Send Us Your Comments


JD Edwards World welcomes your comments and suggestions on the quality and usefulness of this publication. Your input is an important part of the information used for revision.

- Did you find any errors?
- Is the information clearly presented?
- Do you need more information? If so, where?
- Are the examples correct? Do you need more examples?
- What features did you like most about this manual?

If you find any errors or have any other suggestions for improvement, please indicate the title and part number of the documentation and the chapter, section, and page number (if available). You can send comments to us by e-mail at:

jde_world_doc_ww@oracle.com

If you would like a reply, please give your name, address, telephone number, and electronic mail address (optional).

Contact a JD Edwards World representative by calling Oracle Global Support Center at 1-800-289-2999 for current information or if you have any questions regarding this document.
### Work with Software Action Requests

- About SAR System Setup .............................................. 3-39
- Creating Record Type Codes .................................. 3-39
- Defining Record Type Titles ........................................ 3-41

### Work with Software Versions Repository

- Working with Software Versions Repository (SVR) ...... 3-43
- Accessing the Software Versions Repository .......... 3-45
- Member Identifiers ...................................................... 3-45
- Naming Conventions .................................................... 3-52
- The JD Edwards World System Codes ...................... 3-56
- Examples of Program and File Names ....................... 3-61
- Optional Files Workbench .......................................... 3-62
- Navigation Functions .................................................. 3-64

### CASE Profiles

- About CASE Profiles .................................................. 3-71
- Accessing CASE Profiles ............................................ 3-71
- Function Key Exits from the CASE Profiles Program .. 3-75
- Summary of CASE Profiles ........................................ 3-75

### Working with SAR Log

- About SAR Log .......................................................... 3-77
- Setting Up User Input Options for SAR Logging ........ 3-78
- Selecting Types of SAR Information to Log .............. 3-79
- Accessing SAR Log Inquiry ......................................... 3-80
- Summary of the SAR Log Inquiry ............................... 3-83

### Work with Promotion Paths and Projects

- Working with Promotion Paths and Projects ............. 3-85
- Understanding Promotion Paths ................................ 3-85
- Defining a Promotion Path ......................................... 3-87
- Defining a Project ...................................................... 3-92

### Promote a Project

- Promoting a Project .................................................. 3-97
- Update the SARs ........................................................ 3-97
- Validating a Promotion Path ....................................... 3-99
- Promoting a Project .................................................. 3-102

### Promote Project Updates

- Promoting Project Updates ........................................ 3-105
4 Programming Tools

Overview to Programming Tools ................................................................. 4-1
  About Programming Tools ......................................................................... 4-1

Work with Data Modeling ............................................................................ 4-3
  Working with Data Modeling ..................................................................... 4-3
  Accessing Data Modeling ......................................................................... 4-3

Work with the Object Cross Reference Repository .................................. 4-9
  Working with the Object Cross Reference Repository ............................. 4-9

Work with Data Dictionary .......................................................................... 4-13
  About the Data Dictionary Repository ..................................................... 4-13
  Understanding the Data Dictionary Structure .......................................... 4-13
  Locating a Data Item Name ....................................................................... 4-15
  Working with the Data Dictionary ............................................................ 4-16
  Working with Data Item Alias Revisions ................................................ 4-21
  Working with the Data Dictionary Glossary ............................................. 4-22
  Working with User Defined Help Instructions ........................................ 4-26
  Working with Data Field Descriptions ..................................................... 4-27
  Working with the Next Numbers Facility ............................................... 4-27
  About the Field Reference File ............................................................... 4-29
  About the JD Edwards World Message File ........................................... 4-30
  Locating the Rebuild FRF and JD Edwards World Msg File Form ........... 4-31

Work with Data File Design Aid ................................................................. 4-33
  About the Data File Design Aid ............................................................... 4-33
  About Assigning the File Prefix ............................................................... 4-34
  Entering Data File Design Aid ................................................................. 4-35
  Function Keys From File Design Aid ....................................................... 4-38
  What are the Data File Design Aid Standards? ...................................... 4-40
  Data File Design Aid Summary ............................................................. 4-41

Work with Screen Design Aid ..................................................................... 4-43
  About Screen Design Aid ......................................................................... 4-43
  Editing Commands .................................................................................... 4-44
  Prefix Standards ....................................................................................... 4-45
  Field Name Standards ............................................................................. 4-46
  Updating or Adding Fields through SDA ................................................ 4-47
  Working with Screen Design Aid ............................................................ 4-47
  Function Key Exits ................................................................................... 4-48
  Updating an Existing Field ...................................................................... 4-48
  Accessing Fast Path Create for a New Screen ........................................ 4-51
Contents

Adding Fields without Using a Pick List ................................................................. 4-54
Adding a Literal Field............................................................................................ 4-56
Using the *BOTH and *ALL Features.................................................................. 4-56
Compiling Your Screen.......................................................................................... 4-60
Screen Design Standards and Tips........................................................................ 4-60
Adding Screen Fields Using Pick List.................................................................... 4-64
Selecting Database Fields...................................................................................... 4-66
Placing Fields on a Screen Using a Pick List....................................................... 4-70
Function Key Exits from Screen Design Aid ...................................................... 4-70
Changing Subfile Boundaries............................................................................... 4-77
Process Overview - Placing Selected Fields....................................................... 4-78
Process Overview - Revising the Field Definition ............................................. 4-80
Process Overview - Revising Vocabulary and Function Keys............................ 4-81
Function Keys for Screen and Display Format Control....................................... 4-81
Summary of Screen Design Aid........................................................................... 4-82

Work with Report Design Aid............................................................................. 4-83
About Report Design Aid...................................................................................... 4-83
Accessing Report Design Aid.............................................................................. 4-90
Updating a Field in RDA...................................................................................... 4-90
Compiling A Report.............................................................................................. 4-96
Changing the Compile Option Defaults for Reports......................................... 4-97

5 Programming Standards

Overview to Programming Standards................................................................. 5-1
Programming Standards....................................................................................... 5-1

Program Specifications....................................................................................... 5-3
About Program Specifications............................................................................. 5-3
What Are Header (Control) Specifications?..................................................... 5-4
What Are File Description Specifications?....................................................... 5-4
What Are Extension Specifications?................................................................. 5-5
What Are Input Specifications?......................................................................... 5-5
What Are Calculation Specifications?............................................................... 5-6
What Are Output Specifications?...................................................................... 5-7

6 Program Overview

Program Overview.............................................................................................. 6-1
About the Program Overview............................................................................. 6-1
Subroutines.......................................................................................................... 6-1
Error Handling.................................................................................................... 6-3
Indicator Usage................................................................................................... 6-4
Contents

Documentation.................................................................................................................. 6-5
Miscellaneous Items ....................................................................................................... 6-7

Program Structure .......................................................................................................... 6-11
About Program Structure ............................................................................................. 6-11

User Spaces ..................................................................................................................... 6-49
About User Spaces ......................................................................................................... 6-49
What Is a User Space? ........................................................................................................ 6-49
What Are the Advantages of Using a User Space? ....................................................... 6-49
How Does a User Space Function? .............................................................................. 6-50
Creating a User Space ................................................................................................... 6-50
Writing to a User Space ................................................................................................. 6-53
Reading from a User Space .......................................................................................... 6-54

User Indices .................................................................................................................. 6-57
About User Indices ......................................................................................................... 6-57
What Are the Advantages of Using a User Index? ....................................................... 6-58
How Does a User Index Function? ............................................................................... 6-58
Creating a User Index ..................................................................................................... 6-59
Writing to a User Index .................................................................................................. 6-62
Appearance of Records ................................................................................................. 6-63
Retrieving Data from a User Index .................................................................................. 6-64

File Servers ................................................................................................................... 6-87
About File Servers ......................................................................................................... 6-87
What Is a File Server? ...................................................................................................... 6-87
What are the Advantages of Using a File Server? ....................................................... 6-88
What are the Disadvantages of Using a File Server? ............................................... 6-88
How Does a File Server Function? ............................................................................... 6-89
What Are Control Parameters? ..................................................................................... 6-89
What Are Returned Parameters? .................................................................................. 6-92
Implementing a File Server ........................................................................................... 6-92
Searching for Key Lists ................................................................................................. 6-93
Tips when Using File Servers ....................................................................................... 6-94
Commonly Used File Servers ....................................................................................... 6-99

Functional Servers ....................................................................................................... 6-101
About Functional Servers ............................................................................................ 6-101
What Are Functional Servers? ..................................................................................... 6-101
What Are the Advantages of Using a Functional Server? ........................................ 6-102
What Are the Disadvantages of Using a Functional Server? .................................... 6-103
Setting Up Business Rules for an Entry Program ................................................... 6-103
How Does a Functional Server Function? .................................................................... 6-103
# Contents

- Functional Server Highlights ................................................................. 6-104
- The Call Parameters for the Functional Server ....................................... 6-113
- Control Fields within the User Space ......................................................... 6-116
- Error Message Index Line (C00RHX) ......................................................... 6-117
- Example - Functional Server Program Sections ....................................... 6-117
- Available Functional Servers ................................................................. 6-121

**Source Debugger** .................................................................................. 6-123
  - About Source Debugger ........................................................................ 6-123
  - Using Debugger with an Interactive Program ......................................... 6-123
  - Using Debugger with a Batch Program .................................................. 6-126
  - Features of the JD Edwards World Source Debugger ............................. 6-129

**Software Scan and Replace** .............................................................. 6-133
  - About Software Scan and Replace ......................................................... 6-133
  - Report .................................................................................................... 6-134

**Performance Issues** ............................................................................. 6-135
  - DREAM Writer ..................................................................................... 6-135

# 7 Group Jobs

**Overview to Group Jobs** ................................................................. 7-1
  - About Group Jobs ................................................................................ 7-1

**Access the JD Edwards World Group Job window** ............................. 7-3
  - About the JD Edwards World Group Job window ................................ 7-3
  - Accessing the JD Edwards World Group Job window .......................... 7-4
  - Creating New Group Jobs .................................................................... 7-4
  - Activating Suspended Group Jobs......................................................... 7-5
  - Terminating Group Jobs ...................................................................... 7-6
  - Changing to Non-Group Mode ............................................................. 7-6
  - Signing Off with Suspended Group Jobs .............................................. 7-6
  - Work with Non-JD Edwards World Group Jobs ................................... 7-7
  - Advanced Functions of the JD Edwards World Group Job window ........ 7-8

**Work with the Attention MENU window** ............................................ 7-11
  - About the Attention MENU Window .................................................... 7-11
  - Accessing the JD Edwards World Attention Menu window ................. 7-11

**Work with IBM Pass-Through** ......................................................... 7-15
  - About Working with IBM Pass-Through .............................................. 7-15
  - Setting Up Access to Remote Locations .............................................. 7-15
Using IBM Pass-Through with Group Jobs ................................................................. 7-16

