB 'AMD' 2"
R09999 080000000000 BSO15 - STD/M - Action Code
+FLDNC* If a Master File 2 exists, then do S011.
+FILE* "ANY DB 'AMD' 0"
+FILE* "ANY DB 'AMD' 2"
+FLDNC* ACTION 2SS011
C* Return for next input.
C* END TAG
C* ---- ----
C* Set correct message in line 24.
C* '+'IN93 IPEQ '1'
C MOVELsville24E VDL24
C ELSE
C MOVELsville24M VDL24
C END
COPY*******************************************************************************
C* S00EX-1 001000000000
C* SUBROUTINE S00EX - Process Function Keys S00EX-1 002000000000
C* ---------------------------------------- S00EX-1 003000000000
C* S00EX-1 004000000000
C* Processing: 1. Determine function key pressed. S00EX-1 005000000000
C* 2. Process function key request. S00EX-1 006000000000
C* S00EX-1 007000000000
CSR S00EX BEGSR S00EX-1 008000000000
C* ----- ----- S00EX-1 009000000000
+FLDNC* #SFRNO Z@@SRCN S00EX-1 009500000000
CSR T00EXA TAG S00EX-1 010000000000
C* ------ --- S00EX-1 011000000000
C* S00EX-1 012000000000
C* If EOJ requested, exit subroutine. S00EX-1 013000000000
C* S00EX-1 014000000000
CSR @@AID CABEQ#FEOJ ENDEXE LR S00EX-1 015000000000
C* ----- ------ S00EX-1 016000000000
C* S00EX-1 017000000000
C* If Display Keys pressed, exit to help facility and return. S00EX-1 018000000000
C* S00EX-1 019000000000
CSR @@AID CABCQ#FEOJ ENDEXE LR S00EX-1 020000000000
C* ----- ------ S00EX-1 021000000000
#93950 B0010 - STD/M - Action Code DATE - 2/02/17
CSR goto ENDEXE S00EX-1 030000000000
C* ---- ------ S00EX-1 031000000000
CSR END S00EX-1 032000000000
C* S00EX-1 033000000000
C* If Cursor Sensitive Help Pressed, exit to CS Help. S00EX-1 034000000000
C* S00EX-1 035000000000
CSR @@AID IFEQ #FQMRK S00EX-1 036000000000
CSR MOVEA*IN ##IN S00EX-1 037000000000
CSR CALL 'X96CCX' 98 S00EX-1 038000000000
C* ---- -------- S00EX-1 039000000000
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
CSR ##FLDN IFNE *BLANKS S00EX-1 045000000000
CSR EXSR S00VL S00EX-1 046000000000
CSR END S00EX-1 047000000000
CSR MOVEL*BLANKS ##DTAI S00EX-1 048000000000
CSR GOTO ENDEXE S00EX-1 049000000000
C* ---- ------ S00EX-1 050000000000
CSR END S00EX-1 051000000000
C* S00EX-1 052000000000
C* If Display errors pressed, exit to error messages. S00EX-1 053000000000
C* S00EX-1 054000000000
CSR @@AID IFEQ #FERRD S00EX-1 055000000000
CSR Z-ADD1 #G S00EX-1 056000000000
CSR Z-ADD1 #H S00EX-1 057000000000
CSR #G DOWLE64 S00EX-1 058000000000
CSR @MK,#G IFEQ '1' S00EX-1 059000000000
CSR MOVE EMK,#G @ER,#H S00EX-1 060000000000
CSR ADD 1 #H S00EX-1 061000000000
CSR END S00EX-1 062000000000
CSR CALL 'P0000E' 98 S00EX-1 063000000000
CSR PARM @ER S00EX-1 064000000000
CSR GOTO ENDEXE S00EX-1 065000000000
C* ---- ------ S00EX-1 066000000000
CSR END S00EX-1 067000000000
C* S00EX-1 068000000000
C* If HELP key pressed, exit to help facility and return. S00EX-1 069000000000
C* S00EX-1 070000000000
CSR @@AID IFEQ #FHELP S00EX-1 071000000000
CSR CALL 'P00HELP' 98 S00EX-1 072000000000
CSR PARM HS@@ S00EX-1 073000000000
CSR PARM HE@@ S00EX-1 074000000000
 CSR PARM H@@ S00EX-1 075000000000
 CSR PARM H@@ S00EX-1 076000000000
 CSR PARM H@@ S00EX-1 077000000000
 CSR PARM H@@ S00EX-1 078000000000
 CSR PARM H@@ S00EX-1 079000000000
 CSR PARM H@@ S00EX-1 080000000000
 CSR PARM H@@ S00EX-1 081000000000
 CSR PARM H@@ S00EX-1 082000000000
 CSR PARM H@@ S00EX-1 083000000000

Appendix F – Sample Code

CSR       PARM  IOOSC  S00EX-1  084000000000
CSR       PARM  SRVPDS S00EX-1  085000000000
CSR       PARM  IOOCR S00EX-1  086000000000
CSR       GOTO ENDEXE  S00EX-1  087000000000
R039560  B0101 - STD/M - Action Code
C*       ----       ----
CSR       END  S00EX-1  088000000000
C*       If Clear screen pressed, clear screen and return.