8 Universal File Converter

Overview to Universal File Converter ................................................................. 8-1
   About Universal File Converter .................................................................... 8-1

Set Up Universal File Converter ................................................................. 8-7
   About Universal File Converter (UFC) ......................................................... 8-7
   Understanding the Universal File Converter Setup ..................................... 8-7
   Setting Up Universal File Converter .............................................................. 8-9

Work with Crossover Rules ................................................................. 8-15
   Working with the Crossover Rules Screen ..................................................... 8-15
   Displaying Field Descriptions ...................................................................... 8-18
   Adding Fields ................................................................................................. 8-19
   Deleting Records ........................................................................................... 8-20
   Keywords ........................................................................................................ 8-21
   About the Conversion Rule Program .............................................................. 8-22
   Available Functions and Options .................................................................. 8-23

Work with File Conversion ................................................................. 8-25
   Working with File Conversion ....................................................................... 8-25

Print a Report ................................................................................................. 8-29
   Printing a Report ............................................................................................ 8-29

Create Conversion Versions ................................................................. 8-31
   Creating Conversion Versions ....................................................................... 8-31
   Creating Conversion Forms ............................................................................. 8-31

Work with the Data Dictionary Glossary by File ......................................... 8-33
   About Working with the Data Dictionary Glossary by File ......................... 8-33
   Accessing the Data Dictionary Glossary by File ........................................... 8-33
   Adding a File Specific Glossary Item ............................................................. 8-34
   Printing the Data Dictionary Glossary Information ....................................... 8-34

9 Appendices

Appendix A - Common & Production Library Files ....................................... 9-1

Appendix B - Upgrading Customized Source Code ....................................... 9-3
   S/ Compare ..................................................................................................... 9-3
1 Overview
Overview to JD Edwards World

Application Development Cycle

World Computer Aided Software Engineering (CASE) covers the entire spectrum of the application development life cycle, including design tools, code generation, automatic documentation generation, prototyping, repositories and other productivity improvement tools for the development, operation and maintenance of flexible, business application software.

You can describe the Application Development Cycle (A/D Cycle) in three levels, as follows:

Level 1
- The Application Platform, which is described in the Technical Foundation document.

Level 2
- The Design Platform, which is described in the Advanced Programming Concepts and Skills (APCS) document.

Level 3
- The Development Platform, which is described in the CASE document.

Universal Building Blocks of JD Edwards World Software

World CASE covers the entire spectrum of the application development life cycle, including:
- Design tools
- Code generation
- Automatic documentation generation
- Prototyping
- Repositories
- Other productivity improvement tools

The following figure shows the separate modules that contribute to the functioning of a JD Edwards World program.
2 APCS System Overview
Overview to APCS System

Features

Advanced Programming Concepts and Skills (APCS) focuses on the following World CASE features:

- Data Dictionary Repository
- Project Management (Software Action Request System)
- CASE Profiles
- SAR Log Inquiry
- Creating a Development Environment
- Software Versions Repository
- Data Modeling
- File Design Aid
- Screen Design Aid
- Report Design Aid
- JD Edwards World Programming Standards
- File Servers and Functional Servers
- User Spaces and User Indexes
- Group Jobs
- Programming Modifications
- Source Debugger
- Programming Impacts from Software Upgrades

Note: Because this is a programming guide, and the majority of programmers do not use the Java interface, we are using non-Java (green screen) captures in this guide.
3 Version Control
Overview to Version Control

About Version Control

Use the JD Edwards World Version Control system to manage the movement of software between various environments, such as ones you have set up for software development, testing, and production.

The Version Control system works with the Software Action Request (SAR) system and the SAR logging system. It performs three general functions:

- Groups source code members (such as RPG and CL programs, and physical and logical files) and control file data (such as Data Dictionary and menus) together as a project.
- Defines a promotion path, which specifies library information about the project’s current environment and the environment to which it will be moved.
- Promotes the project from the current environment to the target environment as defined by the promotion path.
To set up a software development project for development and promotion, you must:

- Create the SARs that you want to promote, and define promotion paths.
- Link the project to the SARs that are associated with it, and assign a promotion path to it.

All additions or changes you make to programs and control file data are logged in the SAR Log (F9810). Use this log to update the SARs, which are in the Work Order Detail table (F4802).

After you finish developing the software, you promote the software from the Project Elements form to the next environment.

You will work with the following areas:

- Development Environment
- Project Management
- Work with Software Action Request
- Work with Software Versions Repository
- CASE Profiles
Version Control Process Flow

- Set Up the SAR System
  - Create record type codes
  - Define record type titles
  - Create SARs

- Activate SAR Logging
  - Activate SAR logging (CASE profile)

- Develop Software
  - Begin to create or change code related to the project using SVR

- Define a Promotion Path
  - Add a promotion path
  - Assign the “From” and “To” environments to source code members and control tables

- Define a Project
  - Add a project
  - Assign promotion paths and SARs to the project

- Update the SARs by Using the SAR Log
  - Update the Work Order Detailable (Fi802)

- Validate the promotion paths
  - Validate the promotion paths

- Promote a Project
  - Promote the project to the next environment from the Project Elements form

- Use Version Control for Project Updates
  - Perform version control procedures for a project update
Version Control Menu Overview

The figure below is the Version Control Menu. From this form, you will access the different features of the software development and promotion process.
Development Environment

About a Development Environment

A development environment contains objects and data being tested and edited. It is different from your production environment because it should not contain any live data files.

Rules for Creating Development Environments

You should be aware of the following rules when you create development libraries:

- Do not begin library names with Q, JDF, or JD Edwards World because of IBM standards and the upgrade process.
- Create custom libraries for custom modifications.
- Library names should be a maximum of 9 characters in length because of the upgrade process.
- Do not use JDFDATA for your own test data or live data because of the upgrade process.
- Do not include JDFDATA in a live user's library list.

To create a Development Environment complete the following tasks:

- Create Libraries
- Define Access for a User Profile using J98INITA
- Define Access for a User Profile using J98INIT
- Copy Data to Your Development Environment

This section contains the following:

- JD Edwards World Libraries
- Production and Development Examples
- Creating Libraries
- Creating Common and Data Libraries
- Creating a Development Object Library
- Creating a Development Source Library
- Creating JDESRC for JD Edwards World Program Generator
- Creating JDESRC Without the Program Generator
- About User Profiles
• Defining Access for a User Profile using J98INITA
• Defining Access for a User Profile Using J98INIT
• Copy Data to Your Development Environment
• Copying a Library
• Copying a File
• Copying a Record
• Copying JD Edwards World Control Records

JD Edwards World Libraries

The following libraries are delivered with JD Edwards World software:

• Source Library (JDFSRC)
• Object Library (JDFOBJ)
• Data Library (JDFDATA)
• Install Library (JDEINSTAL)
• Plans Library (JDFINS)
• Security Library (CLTSEC) (Optional)

Source Library (JDFSRC)

This is the library that contains source code. Within the JDFSRC library, JD Edwards World has three multi-member source files. The source files and their contents are described below.

The file JDESRC contains the following source code:

• RPG Programs
• Printer files
• Display files
• CL Programs
• DDS for physical files
• DDS for logical files

The file JDECPY contains the source code for copy members.

The file F98CRTCMD contains pre-compiler commands.

• This is used to compile selected JD Edwards World programs.
Object Library (JDFOBJ)

The object library contains objects for your JD Edwards World software.
- RPG programs
- CL programs
- Display files
- Report files

Data Library (JDFDATA)

The data library contains data files for your JD Edwards World software (files in this library contain test data provided by JD Edwards World).

Install Library (JDEINSTAL)

The install library is used to install programs and software that upgrade JD Edwards World software.

Plans Library (JDFINS)

The library is used to plan how to upgrade JD Edwards World software.

Security Library (CLTSEC)

You can create a security library which is shared across all environments. The benefit of having a security library is that you enter a user profile only once to have access to any environment. The following files may exist in the security library:
- User library list (F0092)
- Library list control (F0093)
- Library list master (F0094)
- User Preference (F00921)

In addition, all logical files associated with the above files must also exist in the security library.

The World Technical Foundation manual states that a security library containing the F0092, F00921, F0093, F0094, and F0095 files can be used to minimize profile maintenance between JD Edwards World environments.

**Note:** There are many different security scenarios. For that reason, there is little documentation on the security library concept. This is just a discussion of how to use the Security library concept and is NOT a recommendation.

A security library makes most sense when J98INITA is the Initial Program on the IBM user profile. IBM object security may be necessary in addition to the JD Edwards World security options to complete the user security requirements.
1. The benefits of a security library are minimized if there is more than one. The maintenance and security tasks will have to be duplicated for each environment. If each environment has a different security scenario, a single security library should not be used. However, if separate security libraries are necessary, there will have to be a matching object library with the QJDF data area naming the security library in the User Profile Library field.

2. If the environments are not all at the same release (A73, A81, etc), the sign-on programs, the menu program, and the control files will all have to be at the highest release level.

3. If all the environments are at the same release level, other files besides the security files may be placed in the Security library for maintenance or control purposes (for example SVR, Function Key Security, Action code security). Any files in the Security library should apply to all environments and should not be found in any other user data library. The pristine JDFDATA library should still contain all the JD Edwards World files.

4. On an Upgrade Plan, the Security library should be specified as SEC type. The file duplication issues of paragraph 3 above must be emphasized. The Control File Sets in the Control File Dependencies appendix of the PTF Install Workbook should be maintained.

Production and Development Examples

There are many ways to set up a production and development environment. The following are some examples.

**Basic Production Environment**

<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTEMP</td>
<td>IBM Temporary data files</td>
</tr>
<tr>
<td>CLTOBJ</td>
<td>Client’s objects</td>
</tr>
<tr>
<td>JDFOBJ</td>
<td>JD Edwards World objects</td>
</tr>
<tr>
<td>CLTCOM</td>
<td>Client’s common files</td>
</tr>
<tr>
<td>CLTDTA</td>
<td>Client’s data files</td>
</tr>
<tr>
<td>CLTSEC</td>
<td>Client’s security files</td>
</tr>
<tr>
<td>QGPL</td>
<td>IBM general public library</td>
</tr>
</tbody>
</table>

**Basic Development Environment**

<table>
<thead>
<tr>
<th>Library</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTEMP</td>
<td>IBM Temporary data files</td>
</tr>
</tbody>
</table>
All modifications and tests are performed in the development environment with the program’s object and source residing in DEVOBJ and DEVSRC. After you complete the testing, the program’s object is moved from DEVOBJ to CLTOBJ and the source is moved from DEVSRC to CLTSRC. You must create a separate data and common library (DEVDTA and DEVCOM) to ensure that any data changes during testing in the development environment do not affect live data in the production environment.