C*       ---------------------------------------------
CSR       @@AID IFEQ #FCLR S00EX-1  090000000000
CSR       EXSR S001 S00EX-1  091000000000
C*       ---- ----
CSR       GOTO ENDEXE  S00EX-1  093000000000
C*       ---- ----
CSR       EXITCCSR END S00EX-1  096000000000
C*       S00EX-1  099000000000
C*       Process roll up and down keys.
C*       ----------------------------------
CSR       @@AID IFEQ #FROLU S00EX-1  101000000000
CSR       @@AID OREQ #FROLD S00EX-1  102000000000
CSR       $SECUR DOUEQ' ' S00EX-1  104000000000
CSR       MOVE ' ' $SECUR 1 S00EX-1  106000000000
C*       S00EX-1  109000000000
C*       If ROLL UP key pressed, process read next.
C*       -------------------------------
CSR       @@AID IFEQ #FROLU S00EX-1  111000000000
C*       S00EX-1  113000000000
C*       Reset error indicators if roll
C*       S00EX-1  115000000000
CSR       MOVEA$RESET *IN,41 S00EX-1  117000000000
CSR       MOVE '0' *IN,40 S00EX-1  119000000000
CSR       SETOF 818299 S00EX-1  121000000000
MF CSR % READ &01FORMAT 9981 S00EX-1  123000000000
CSR       *IN81 IFEQ '1' S00EX-1  125000000000
MF CSR % READP&01FORMAT 9982 S00EX-1  127000000000
C*       S00EX-1  129000000000
C*       If ROLL DOWN key pressed, process read prior.
C*       ------------------------------------------
CSR       @@AID IFEQ #FROLD S00EX-1  131000000000
C*       S00EX-1  133000000000
C*       Reset error indicators if roll
C*       S00EX-1  135000000000
CSR       MOVEA$RESET *IN,41 S00EX-1  137000000000
CSR       MOVE '0' *IN,40 S00EX-1  139000000000
CSR       SETOF 818299 S00EX-1  141000000000
MF CSR % READP&01FORMAT 9981 S00EX-1  143000000000
CSR       *IN81 IFEQ '1' S00EX-1  145000000000
MF CSR % READP&01FORMAT 9982 S00EX-1  147000000000
C*       S00EX-1  149000000000
C*       Load video screen data on roll keys.
C*       ------------------------------
CSR       @@AID IFEQ #FROLU S00EX-1  151000000000
CSR       @@AID OREQ #FROLD S00EX-1  153000000000
/* S00EX-1  155000000000
/* Include record lock logic if update files exist. S00EX-1  157000000000
/* S00EX-1  159000000000
+FILEC* *ANY DB ZUNLOCK @ S00EX-1  161000000000
MCU01C* Cost Center security edit.