### No Source in Production Environment and a Common Shared Library

<table>
<thead>
<tr>
<th>Library</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTEMP</td>
<td>IBM Temporary data files</td>
</tr>
<tr>
<td>CLTOBJ</td>
<td>Client’s objects</td>
</tr>
<tr>
<td>JDFOBJ</td>
<td>JD Edwards World objects</td>
</tr>
<tr>
<td>CLTCOM</td>
<td>Client’s common files</td>
</tr>
<tr>
<td>COMMON</td>
<td>Common (shared) files</td>
</tr>
<tr>
<td>CLTDTA</td>
<td>Client’s data files</td>
</tr>
<tr>
<td>CLTSEC</td>
<td>Client’s security files</td>
</tr>
<tr>
<td>QGPL</td>
<td>IBM general public library</td>
</tr>
</tbody>
</table>
Basic Development Environment with a Shared Common

<table>
<thead>
<tr>
<th>Library</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTEMP</td>
<td>IBM Temporary data files</td>
</tr>
<tr>
<td>DEVOBJ</td>
<td>Development objects</td>
</tr>
<tr>
<td>CLTOBJ</td>
<td>Client’s objects</td>
</tr>
<tr>
<td>JDFOBJ</td>
<td>JD Edwards World objects</td>
</tr>
<tr>
<td>DEVCOM</td>
<td>Development common files</td>
</tr>
<tr>
<td>COMMON</td>
<td>Common (shared) files</td>
</tr>
<tr>
<td>DEVDTA</td>
<td>Development data files</td>
</tr>
<tr>
<td>CLTSEC</td>
<td>Client’s security files</td>
</tr>
<tr>
<td>DEVSRC</td>
<td>Development source files</td>
</tr>
<tr>
<td>CLTSRC</td>
<td>Client’s source files</td>
</tr>
<tr>
<td>JDFSRC</td>
<td>JD Edwards World source files</td>
</tr>
<tr>
<td>QGPL</td>
<td>IBM general public library</td>
</tr>
</tbody>
</table>

No source libraries exist in the production environment because source code is not necessary to run JD Edwards World programs. This makes the production environment easier to maintain. The only restriction is that users in the production environment cannot view source code. Another difference is that a third shared common library (COMMON) has been added to the environments. This library contains common files whose data may be changed during the testing process. For example, F98HELP may not be changed but the Vocabulary Overrides, Data Dictionary, User Defined Codes or other Control Files may be changed. By having this type of common library not only are the environments easy to maintain, but you save considerable machine resources.

One Development Source and Object Library

<table>
<thead>
<tr>
<th>Library</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>QTEMP</td>
<td>IBM Temporary data files</td>
</tr>
<tr>
<td>CLTMOD</td>
<td>Client’s source and objects under modification</td>
</tr>
<tr>
<td>CLTOBJ</td>
<td>Client’s objects</td>
</tr>
<tr>
<td>JDFOBJ</td>
<td>JD Edwards World objects</td>
</tr>
<tr>
<td>DEVCOM</td>
<td>Development common files</td>
</tr>
<tr>
<td>COMMON</td>
<td>Common unchanged files</td>
</tr>
</tbody>
</table>
### Library Explanation

<table>
<thead>
<tr>
<th>Library</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVDTA</td>
<td>Development data files</td>
</tr>
<tr>
<td>CLTSEC</td>
<td>Client’s security files</td>
</tr>
<tr>
<td>CLTSRC</td>
<td>Client’s source files</td>
</tr>
<tr>
<td>JDFSRC</td>
<td>JD Edwards World source files</td>
</tr>
<tr>
<td>QGPL</td>
<td>IBM general public library</td>
</tr>
</tbody>
</table>

DEVOBJ and DEVSRFC have been combined into one library called CLTMOD. This library contains both source code and compiled objects for programs while they are being modified and tested. After testing, the program objects are copied to CLTOBJ and source code is moved to CLTSRC. The purpose of having one object and source code library like CLTMOD is to aid in change management and simplify the development library list by having one place where all modifications and testing take place.

### Creating Libraries

Create the following libraries:
- Common and data libraries
- Development object library
- Development source library

If you create a common library (DEVCOM), be sure to specify it each time you create another development environment. If you do not specify the common library each time, the files will be created in your development data library.

Your common library should contain control files holding data that is used in several environments that is not application data. These control files may be changed during the development process.

See Appendix A: Common and Production Library Files for a list of common and production files. Common files will be marked with a “Y” in the Common File field on the SVR screen. Files marked with an “N” in this field are Production files. If the data is to be copied from JDFDATA into either of these files, the SVR Copy Data Y/N field will be marked ‘Y’.

### Creating Common and Data Libraries

You will create the libraries that contain common data files (DEVCOM) and test data files (DEVDTA).
To create common and data libraries

From Computer Operations (G96), choose Data Base Management
From Data Base Management (G9645), choose Data Libraries

On Create Production Environment

<table>
<thead>
<tr>
<th>98112</th>
<th>Create User Data Libraries</th>
<th>Form ID.  .  .  .  F98102</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Version.  .  .  .  .  .   ZJD80001</td>
<td></td>
</tr>
</tbody>
</table>

Create Production Environment
This job has various options described below. Enter the desired values and press ENTER to continue.

Enter the "FROM" Library where data is to be copied from (e.g. JDEDATA).

Enter the "TO" Production Library where you are creating files (e.g. PROCLIB).

Enter the "TO" Common Library where you are creating common files (e.g. COMMON).

If you do not enter a Common library, all common files will be created in the Production Library.

1. Complete the Create User Data Libraries form.
   - Once you correctly complete the form and press Enter, the job (J98102) is submitted to batch.

2. Repeat the above step for each of the development data libraries you have.
   - The program automatically:
     - Creates your libraries
     - Creates the physical and logical files that should be maintained in your common library
     - Creates the physical and logical files necessary for operations control in your development library
     - Creates the physical and logical files for various applications in your development library
     - Generates reports to identify all the physical, logical, and join files created and to identify where they were created
     - Generates a report to identify all the optional files. The report explains why the files are optional so that you can determine if they should be deleted.
Creating a Development Object Library

To create a development object library

Type the IBM command Create Library (CRTLIB) and press F4.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Your development object library name.</td>
</tr>
<tr>
<td>Library Type</td>
<td>*PROD or *TEST</td>
</tr>
<tr>
<td>Text 'description'</td>
<td>The description of your library</td>
</tr>
</tbody>
</table>

Creating a Development Source Library

To create the development source library (DEVSRC), you create a source environment and a source physical file. The source physical file is the Program Source File (JDESRC). All JD Edwards World source members are located in the JDESRC file.

There are two possible methods to create the JDESRC file. You must determine if you have the JD Edwards World Program Generator (CASE) and then choose the appropriate method.
To create a development source library

Type the IBM command Create Library (CRTLIB) and press F4.

<table>
<thead>
<tr>
<th>Create Library (CRTLIB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
</tr>
<tr>
<td>Library . . . . . . . . . DEVSRC Name</td>
</tr>
<tr>
<td>Library type . . . . . . . *TEST *PROD, *TEST</td>
</tr>
<tr>
<td>Text 'description' . . . . *BLANK</td>
</tr>
</tbody>
</table>

Field | Explanation
---|---
Library | Your development object library name.
Library Type | *PROD or *TEST
Text 'description' | The description of your library

Creating JDESRC for JD Edwards World Program Generator

When an RPGIII or RPGIV program is moved into production at JD Edwards World, the record length is 92 bytes. If you have J.D. Edward's Program Generator product, the program source file format must be 142 bytes to allow for the Program Generator Serial Number and additional required data.

To create JDESRC for JD Edwards World Program Generator

1. To copy an existing file with the correct format (F93002), type the IBM Copy File command (CPYF) and press F4.

<table>
<thead>
<tr>
<th>Copy File (CPYF)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
</tr>
<tr>
<td>From file . . . . . . . . P93002 Name</td>
</tr>
<tr>
<td>To file . . . . . . . . DEVSRC Name, *LIBL, *CURLIB</td>
</tr>
<tr>
<td>Library . . . . . . . . DEVSRC Name, *LIBL, *CURLIB</td>
</tr>
<tr>
<td>From member . . . . . . . <em>FIRST Name, generic</em>, *FIRST, *ALL</td>
</tr>
<tr>
<td>To member or label . . . . *FIRST Name, *FIRST, *PROMSER</td>
</tr>
<tr>
<td>Replace or add records . . . *NONE *NONE, *ADD, *REPLACE</td>
</tr>
<tr>
<td>Create file . . . . . . . *YES *NO, *YES</td>
</tr>
<tr>
<td>Print format . . . . . . . *CHAR *CHAR, *HEX</td>
</tr>
</tbody>
</table>

Bottom

F1=Exit F4=Prompt F5=Refresh F16=Additional parameters F12=Cancel
F13=How to use this display F24=More keys
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>From file</td>
<td>The file and library containing the data to be copied. The file is F93002 and the library can default to *LIBL.</td>
</tr>
<tr>
<td>To file</td>
<td>The name of the source file and your development source library. Generally, the file is JDESRC and the library is DEVSRC.</td>
</tr>
<tr>
<td>From member</td>
<td>The member name that will be the beginning of the copy process. Generally, this value is *FIRST.</td>
</tr>
<tr>
<td>To member or label</td>
<td>The member name that will be the beginning of the receiving process. Generally, this value is *FIRST.</td>
</tr>
<tr>
<td>Replace or add records</td>
<td>Specifies whether the records copied should replace or be added to the records in the To file. In this case since the To file does not exist, this value is *NONE.</td>
</tr>
<tr>
<td>Create file</td>
<td>Specifies whether the To file does not exist and needs to be created. This value is *YES.</td>
</tr>
<tr>
<td>Print format</td>
<td>Specifies whether the characters are printed in character or character and hexadecimal format. This option only applies if the To file is *PRINT.</td>
</tr>
</tbody>
</table>

2. To remove the empty member copied from JDESRC, type the IBM Remove Member command (RMVM) and press F4.

```
Remove Member (RMVM)
Type choices, press Enter.
Data base file . . . . . . . . JDESRC     Name
Library . . . . . . . . . . . . DEVSRC     Name, *LIBL, *CURLIB
Member . . . . . . . . . . . . F93002      Name, generic*, *ALL
```

### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database file</td>
<td>Type the source file and your development source library that contains the record to be removed. Generally, this file is JDESRC and the library is DEVSRC.</td>
</tr>
<tr>
<td>Member</td>
<td>Type the name of the member that is to be removed. This is F93002.</td>
</tr>
</tbody>
</table>
Creating JDESRC Without the Program Generator

If you do not have JD Edwards World World Program Generator product, the program source file format can remain at 92 bytes, as it is when a program is moved into production at JD Edwards World. To create the JDESRC file with a 92 byte record format, you can execute the IBM Create Source Physical File command (CRTSRCPF).

To create JDESRC without the Program Generator

Type the Create Source Physical File command (CRTSRCPF) and press F4.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>File</td>
<td>The source file and your development source library that contains the record to be removed. Generally, this file is JDESRC and the library is DEVSRC.</td>
</tr>
<tr>
<td>Record Length</td>
<td>The number of bytes in the length of the records to be stored in the source file. This value is 92.</td>
</tr>
<tr>
<td>Member, if desired</td>
<td>The member to be added to the source file. Generally, this member is left to *NONE.</td>
</tr>
<tr>
<td>Text Description</td>
<td>The description of your source file.</td>
</tr>
</tbody>
</table>

About User Profiles

You must create profiles that allow users to have access to new environments. There are two separate methods to define access to an environment. The method you choose depends upon whether the User Profile accesses JD Edwards World software using J98INITA or J98INIT.
Defining Access for a User Profile using J98INITA

From Security Office (G94), choose Library List Control
From Library List Control (G944), choose Library List Revisions

If you are allowing access to your development environment for a user profile that is using J98INITA, you must define a development environment library list name. In addition, the User Signon List must contain the Development Environment Library List name.

To define access for a user profile using J98INITA

1. On Library List Revisions define the libraries that comprise the list.

```plaintext
0094 Library List Revisions
Action Code . . . I
Library List Name . . TEMP
Description . . . Technical Training Example
Menu Program ID . . FOMENU
Library List . . . OTMSP_DEVOBJ CLTRBJ_JDPJOB DEVITA_DEVCOM DEVSEC_CLTRBJ_DEVSEC QJF

F21=Print Library List F9=Library Search
```

2. To assign the library list to each user, select User Signon List Revisions from the Library List Control menu (G944).

```plaintext
0094 User Signon List Revisions
Action Code . I
User ID . . FRAZINI
Seq Library Sign-on
Number List Menu Description
5.00 FRAZINI A32 MASTER FRAZINI DATA LIBL
10.00 A5JDEV A32 A5-2 Case Cert & S Development
11.00 FRAZINI A32 Testing A52 Program Generator
20.00 TEC_TP A32 * List Name Not in Master File
30.00 TEC_TP A32 Testing A52 Tech Foundations
55.00 REDCASE A32 * List Name Not in Master File
```

Defining Access for a User Profile Using J98INIT

If you are allowing access to your development environment for a user profile that is using J98INIT, you must define a new library list.
To define access for a User Profile using J98INIT

Select User Information from the Security Officer menu (G94).

<table>
<thead>
<tr>
<th>092</th>
<th>User Information</th>
<th>Action Code</th>
<th>User ID</th>
<th>User Key</th>
<th>Allow Command Entry (Y/N)</th>
<th>Initial Menu to Execute</th>
<th>Allow Menu Traveling (Y/N)</th>
<th>Initial Program to Execute</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: Each user profile for the JD Edwards World software must have an IBM profile. To define an IBM profile, use the command, Create User Profile (CRTUSRPRF). When a JD Edwards World user profile is created or changed, the IBM JOBD is created or changed in the library specified in the Processing Option for J00JOBD. The default library is QGPL. If J98INITA is used as the initial program to execute, the JOBD is copied into QTEMP and modified.

Copy Data to Your Development Environment

You can use several methods to copy data to your development environment. The method you choose should depend upon how much data you need to copy to your development environment. You may copy the following:

- Libraries
- Files
- Records
- JD Edwards World Record Types

Copying a Library

If you need to duplicate several files into your development environment you can copy one or more libraries.
To copy a library

To display the parameters, type the IBM Copy Library command (CPYLIB) and press F4.

```
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Existing Library</td>
<td>The library to be copied in your Production Environment.</td>
</tr>
<tr>
<td>New Library</td>
<td>The new library that will be used in your Development Environment</td>
</tr>
<tr>
<td>Create Library</td>
<td>Specifies whether the New Library does not exist and needs to be created.</td>
</tr>
</tbody>
</table>
```

**Caution:** If you use CPYLIB, any files that are in use are not copied.

### Copying a File

From Computer Operations (G96), choose **Data Base Management**
From Data Base Management (G9645), choose **Data Files**

If you need to copy specific files from a library in your production environment to a library in your development environment, you use the JD Edwards World copy file utility.

#### To copy a file

1. On Data File Creation enter the following fields:
   - System Code
   - Create in Library
   - FROM Library
2. Do the following:
   - Enter a 1 next to the files you wish to create from source.
   - Enter a 2 next to the files to be duplicated without data. This is mainly Logical files but could be a Physical file also. Logical files will only be created if the based-on physical file is in the destination library.
Enter a 3 next to the files to be duplicated with data. This applies only to Physical files.

3. Each selection will be submitted to batch.

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0070</td>
<td>PHYSICAL</td>
<td>Country Constants Master File</td>
</tr>
<tr>
<td>P009101</td>
<td>PHYSICAL</td>
<td>Word Search Occurrence Master</td>
</tr>
<tr>
<td>P0101</td>
<td>PHYSICAL</td>
<td>Address Book Master</td>
</tr>
<tr>
<td>P0101A</td>
<td>PHYSICAL</td>
<td>Address Book Master File Audit Log</td>
</tr>
<tr>
<td>P0101E</td>
<td>PHYSICAL</td>
<td>Address Book Master</td>
</tr>
<tr>
<td>P0101F1</td>
<td>PHYSICAL</td>
<td>Address Book Batch File</td>
</tr>
<tr>
<td>P01090</td>
<td>PHYSICAL</td>
<td>Supp. Data Base - CORE</td>
</tr>
<tr>
<td>P01092</td>
<td>PHYSICAL</td>
<td>Supp. Data Base - Code</td>
</tr>
<tr>
<td>P01093</td>
<td>PHYSICAL</td>
<td>Supp. Data Base - Narrative</td>
</tr>
<tr>
<td>P01094</td>
<td>PHYSICAL</td>
<td>User Search Preference</td>
</tr>
<tr>
<td>P0111</td>
<td>PHYSICAL</td>
<td>Address Book - Who’s Who</td>
</tr>
<tr>
<td>P0114</td>
<td>PHYSICAL</td>
<td>Address Book Memo/Text Information</td>
</tr>
<tr>
<td>P0118W</td>
<td>PHYSICAL</td>
<td>WF - Memo Information Work File</td>
</tr>
<tr>
<td>P0116</td>
<td>PHYSICAL</td>
<td>Address Book Locations</td>
</tr>
<tr>
<td>P01800</td>
<td>PHYSICAL</td>
<td>Address Book Word Search Master</td>
</tr>
</tbody>
</table>

Note: When using this utility, be sure to copy all related files. All the physical files are listed first, followed by the logical files.

Copying a Record

If you wish to copy a file with only selected records, use the IBM Copy File command (CPYF).

To copy a record

1. Type the Copy File command (CPYF) and press F4.

To copy a record

2. Press F10 to display additional parameters.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>From file</td>
<td>The file and library containing the data to be copied.</td>
</tr>
</tbody>
</table>
### 3B Development Environment

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>To file</td>
<td>The name of the file and your development library the data will be copied to.</td>
</tr>
<tr>
<td>From member</td>
<td>The member name that will be the beginning of the copy process.</td>
</tr>
<tr>
<td>Replace or add records</td>
<td>Specifies whether the records copied should replace or be added to the records in the To file.</td>
</tr>
<tr>
<td>Create file</td>
<td>Specifies whether the To file does not exist and needs to be created.</td>
</tr>
<tr>
<td>Print format</td>
<td>Specifies whether the characters are printed in character or character and hexadecimal format. This option only applies if the To file is *PRINT.</td>
</tr>
<tr>
<td>Copy from record number</td>
<td>Specifies the record number from which to start the copy.</td>
</tr>
</tbody>
</table>

3. Page down and enter the record number of the record to which you wish to copy.

The Copy to record number is the field in which you specify the record number of the last record to be copied.

![Copy File (CFTF)](image)

**Field** | **Explanation**
--- | ---
Copy to Record Number | Specifies the record number of the last record to be copied.
Copy from Record Key | Only applies when copying a file with keyed fields.

---

**Copying JD Edwards World Control Records**

You can copy any of the following control records:

- Vocabulary Overrides
To copy a JD Edwards World record type

From Computer Assisted Programming (CAP) (G93), choose Developer's Workbench
From Developer's Workbench (G9362), choose Copy DD, VO, DW, UDC, SVR, Menus

On Copy DD, VO, DW, UDC, SVR, Menus.

<table>
<thead>
<tr>
<th>Field Explanation</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Library</td>
<td>The library containing the data to be copied.</td>
</tr>
<tr>
<td>To Library</td>
<td>The library in your Development Environment to receive the data.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Dictionary Item           | The RPG data name. This data field has been set up as a 10-byte field for future use. Currently, it is restricted to 4 bytes so that, when preceded by a 2-byte file prefix, the RPG data name does not exceed 6 bytes. Within the Data Dictionary, all data items are referenced by this 4-byte data name. As they are used in database tables, a 2-character prefix is added to create unique data names in each table specification (DDS). Special characters are not allowed as part of the data item name, with the exception of #, @, $.
|                           | You can create protected data names by using $xxx and @xxx, where you define xxx. Messages can contain up to 10 characters. Types of messages are further defined by glossary group. |
| Vocabulary Overrides      | The name of the screen or report record to be copied. All records for soft coding will be copied.                                              |
| DREAM Writer Form         | The name of the DREAM Writer Form ID to be copied. All versions of the specified form will be copied.                                          |
| User Def Codes Sys        | The system code and type of the table to be copied. All values for the specified table will be copied.                                        |
| Software Versions Rep     | The record of the Software Versions Repository member to be copied.                                                                        |
| Menu Identification       | The menu ID and the display language of the record to be copied.                                                                          |

**Note:** You can enter and copy only one item at a time. If the item exists in the To Library, it is replaced.
Project Management

About Project Management

To manage projects you may use Work Order Processing. This section contains the following:

- Understanding Work Order Processing
- Creating Work Orders
- Accessing the Scheduling Workbench
- Adding Record Types
- Changing Record Types

Understanding Work Order Processing

The Software Action Request System (SAR) is shipped to clients under the name of Work Order Processing.

The Work Order Processing system allows you to:

- Create and classify work orders with simple budgets or estimates
- Schedule and expedite work orders.
- Perform cost accounting by specific work orders or family of work orders.

Unlike jobs that are often preplanned and thoroughly budgeted, software work orders are often completed without the prior knowledge of the accounting department. Work orders are frequently spontaneous and of short duration.

Note: If you purchased system 48 (Work Order Processing), you have all of the programs associated with Work Orders (SARs). If you have not purchased the Work Order Processing system, you have only the programs from the Work Order Processing system that are defined as being part of the General Back Office System (00).
Creating Work Orders

There are only three required fields when creating a new work order:

- Work Order Number (If you do not provide a work order number, the system assigns one automatically.)
- Description (short)
- Charge to Business Unit

To create work orders

From the Simple Project Management menu (G4812), select Single Task Details.