MCU01CSR MOVEL&01(FILE )#FILE S00EX-1  165000000000
MCU01C* S00EX-1  167000000000
MCU01C* S00EX-1  169000000000
MCU01C* S00EX-1  171000000000
MCU01CSR MOVEL&01(FILE )#FILE
```c
MOVE '1' $SECUR

CSR BEGSR

##RVAL IFEQ '*BLANK'
MOVE ' ' ##RVAL

CSR #ENDS

END

SUBROUTINE 00V3 - Edit Key

2. Load input keys.
3. Validate master file key.
4. Release master file record lock.

CLR

CSR *NOKEY CLEAR&01FORMAT

COPY

S00V3 E000 - STD/M - Action Code
DATE - 2/02/17

---

END

SUBROUTINE 00V3 - Edit Key

2. Load input keys.
3. Validate master file key.
4. Release master file record lock.

CLR

CSR *NOKEY CLEAR&01FORMAT

COPY

S00V3 E000 - STD/M - Action Code
DATE - 2/02/17

---

END
```
Appendix F – Sample Code

C* 5. Load video screen output on inquiry. S003-1
CSR S003 BEGSR
C*
C*
C* Load data field dictionary parameters (one cycle only).
CSR $998 CASEQ’ ’ S998
C* ----- ----- S003-1
C* S003-1 012100000000
C*
C* Reset error indicators and arrays.
C* CSR MOVE ‘ALL’$RESET 39 S003-1
CSR MOVE ‘BLANK $RESET1 63 S003-1
CSR MOVEAR$RESET ‘IN’,61 S003-1
CSR MOVEAR$RESET1 8MK,2 S003-1
CSR CLEAR@ER S003-1
C*----------------------------------------------------------------
C* S003-1
C* S003-1
C* S003-1
C* S003-1
C* S003-1
CSR END S003-1
C* S003-1
C* Reset error indicators and arrays.
C* CSR MOVE ‘ ‘$SECR 1 S003-1
CSR GOTO END003 S003-1
C* ---- ------ S003-1
CSR END S003-1
C* Edit result of read and action code.
C* CSR *IN98 IFEQ ‘1’ R93950 S003-1
CSR ELSE S003-1
CSR *IN21 COMP ‘1’ 41 ‘error* S003-1
R93950 BDS01S - STD/M - Action Code DATE - 2/02/17
C* End
CSR END S003-1
C* If not inquiry, skip remainder of subroutine.
C* CSR ‘IN24 CASEQ’O’ ENDO03 S003-1
C* Release record lock on master file.
C* CSR ‘IN99 IFEQ ’O’ S003-1
CSR ‘IN99 ANDEQ’O’ S003-1
CSR EXCPUNLOCK S003-1
CSR END S003-1
C* If errors, skip remainder of subroutine.
C* CSR ‘IN93 CASEQ’I’ ENDO03 S003-1
C* Move data base information to video screen.
COPY C****************************************************************
C* S003-1 001000000000
C* SUBROUTINE S003 - Load Video Screen Data
C* -------------------------------------------------------------
CSR EXSR S003-1 002000000000
C* ---- ---- S003-1 003000000000
C*---------------------------------------------------------------- S003-1 004000000000
CSR END003 ENDSR S003-1 005000000000
COPY C****************************************************************
C* S004-1 001000000000
C* SUBROUTINE S004 - Load Video Screen Data
C* ----------------------------------------
C* S004-1 004000000000
C* Processing: 1. Move data base information to video screen.
C* All video screen fields are alpha and therefore numeric information must be
C* processed through subroutine C0014 to set proper decimals and provide editing for
C* display on screen.
C* Date fields must be converted from their internal format of month, day and year or
R93950 B0010 - STD/M - Action Code DATE - 2/02/17
C* julian using program X0028.
C*--------------------------------------------------------
CSR S004 BEGSR S004-1 017000000000
C* ---- ----- S004-1 018000000000
DSP1 C* S004-1 025000000000
CSR END004 ENDSR S004-1 026000000000
COPY C****************************************************************
C* S005-1 001000000000
C* SUBROUTINE S005 - Scrub Input
C* ------------------------------
C* S005-1 004000000000
C* Processing: 1. Validate all video input. All numeric fields must be processed
C* thru subroutines C0012 and C0015 in order to scrub the alpha input field and convert
C* back to internal numeric representation of 15 digits and 0 decimals.