<table>
<thead>
<tr>
<th>48014</th>
<th>Single Task Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Code...</td>
<td>I</td>
</tr>
<tr>
<td>Description...</td>
<td>ACOS Class</td>
</tr>
<tr>
<td>Status Comment</td>
<td>Student SAR</td>
</tr>
<tr>
<td>Search X-Ref...</td>
<td>Cost Code...</td>
</tr>
<tr>
<td>Ret. Hours...</td>
<td>40</td>
</tr>
<tr>
<td>Ret. Amount...</td>
<td>1,500</td>
</tr>
<tr>
<td>Phase...</td>
<td>55</td>
</tr>
<tr>
<td>Type...</td>
<td>2</td>
</tr>
<tr>
<td>Tax Expl Code...</td>
<td>1001</td>
</tr>
<tr>
<td>Subledger Inact.</td>
<td>001</td>
</tr>
<tr>
<td>Customer No.</td>
<td>Edwards, J.D.</td>
</tr>
<tr>
<td>Manager.</td>
<td>Allen, Ray</td>
</tr>
<tr>
<td>Description</td>
<td>Option</td>
</tr>
</tbody>
</table>

GAR setup for work to be performed during the Advanced Programming Concepts and Skills class

(continued)

What You Should Know About

Accessing the W.O. Detail form

To access the W.O. Detail form, choose More Description (F5).

Searching for address numbers

To search for address numbers for the Customer Number and Manager fields, use F1 on the field, or choose More Keys (F24), then Exit to Name Search.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Parent W.O. No        | This is the parent work order number. You can use this number to:  
  1. Enter default values for newly added work orders, for example, Type, Priority, Status, or Manager.  
  2. Group work orders for project setup and reporting  
  Form-specific information |
| Action Code           | A code that indicates the activity you want to perform. Valid codes are:  
  A Add new record  
  C Change existing record  
  D Delete existing record  
  I Inquire on existing record  
  . End of program or function  
  space Clear the form  
  If you enter a code that is not active, the system highlights the code and no action occurs. |
<p>| W.O. Number           | The number that identifies an original document. This can be a voucher, an order number, an invoice, unapplied cash, a journal entry number, and so on. |
| Description           | A brief description of an item, a remark, or an explanation.                                                                                |
| Status Comment        | A brief description to explain the status of the work order.                                                                               |</p>
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Charge to BU</td>
<td>An alphanumeric field that identifies a separate entity within a business for which you want to track costs. For example, a business unit might be a warehouse location, job, project, work center, or branch/plant. You can assign a business unit to a voucher, invoice, fixed asset, and so on, for purposes of responsibility reporting. For example, the system provides reports of open accounts payable and accounts receivable by business units to track equipment by responsible department. Security for this field can prevent you from locating business units for which you have no authority. <strong>Note:</strong> The system uses this value for Journal Entries if you do not enter a value in the AAI table.</td>
</tr>
<tr>
<td>Search X-Ref</td>
<td>An alphanumeric value used as a cross-reference or secondary reference number. Typically, this is the customer number, supplier number, or job number.</td>
</tr>
<tr>
<td>Cost Code</td>
<td>A subdivision of an object account. Subsidiary accounts include more detailed records of the accounting activity for an object account.</td>
</tr>
<tr>
<td>Est. Hours</td>
<td>The estimated hours that are budgeted for this work order.</td>
</tr>
<tr>
<td>Est. Amount</td>
<td>The estimated dollar amount that is budgeted for this work order.</td>
</tr>
<tr>
<td>Start Date</td>
<td>This is a start date that you can enter, or an automatic start date which the planning system calculates using a backscheduling routine. The routine starts with the required date and offsets the total leadtime to calculate the appropriate start date. Will default from system date or you can enter a date.</td>
</tr>
<tr>
<td>Planned Comp</td>
<td>The date the work order is planned to be completed.</td>
</tr>
<tr>
<td>Phase</td>
<td>A user defined code (00/W1) that indicates the current stage or phase of development for a work order. You can assign a work order to only one phase code at a time. <strong>Note:</strong> Certain forms contain a processing option that allows you to enter a default value for this field. If you enter a default value on a form for which you have set this processing option, the system displays the value in the appropriate fields on any work orders that you create. The system also displays the value on the Project Setup form. You can either accept or override the default value.</td>
</tr>
<tr>
<td>Completed</td>
<td>The date the work order or engineering change order is completed or canceled.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Type</td>
<td>A user defined code (00/ TY) that indicates the type classification of a work order or engineering change order. You can use work order type as a selection criteria for work order approvals.</td>
</tr>
<tr>
<td>Priority</td>
<td>A user defined code (system 00, type PR) that indicates the relative priority of a work order or engineering change order in relation to other orders.</td>
</tr>
<tr>
<td></td>
<td>A processing option for some forms lets you enter a default value for this field. The value then displays automatically in the appropriate fields on any work order you create on those forms and on the Project Setup form. You can either accept or override the default value.</td>
</tr>
<tr>
<td>Status</td>
<td>A user defined code (00/ SS) that describes the status of a work order or engineering change order. Any status change from 90 thru 99 automatically updates the date completed.</td>
</tr>
<tr>
<td>Customer No</td>
<td>A number that identifies an entry in the Address Book system. Use this number to identify employees, applicants, participants, customers, suppliers, tenants, and any other Address Book members.</td>
</tr>
<tr>
<td>Manager</td>
<td>The address book number of a manager or planner.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> A processing option for some forms lets you enter a default value for this field based on values for Category Codes 1 (Phase), 2, and 3. Set up the default values on the Default Managers and Supervisors form. After you set up the default values and the processing option, the information displays automatically on any work orders you create if the category code criterion is met. (You can either accept or override the default value.)</td>
</tr>
<tr>
<td>Transaction</td>
<td>The date that an order was entered into the system. This date determines which effective level that the system uses for inventory pricing.</td>
</tr>
<tr>
<td>Date Assigned</td>
<td>The date the person responsible for the work order receives the work order.</td>
</tr>
<tr>
<td>Tax Expl Code</td>
<td>A user defined code (00/ EX) that controls how a tax is assessed and distributed to the general ledger revenue and expense accounts. You assign this code to a customer or supplier to set up a default code for their transactions. Do not confuse this with the taxable, non-taxable code. A single invoice can have both taxable and non-taxable items. The entire invoice, however, must have one tax explanation code.</td>
</tr>
</tbody>
</table>
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tax Rate/Area</strong></td>
<td>A code that identifies a tax or geographic area that has common tax rates and tax distribution. The tax rate/area must be defined to include the tax authorities (for example, state, county, city, rapid transit district, or province), and their rates. To be valid, a code must be set up in the Tax Rate/Area table (F4008). Typically, U.S. sales and use taxes require multiple tax authorities per tax rate/area, whereas VAT requires only one simple rate. The system uses this code to properly calculate the tax amount.</td>
</tr>
<tr>
<td><strong>Subledger Inact</strong></td>
<td>A code that indicates whether a specific subledger is active or inactive. Any value other than blank indicates that a subledger is inactive. Examples are jobs that are closed, employees that have been terminated, or assets that have been disposed. If a subledger becomes active again, set this field back to blank. If you want to use subledger information in the tables for reports but want to prevent transactions from posting to the master record, enter a value other than blank in this field.</td>
</tr>
</tbody>
</table>

### Processing Options

Processing options associated with the Single Task Details program allow you to default the value for the Type, Priority, Status, Phase, Category Code 2, Category Code 3, and Manager fields. To see the processing options, type the selection number for Single Task Details and press F18.
Function Keys from Single Task Details

F5 - Detailed Specifications

F5 - Allows you to enter additional detailed information about your work order. Each detail screen is based on Record Type. Record Type A provides room for you to enter more description. You can customize other Record Types to fit your requirements. The steps to add and change Record Types are described later in this chapter.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - Insert</td>
<td>Insert a blank line for additional text.</td>
</tr>
<tr>
<td>9 - Delete</td>
<td>Delete a line of text</td>
</tr>
</tbody>
</table>

F8 - Category Codes

F8 - Allows you to update other work order values.
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>W.O. Number</td>
<td>The work order identification number. This value defaults from the Single Task Details.</td>
</tr>
<tr>
<td>W.O. Flash Message</td>
<td>A highlighted message that will be attached to the work order.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>The flash message appears as a highlighted message on Backlog Management, replacing the work order description.</td>
</tr>
<tr>
<td>Phase</td>
<td>A user defined code describing a stage or category in the development of a project. This value defaults from the Single Task Details.</td>
</tr>
<tr>
<td>Category 02</td>
<td>Category Codes that are user defined values associated with the work order.</td>
</tr>
<tr>
<td>Originator</td>
<td>The address number of the person who entered the work order. Must be a valid number in the Address Book Master file (F0101).</td>
</tr>
<tr>
<td>Supervisor</td>
<td>The address number of the work order supervisor. Must be a valid number in the Address Book Master table (F0101).</td>
</tr>
<tr>
<td>Std. Desc</td>
<td>A user defined code describing instructional information. Must be a valid number in the Address Book Master file (F0101).</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>For Equipment/ Plant Maintenance users:</td>
</tr>
<tr>
<td></td>
<td>You can use this code to assign narrative text for a standard procedure. The information appears on the Item PM schedule and the work order routing.</td>
</tr>
<tr>
<td>Search X-Ref</td>
<td>Any number or characters that will be used to cross-reference work orders. This value will default from the Single Task Defaults screen.</td>
</tr>
</tbody>
</table>

### F9 - Name Search

F9 - Allows you to search for a specific address book number.
F15 - Work Order Search Window

F15 - Allows you to search for work order descriptions. It will only return the description.

<table>
<thead>
<tr>
<th>W.O. No.</th>
<th>289</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Name</td>
<td>APCS Class</td>
</tr>
<tr>
<td>Status</td>
<td>Student SAR</td>
</tr>
<tr>
<td>Est. Hours</td>
<td>40</td>
</tr>
<tr>
<td>Est. Amount</td>
<td>1,200</td>
</tr>
<tr>
<td>Phase</td>
<td>15</td>
</tr>
<tr>
<td>Priority</td>
<td>2</td>
</tr>
<tr>
<td>Status</td>
<td>10</td>
</tr>
</tbody>
</table>

F21 - Print Work Order

F21 - Allows you to print the work order, including all of the associated record types.

Accessing the Scheduling Workbench

The Scheduling Workbench program allows you to review and update work orders. You can retrieve information about work orders in multiple ways. After retrieving the work orders that meet your search criteria, you can update selected fields in those work orders directly from the Scheduling Workbench form.

To access the Scheduling Workbench

From the Simple Project Management menu, select Scheduling Workbench.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Category Codes</td>
<td>Any number or characters that will be used to cross-reference work orders. This value will default from the Single Task Defaults screen.</td>
</tr>
<tr>
<td>Job or BU</td>
<td>The business unit that is responsible for charges incurred. Must be a valid business unit setup in the Business Unit Master File (F0006).</td>
</tr>
<tr>
<td>Originator</td>
<td>The address number of the person who entered the work order. Must be a valid number in the Address Book Master file (F0101).</td>
</tr>
<tr>
<td>Customer</td>
<td>The Address Number of the customer. Must be a valid number in the Address Book Master file (F0101).</td>
</tr>
<tr>
<td>Manager</td>
<td>The Address Number of the manager in charge of the work order. Must be a valid number in the Address Book Master File (F0101).</td>
</tr>
<tr>
<td>Supervisor</td>
<td>The address number of the work order supervisor. Must be a valid number in the Address Book Master table (F0101).</td>
</tr>
<tr>
<td>Parent W.O. No</td>
<td>Through parent work order number, you can group work orders together based on one parent work order, such as the installation of a computer and its associated electrical wiring, which may involve more than one customer or manager.</td>
</tr>
<tr>
<td>Form-specific information</td>
<td>The parent work order number which groups work orders together in a “family”.</td>
</tr>
<tr>
<td>Model</td>
<td>Determines whether model work orders will be displayed on the screen.</td>
</tr>
<tr>
<td>M</td>
<td>Determines whether model work orders will be displayed on the screen.</td>
</tr>
<tr>
<td>Search X-Ref</td>
<td>Any number or characters that will be used to cross-reference work orders. This value will default from the Single Task Defaults screen.</td>
</tr>
<tr>
<td>Cost Code</td>
<td>The subsidiary account responsible for incurred charges.</td>
</tr>
<tr>
<td>Number</td>
<td>The work order identification number. This value defaults from the Single Task Details.</td>
</tr>
</tbody>
</table>
### Field Explanation

**Description**
Describes the function or option exit.
Cannot exceed 40 characters.

Form-specific information
The name or a brief description of the work order.

**X-Ref No**
Any number or characters that will be used to cross-reference work orders. This value will default from the Single Task Defaults screen.

**Status**
A user defined code used to describe the current status of the work order; for example, planned, started, or completed.

**Status Comment**
This line allows status comments or further description of the work.

**Type**
User defined code describing the work order type.

**Priority P**
A user defined code used to assign the priority of the work order; for example, high, medium, or low.

---

**F4 - More Detail**

F4 - Displays additional information concerning each work order that is hidden in the Detail Area.

---

**Field**

**Explanation**

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planned Comp</td>
<td>The date the work is scheduled to be completed.</td>
</tr>
<tr>
<td>Hours Scheduled</td>
<td>The hours of work that has been scheduled.</td>
</tr>
<tr>
<td>Est. Hours</td>
<td>Total number of hours estimated for the work order.</td>
</tr>
<tr>
<td>Start Date Range</td>
<td>The initial date the work is scheduled to begin.</td>
</tr>
</tbody>
</table>
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>W.O. Flash Message</td>
<td>A highlighted message that will be attached to the work order.</td>
</tr>
<tr>
<td>W.O. Date</td>
<td>The date the work order was entered. Must be a valid number in the Address Book Master File (F0101).</td>
</tr>
</tbody>
</table>

### Selection Exits from the Scheduling Workbench

#### Selection 1 - Work Order Entry

Takes you to the Work Order Entry screen and automatically inquires on the selected work order.

#### Processing Options

There are processing options associated with the Scheduling Workbench program that allow you to default a Work Order Status Range and a Work Order Type. In addition, you can call either Project Task Details (P48014) or the Equipment Work Orders (P48011) when the W.O. Entry option is selected. Be aware that Equipment Work Orders (P48011) is part of the Work Order Processing system (48). To see the processing options, type the selection number for Scheduling Workbench and press F18.

### Adding Record Types

#### To add record types

1. From the Misc Additional Features menu (G4841), select Detail Spec. Types.

```
<table>
<thead>
<tr>
<th>00051</th>
<th>Detail Spec. Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Code. . . .</td>
<td>Install System Code. . . . 00</td>
</tr>
<tr>
<td>. . . .</td>
<td>User Defined Code . . . RT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>01 Character</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Full Description of Request</td>
</tr>
<tr>
<td>B</td>
<td>Final Disposition Remarks</td>
</tr>
<tr>
<td>C</td>
<td>Tool and Equipment Instructions</td>
</tr>
<tr>
<td>D</td>
<td>Safety Provisions</td>
</tr>
<tr>
<td>E</td>
<td>Plan and Drawing Reference</td>
</tr>
<tr>
<td>F</td>
<td>Equipment Drawn Items</td>
</tr>
</tbody>
</table>

P5 = Code Types  P14 = Memo  P15 = Where Used  P18 = Language  P21 = Print  P24 = More Keys
```

2. Add your specified record type and description to the table.
Changing Record Types

You can change the format of your record type.

To change record types

1. From the Misc Additional Features menu (G4841), select Detail Spec. Over Titles.

```
  4802    Detail Spec. Over Titles
  Action Code... L
  Record Type... E

  Sub-Title 1  Sub-Title 2  Sub-Title 3
  Equipment#  ProAction  ProAction
  _Number_    _Time Out_  _Time In_
```

2. Enter the heading text of each column you wish to add to the format of your Record Type.

   - Work Order (SAR) file is F4801
   - Detail Record Type file is F4802
   - Method of tracking programming projects

See Also

- For more information, consult the Work Orders Guide.
Work with Software Action Requests

About SAR System Setup

To set up a project, you must assign SARs and promotion paths to it. You create the SARs and define promotion paths first because the version control process uses the definitions.

After you set up your SAR system, you can develop the software. The SAR logging program keeps track of your changes as you have specified. While you develop the software, you can also define promotion paths and projects, and attach SARs to projects.

After you finish developing the software, you must update the SARs by using the SAR log before you promote the SAR.

Complete the following tasks:

- Creating Record Type Codes
- Defining Record Type Titles

Before You Begin

- The SAR system uses the Work Order files (F4801 and F4802). If your production environment uses these files, and if the F4802 file has different record types than what version control needs, define a separate library that contains these files for version control purposes only.

See Also

- Defining a Promotion Path

Creating Record Type Codes

The Work Order Instructions table (F4802) has an essential role in the version control process. It identifies and captures, for promotion purposes, all the source code members and control table data associated with a SAR. The Version Control system assigns a record type code to each source code member or control table data item, which classifies it for promotion. You must create record type codes that your Work Order Instructions table does not have currently.

To create record type codes

1. From the Version Control menu (G9261), choose Record Type Codes.
2. On User Defined Code Revisions enter the following character codes and descriptions:

<table>
<thead>
<tr>
<th>Character Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original Request</td>
</tr>
<tr>
<td>C</td>
<td>Members Affected</td>
</tr>
<tr>
<td>D</td>
<td>Menu Modifications</td>
</tr>
<tr>
<td>E</td>
<td>Automatic Accounting Instructions</td>
</tr>
<tr>
<td>F</td>
<td>Software Inventory Record Updates</td>
</tr>
<tr>
<td>G</td>
<td>Processing Options/ DREAM Writer</td>
</tr>
<tr>
<td>H</td>
<td>Vocabulary Override Changes</td>
</tr>
<tr>
<td>I</td>
<td>Database Changes</td>
</tr>
<tr>
<td>J</td>
<td>Constants Data File Changes</td>
</tr>
<tr>
<td>K</td>
<td>User Defined Code Changes</td>
</tr>
<tr>
<td>M</td>
<td>Connected SAR Numbers</td>
</tr>
<tr>
<td>N</td>
<td>Generic Rate/ Message Type Changes</td>
</tr>
<tr>
<td>O</td>
<td>Connected SAR Numbers</td>
</tr>
<tr>
<td>Q</td>
<td>Generic Rate/ Message Type Changes</td>
</tr>
<tr>
<td>S</td>
<td>Status History</td>
</tr>
<tr>
<td>U</td>
<td>Post-Installation Instructions</td>
</tr>
<tr>
<td>W</td>
<td>Pre-Compiler Commands</td>
</tr>
<tr>
<td>Z</td>
<td>First Included in PTF</td>
</tr>
</tbody>
</table>

PS=Code Types   F14-Memo   P15=Where Used   P21=Print   P18=Translate   F24=More
Defining Record Type Titles

For each record type code you create, you must also define record type titles, which appear as column headings on the W.O. Detail Entry form.

Before You Begin

Create record type codes before you define record type titles. See Creating Record Type Codes.

To define record type titles

1. From the Version Control menu (G9261), choose Record Type Titles.

2. For each record type you created, complete the following fields with the information in the chart that follows:
What You Should Know About

Verifying the record type titles

After you define the record type titles, you can view them to verify their accuracy. On Single Task Details, choose More Description. On W.O. Detail Entry, locate a record type you want to view by using the Record Type field.

To access Single Task Details, see Creating SARs.
The Software Versions Repository (SVR) consists of the following master directories.

- **F9801 Software Versions Repository Master**
  - A master directory of all files, programs, screens, reports, and copy modules.

- **F9802 Software Versions Repository Master**
  - Stores the member locations for each member master record.

**Working with Software Versions Repository (SVR)**

The Software Versions Repository indicates in what environments a requested member is located and whether the environment is a production or development environment. The data is used extensively for documentation and plays an important role in JD Edwards World Design and Development tools.

The Software Versions Repository is the natural starting point for all programming and software inquiry functions. It provides exits to the following features:

- Source Entry Utility (SEU)
- SAR Detail Entry
- Screen Design Aid (SDA)
- Report Design Aid (RDA)
- File Design Aid (FDA)
- The Program Generator
- Precompiler Commands
In addition, it provides access to the following functions:

- Copy Source
- Print Source
- Submit Creation of Object
- Generate Program Source and Help
- Edit Help Instructions
- Delete Source
- Print Help Instructions

This section contains the following:

- [Accessing the Software Versions Repository](#)
- [Member Identifiers](#)
- [Naming Conventions](#)
- [The JD Edwards World System Codes](#)
- [Examples of Program and File Names](#)
- [Optional Files Workbench](#)
- [Navigation Functions](#)
Accessing the Software Versions Repository

The Software Versions Repository serves as the front-end for all JD Edwards World design aids and programming utilities. You can also utilize this form as your own inventory file.

To access the Software Versions Repository

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

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member ID</td>
<td>The name of the Software Versions Repository member. Form-specific information</td>
</tr>
<tr>
<td></td>
<td>The source file contains the source member. In JD Edwards World, three source files reside in the JDFSRC library. They are:</td>
</tr>
<tr>
<td></td>
<td>• JDECPY for copy modules</td>
</tr>
<tr>
<td></td>
<td>• JDESRC for RPG, DDS, and CL source code</td>
</tr>
<tr>
<td></td>
<td>• F98CRTCMD for precompiler commands</td>
</tr>
</tbody>
</table>
### Field Explanation

**Description**
Identifying information of the member, such as Trial Balance by Business Unit. Associated programs, screens, and reports should share the same description. The description associated with each member is used to further identify the purpose of the member.