C* Date fields must be converted from system format to their internal format of month,
C* day and year or julian using program X0028.
C* 2. Update data record fields from video.
C*--------------------------------------------------------
CSR S005 BEGSR S005-1 010000000000
C* ---- ----- S005-1 011000000000
CSR *IN21 IFEQ '0' S005-1 012000000000
CSR END S005-1 015000000000
COPY C****************************************************************
C* S010-1 001000000000
C* SUBROUTINE S010 - Update Data Base
C* -----------------------------------
C* S010-1 004000000000
C* Processing: 1. Update data base file based upon valid action codes.
C* If not addition or change, bypass subroutine.
C* If add action, add record.
C* If change action, update record.
C* If delete action, delete record.
R93950 B0010 - STD/M - Action Code DATE - 2/02/17
C*--------------------------------------------------------
CSR S010 BEGSR S010-1 001000000000
C* ---- ----- S010-1 002000000000
CSR END S010-1 003000000000
CASE Computer Aided Software Engineering Guide (Revised - May 1, 2009) 9-95
C* SUBROUTINE S998 - Load dictionary parameters.
C* ----------------------------------------------
C* S998 BEGSR
C* ---- -----
DPARM*
C* Set subroutine execution flag.
CSR MOVE '1' $998 1
CSR END998 ENDSR
C*****************************************************************
C* SUBROUTINE S999 - Housekeeping
C* -------------------------------
CSR S999 BEGSR
C* Required program parameters.
ENTRYCSR *ENTRY PLIST
AUTOIC*
C*----------------------------------------------------------------
C* Load video screen text.
CSR MOVEL@@FILE PSKEY 10
VTXI C*COPY JDECPY,C00SC
C*----------------------------------------------------------------
/* If processing options exist, load processing options */
R93950 B0010 - STD/M - Action Code DATE - 2/02/17
+FLDNC* *OPTION ZOPTIONX
KLISTC*
C* Load roll key upper and lower key values.
CSR MOVE @FILE $RUKEY S999-1
CSR MOVEL *LOVAL $RUKEY S999-1
CSR MOVEL *ALL'9' $RDKEY S999-1
C*----------------------------------------------------------------
C* Load error messages array.
CSR MOVE '0001' EMK,01 Inv Action S999-1
CSR MOVE '0002' EMK,02 Inv Key S999-1
CSR MOVE '0003' EMK,03 Inv Blanks S999-1
CSR MOVE '0004' EMK,04 Inv Next Mbr S999-1
CSR MOVE '0005' EMK,05 Inv Next Mbr S999-1
CSR MOVE '0006' EMK,06 In Date S999-1
CSR MOVE '0007' EMK,07 Inv Values S999-1
CSR MOVE '0026' EMK,08 Inv MCU S999-1
CSR MOVE '0027' EMK,09 Inv Dec Ttl S999-1
EMK C*----------------------------------------------------------------
C* Load invalid action code array.
CSR MOVEA' ' @NAC S999-1
ACTN CSR MOVEA' ' @NAC
C*----------------------------------------------------------------
C*******************************************************************************
/*
/*  If processing options exist, include copy module
/*
COPY C*******************************************************************************
MF G401FMT E  UNLOCK

098000000000
S999-1 098000000000
S999-1 099000000000
S999-1 100000000000
S999-1 101000000000
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
Index

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
c Conditional, 8-35
c exception, 8-34
c functional, 8-25
substitution, 8-32
understanding, 8-25
disable data dictionary editing, 3-55
d Display Action Diagram screen (P92705), 7-21, 7-23
d DREAM writer considerations for report formats, 6-14
e E0010 — Interactive Window program, 9-28
e Edit screen, 3-28, 3-30, 8-22, 8-52
e Editing, parsing, and source generation of PDL, 4-17
e Enabling database update function for subfiles, 3-51
e Error handling using arrays, 9-5
e 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
Index

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 Action Request, 2-12
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