- Physical files should have a description that explains the purpose of the file.
- Screens, reports, and CL programs should have the same description as the associated RPG program.
- Logical files should be designated as follows: LF - fldname, fldname, fldname: where fldname is a key field.
- Join files should be designated as follows: JF - filename/ filename/ filename - fldname, fldname, fldname; where the filename is a file over which the join is built and fldname is the key field joining the files.
- Work files should be designated as follows: WF - filename; where filename is the file that the work file accesses.
- Copy modules carry their own unique descriptions.
- File Server programs should be designated as follows: File Server - filename; where filename is the file being served.

---

### Type, Use, and Associated Systems

The following fields identify the associated systems, along with their type and use.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Function Code</strong></td>
<td>Designates the object type such as display file, physical and logical files. Use F1 in the field to view the available types. RPG IV programs will use RPGL. RPG IV copy members will use CPYL. UDC 98/ F is used to control Member ID and Function Code combinations.</td>
</tr>
<tr>
<td><strong>Function Use</strong></td>
<td>Displays how the object is used in the system.</td>
</tr>
<tr>
<td><strong>System Code/ Product Code</strong></td>
<td>The system code the object is assigned to and delivered under.</td>
</tr>
<tr>
<td></td>
<td><strong>Form-specific information</strong></td>
</tr>
<tr>
<td></td>
<td>Designates the system number associated with the member. The configuration of installation media and the install process itself are driven by this install system code. Use F1 in the field to view valid codes.</td>
</tr>
</tbody>
</table>
Field | Explanation
--- | ---
Reporting System | Designates the system number for the using system. This may differ from the System/Product code. Exceptions occur for data files used by more than one system.

Member Relationship and Compiling Information

The following fields identify the logical grouping of members and information used in the compile process.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Base Member Name    | This field allows for the logical grouping of members. Form-specific information  
|                     | This field simply allows for logical grouping of members.  
|                     | For screens, reports, RPG programs and CL jobs, this name is usually the RPG program name associated with a particular member.  
|                     | For logical files, this name is the physical file upon which it is based and is required.                                                                                                                     |
| Omit Option         | Designates items in the Software Versions Repository file that would be bypassed for a new release. These codes are as follows:  
|                     | H Held from all releases  
|                     | X Omit from all releases  
|                     | S Omit Source from all releases  
|                     | O Omit Execution Object from all releases  
| Generation Sev      | Allows the user to designate a severity level when compiling a member.  
|                     | Because some JD Edwards World programs contain messages that appear in the compile listing as a severity level 10 error, it is suggested that you override the IBM default of a severity level 9 to a level 20 for all programs. To do this, enter the following on any command line:  
|                     | CHGCMDDFT CMD(CRTRPGPGM)  
|                     | NEWDFT(’GENLVL(20)’)  
|                     | RPG IV programs use CRTBNDRPG so the command to change that default is  
|                     | CHGCMDDFT CMD(CRTBNDRPG)  
|                     | NEWDFT(’GENLVL(20)’)  
|                     | For those specific programs that must override the new default severity level of 20, you can enter the override value in the Generation Severity field.                                                         |
Field | Explanation
---|---
Maint/ RSTDSP | Designates the type of maintenance on a logical file, how a screen will be processed, or if the program contains embedded SQL statements.

### Maintenance on a Logical File

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>No maintenance; or the logical is created dynamically.</td>
</tr>
<tr>
<td>1</td>
<td>Logical will be immediately updated when physical is updated.</td>
</tr>
<tr>
<td>2</td>
<td>Logical update will be delayed until the next time it is opened. - USE WITH CAUTION!</td>
</tr>
</tbody>
</table>

### Processing a Screen

<table>
<thead>
<tr>
<th>Value</th>
<th>Field Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RSTDSP = *NO&lt;br&gt;(Restore Display)&lt;br&gt;DFRWRT = *YES&lt;br&gt;(Defer Write)</td>
<td>Use with OVERLAY. Do not use with PUTOVR/ OVRDTA. All writes to the form field or file formats will be collected and written at one time.</td>
</tr>
<tr>
<td>A</td>
<td>RSTDSP = *NO&lt;br&gt;DFRWRT = *NO</td>
<td>Overlay&lt;br&gt;Each write statement is written to the screen.</td>
</tr>
<tr>
<td>B</td>
<td>RSTDST = *YES&lt;br&gt;DFRWRT = *NO</td>
<td>Use with PUTOVER to clear and write screen at field level.</td>
</tr>
<tr>
<td>S</td>
<td>RSTDSP = *YES&lt;br&gt;DFRWRT = *NO</td>
<td>Used when compiling SQL, RPG, RPGL and PL1 programs. For example, if SQL statements exist within an RPG program, the compiler: 1) Executes a create SQL program statement 2) Executes the SQL statements (converts them to calls) 3) Comments them out 4) Executes a CRTSQLRPG program statement and continues as normal</td>
</tr>
</tbody>
</table>

### File Information

The following fields identify the file information.
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Prefix</td>
<td>This field indicates the prefix associated with a file. Use F1 to display all file prefixes in use. Each physical file should have a unique file prefix.</td>
</tr>
<tr>
<td>Copy Data (Y/N)</td>
<td>Used to indicate if a database file must be copied with or without data. The Create User Data Libraries (option 2 from menu G9645) utility accesses this field to determine if the file copied will be copied with data.</td>
</tr>
<tr>
<td>Optional File</td>
<td>Indicates the file may be optional in your production environment. F8 provides a list of optional files.</td>
</tr>
<tr>
<td>Form-specific information</td>
<td>Designates if the file may not be needed at a client installation. The explanation of these situations can be found in the Generic Rate/Message information for that file for Generic Rate/Message Type 96/OF (F8 above). All of these files that exist in a specified library can be listed in the Optional File Report on menu G9645.</td>
</tr>
<tr>
<td>Common File</td>
<td>Indicates when a file should exist in the common library or user production library. The Create User Data Libraries (option 2 from menu G9645) utility accesses this field to determine if the file should be placed in a common library if specified, or the production library.</td>
</tr>
</tbody>
</table>

### Where Are Members Maintained?

The bottom half of the Software Versions Repository form lists the libraries in which the member is maintained. This information is stored in the Software Versions Repository Detail file (F9802).
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Library</td>
<td>The library containing the data to be accessed.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>The source library where the source file for the object is maintained.</td>
</tr>
<tr>
<td></td>
<td>This library is usually JDFSRC (for JD Edwards World) or CLTSRC (for the client) for production and DEVSRC for development.</td>
</tr>
<tr>
<td>Object Library</td>
<td>The library in your Development Environment to receive the compiled object.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>The destination library for the compiled object. This is for compile purposes only and no check is made to ensure that the object is in that library.</td>
</tr>
<tr>
<td></td>
<td>Leave the object library name blank for copy modules since they are not compiled objects.</td>
</tr>
<tr>
<td>Source File</td>
<td>The source file containing the source member.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>At JD Edwards World, three source files reside inside of the JDFSRC library.</td>
</tr>
<tr>
<td></td>
<td>They are</td>
</tr>
<tr>
<td></td>
<td>• JDECPY for copy modules,</td>
</tr>
<tr>
<td></td>
<td>• JDESRC for all other source code, and</td>
</tr>
<tr>
<td></td>
<td>• F98CRTCMD for precompiler commands.</td>
</tr>
</tbody>
</table>
Field | Explanation
--- | ---
SAR Number | An abbreviation for software action request (SAR).
  - If the CASE profile has "NONE" for SAR Number, 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 will be validated against the Work Order file, F4801.

Form-specific information
The JD Edwards World Development process uses the most recent Software Action Request (SAR)/ Work Order number associated with the member. This number must be valid, and if the status of the SAR number is 'complete', you should enter a new SAR to perform additional work on the member. A basic version of the Work Order system is sent to clients who have purchased the Computer Assisted Design (CAD) system and serves as a means for the client to keep track of their projects.

If an Employee Address Number is specified on the User Information screen (F0092 file), that number will be used to list SARs currently assigned to that person when F1 is pressed on this field.

The edit for this field is controlled by the SAR information entered in CASE Profiles.

Version ID | The software version number to be defaulted in the Software Versions Repository file.
  
Form-specific information
Identifies the JD Edwards World release level of the member in the designated environment.

Validated against User Defined Codes 98/RL.

S C (Status Code) | Determines the status of the software in the development process.

Development | Development Progress Code.
  
Indicates the progress of modifications made to the member.

User ID | The IBM-defined user profile.
  
Form-specific information
User ID that last modified the member (automatically updated).

Date Modified | The date the member was last updated (automatically updated).
Note: Each subfile line represents a record in the Software Versions Repository detail file (F9802). The information can be entered manually or through the upgrade process. There is no presence check for the object so if the object was deleted or the information was entered in error, there may be invalid entries.

Typing “D” in the Action Code deletes all the members and control data from the following:

- Software Versions Repository Master file (F9801)
- Software Versions Repository Detail file (F9802)
- Source and Object, if applicable
- Data Dictionary (F9200, F9203, F9816, F98163)
- Vocabulary Overrides (F9220)
- Function Key Definition (F9601,F9611)
- DREAM Writer forms (F98301, F9831, F98311, F98312)
- Cursor Sensitive Helps (F9620, F9621)
- Processing Options (F98302)
- Program Generator, if applicable

Naming Conventions

The following forms show how the report and CL program share the same description and base member as the program name. The same convention is true for the CL program and the special form.

<table>
<thead>
<tr>
<th>9801</th>
<th>Software Versions Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Code...</td>
<td>1</td>
</tr>
<tr>
<td>Member ID...</td>
<td>E4256X</td>
</tr>
<tr>
<td>Description...</td>
<td>Sales Order Invoices Print</td>
</tr>
<tr>
<td>Function Code...</td>
<td>ESP</td>
</tr>
<tr>
<td>Function Use...</td>
<td>164</td>
</tr>
<tr>
<td>System Code...</td>
<td>32</td>
</tr>
<tr>
<td>Reporting System...</td>
<td>32</td>
</tr>
<tr>
<td>Base Member Name...</td>
<td>ESOPF5C</td>
</tr>
<tr>
<td>Maint./EDIT...</td>
<td>Omit Option...</td>
</tr>
<tr>
<td>Copy Data (Y/N)...</td>
<td>N</td>
</tr>
</tbody>
</table>

DREAM Writer form exists

<table>
<thead>
<tr>
<th>O</th>
<th>Source</th>
<th>Object</th>
<th>Source</th>
<th>SAR</th>
<th>Version</th>
<th>SD</th>
<th>User</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Library</td>
<td>Library</td>
<td>File</td>
<td>Number</td>
<td>ID</td>
<td>F</td>
<td>ID</td>
<td>Modified</td>
</tr>
<tr>
<td></td>
<td>JRESC</td>
<td>JREDCJ</td>
<td>JRESC</td>
<td>68535E</td>
<td>303</td>
<td>1</td>
<td>UDR</td>
<td>11/12/93</td>
</tr>
</tbody>
</table>
The JD Edwards World naming structure identifies and describes major software components. The first character of the name indicates the type of component, such as program or data file. The second and third characters denote the system and are referred to extensively throughout the software. The fourth, fifth, and sixth characters represent the component group type, such as the function to be performed by the indicated component. The seventh through the tenth characters identify component versions. File names vary from four to eight characters in length, while all other component names are at least six characters long.
Use the following chart as your guide when naming objects.

<table>
<thead>
<tr>
<th>First digit — Component</th>
<th>X</th>
<th>XX</th>
<th>XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>C — Common subroutine</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I — Data structure; record formats</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J — OL program</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P — RPG program</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>R — Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S — Special form</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T — Temporary work files</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V — Video screen display file</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X — Scrub and Edit Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XF — Input/Output File Server</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>XS — Input only/Caching Server</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Second and third digits — System Code

<table>
<thead>
<tr>
<th>System Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>World Foundation Environment</td>
</tr>
<tr>
<td>01</td>
<td>Address Book</td>
</tr>
<tr>
<td>03</td>
<td>Accounts Receivable</td>
</tr>
<tr>
<td>55</td>
<td>Reserved for clients</td>
</tr>
</tbody>
</table>

Fourth, Fifth, and Sixth Digits — Group Type

<table>
<thead>
<tr>
<th>Group Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>000 to 099</td>
<td>File maintenance</td>
</tr>
<tr>
<td>100 to 199</td>
<td>Transaction processing</td>
</tr>
<tr>
<td>200 to 299</td>
<td>Inquiry only</td>
</tr>
<tr>
<td>300 to 399</td>
<td>Input registers and journals</td>
</tr>
<tr>
<td>400 to 499</td>
<td>Operating reports</td>
</tr>
<tr>
<td>500 to 599</td>
<td>Special purpose reports</td>
</tr>
<tr>
<td>600 to 799</td>
<td>Standard management reports</td>
</tr>
<tr>
<td>800 to 999</td>
<td>Housekeeping</td>
</tr>
<tr>
<td>DS</td>
<td>Data structure</td>
</tr>
<tr>
<td>Other</td>
<td>Window designations</td>
</tr>
</tbody>
</table>

A CL program, RPG program and a Report file may have identical names with different prefixes.

For example: J01051, P01051, R01051 (Address Book Revisions).
Use the following as a guide when naming files.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance program</td>
<td>The maintenance program for a file has the same name with a different prefix. For example, F9220 and P9220 or F9601 and P9601.</td>
</tr>
<tr>
<td>Logical files</td>
<td>For logical files over one physical, the logical file has the same name as the physical followed by an L, followed by A thru Z. For example, F0101 has logicals F0101LA, F0101LB, F0101LC, and F0101LD.</td>
</tr>
<tr>
<td>Join logical files</td>
<td>Join Logical files have the same name as the principal based-on file, a suffix of J followed by A thru Z. For example, the system names the join of F0006 and F0911 as F0006JA.</td>
</tr>
</tbody>
</table>
| Temporary files | Batch jobs use T files doing a CRTDUPOBJ. The job then removes the object after completion.  
  - Usually Physical Files  
  - Begin with T  
  - Found in JDFOBJ |
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic work files</td>
<td>Dynamic work files are usually FASTR processing requirements. Dynamic work files create and delete after the job is complete.</td>
</tr>
<tr>
<td></td>
<td>- Usually logical files</td>
</tr>
<tr>
<td></td>
<td>- Have same name as program</td>
</tr>
</tbody>
</table>

### The JD Edwards World System Codes

When used in menus, the system code follows the letter in the menu name. Shown below are the system codes for the standard AS/400 systems:

#### General System Codes

<table>
<thead>
<tr>
<th>System Number</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>General Back Office</td>
</tr>
<tr>
<td>01</td>
<td>Address Book</td>
</tr>
<tr>
<td>02</td>
<td>Electronic Mail</td>
</tr>
<tr>
<td>03</td>
<td>Accounts Receivable</td>
</tr>
<tr>
<td>03B</td>
<td>Enhanced Accounts Receivable</td>
</tr>
<tr>
<td>03C</td>
<td>Customer Issue Tracking</td>
</tr>
<tr>
<td>04</td>
<td>Accounts Payable</td>
</tr>
<tr>
<td>05</td>
<td>Standalone Time Accounting</td>
</tr>
<tr>
<td>06</td>
<td>Old Payroll</td>
</tr>
<tr>
<td>07</td>
<td>New Payroll</td>
</tr>
<tr>
<td>08</td>
<td>Human Resources</td>
</tr>
<tr>
<td>08A</td>
<td>Application Tracking</td>
</tr>
<tr>
<td>08B</td>
<td>Benefits Administration</td>
</tr>
<tr>
<td>08C</td>
<td>Canadian Specific HR Functions</td>
</tr>
<tr>
<td>08H</td>
<td>Health and Safety</td>
</tr>
<tr>
<td>08P</td>
<td>Position Control</td>
</tr>
<tr>
<td>08R</td>
<td>Requisitions</td>
</tr>
<tr>
<td>08U</td>
<td>US Specific HR Functions</td>
</tr>
<tr>
<td>System Number</td>
<td>System</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>08W</td>
<td>Wage and Salary</td>
</tr>
<tr>
<td>09</td>
<td>General Accounting</td>
</tr>
<tr>
<td>09E</td>
<td>Expense Reimbursement Reporting System Only</td>
</tr>
<tr>
<td>10</td>
<td>Financial Reporting</td>
</tr>
<tr>
<td>10C</td>
<td>Multi-Site Consolidations</td>
</tr>
<tr>
<td>11</td>
<td>Foreign Currency/ Cash Basis</td>
</tr>
<tr>
<td>11C</td>
<td>Cash Basis Accounting</td>
</tr>
<tr>
<td>12</td>
<td>Fixed Assets</td>
</tr>
<tr>
<td>13</td>
<td>Equipment/ Plant Management</td>
</tr>
<tr>
<td>14</td>
<td>Modeling, Planning, and Budgeting</td>
</tr>
<tr>
<td>15</td>
<td>Commercial Property Management</td>
</tr>
<tr>
<td>16</td>
<td>Residential Property Management</td>
</tr>
<tr>
<td>17</td>
<td>Customer Service Management</td>
</tr>
<tr>
<td>17A</td>
<td>Ariba Integration Reporting Purposes Only</td>
</tr>
<tr>
<td>17C</td>
<td>Call Management Reporting Purposes Only</td>
</tr>
<tr>
<td>18</td>
<td>Resource Scheduling Reporting Purposes Only</td>
</tr>
<tr>
<td>19</td>
<td>Utility CIS</td>
</tr>
<tr>
<td>20</td>
<td>Energy Data Base</td>
</tr>
<tr>
<td>21</td>
<td>Lease Management</td>
</tr>
<tr>
<td>22</td>
<td>Production System</td>
</tr>
<tr>
<td>23</td>
<td>Revenue Distribution</td>
</tr>
<tr>
<td>24</td>
<td>Contracts</td>
</tr>
<tr>
<td>25</td>
<td>Joint Interest Billing</td>
</tr>
<tr>
<td>26</td>
<td>Gas Balancing</td>
</tr>
<tr>
<td>27</td>
<td>Investor Services</td>
</tr>
<tr>
<td>28</td>
<td>Projects on Hold – Energy</td>
</tr>
<tr>
<td>29</td>
<td>AFE Accounting</td>
</tr>
<tr>
<td>30</td>
<td>Product Data Management</td>
</tr>
<tr>
<td>30A</td>
<td>Product Costing Reporting Only</td>
</tr>
<tr>
<td>System Number</td>
<td>System</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>31</td>
<td>Shop Floor Control</td>
</tr>
<tr>
<td>31A</td>
<td>Manufacturing Accounting</td>
</tr>
<tr>
<td>3110</td>
<td>Process Control</td>
</tr>
<tr>
<td>32</td>
<td>Configuration Processing</td>
</tr>
<tr>
<td>32C</td>
<td>Custom Works</td>
</tr>
<tr>
<td>33</td>
<td>Capacity Requirements Planning</td>
</tr>
<tr>
<td>34</td>
<td>DRP/ MPS/ MRP</td>
</tr>
<tr>
<td>34A</td>
<td>Advanced Planning &amp; Scheduling</td>
</tr>
<tr>
<td>35</td>
<td>Enterprise Facility Planning</td>
</tr>
<tr>
<td>36</td>
<td>Forecasting</td>
</tr>
<tr>
<td>37</td>
<td>Quality Management</td>
</tr>
<tr>
<td>38</td>
<td>Distribution Contracts</td>
</tr>
<tr>
<td>39</td>
<td>Bulk Stock Control</td>
</tr>
<tr>
<td>40</td>
<td>Inventory/ OP Base</td>
</tr>
<tr>
<td>4010</td>
<td>Advanced Price Adjustments</td>
</tr>
<tr>
<td>41</td>
<td>Inventory Management</td>
</tr>
<tr>
<td>41B</td>
<td>Bulk Stock Control</td>
</tr>
<tr>
<td>42</td>
<td>Sales Order Processing</td>
</tr>
<tr>
<td>42A</td>
<td>Sales Force Automation Reporting Purposes Only</td>
</tr>
<tr>
<td>42B</td>
<td>Enterprise One Sales Order Entry</td>
</tr>
<tr>
<td>42E</td>
<td>ECS Sales Order Processing</td>
</tr>
<tr>
<td>43</td>
<td>Purchase Order Processing</td>
</tr>
<tr>
<td>44</td>
<td>Contract Administration</td>
</tr>
<tr>
<td>44H</td>
<td>Homebuilder Management</td>
</tr>
<tr>
<td>4401</td>
<td>Homebuilder Management</td>
</tr>
<tr>
<td>45</td>
<td>Advanced Price Analysis</td>
</tr>
<tr>
<td>46</td>
<td>Warehouse Management</td>
</tr>
<tr>
<td>47</td>
<td>Electronic Data Interchange</td>
</tr>
<tr>
<td>48</td>
<td>Work Order Processing</td>
</tr>
<tr>
<td>System Number</td>
<td>System</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>48S</td>
<td>Service Billing</td>
</tr>
<tr>
<td>49</td>
<td>Transportation</td>
</tr>
<tr>
<td>50</td>
<td>Job Cost Accounting</td>
</tr>
<tr>
<td>51</td>
<td>Job Cost Accounting</td>
</tr>
<tr>
<td>52</td>
<td>Job Cost Billing</td>
</tr>
<tr>
<td>53</td>
<td>Change Management</td>
</tr>
<tr>
<td>55-59</td>
<td>Reserved for Clients</td>
</tr>
<tr>
<td>60-69</td>
<td>Reserved for JD Edwards World Custom</td>
</tr>
<tr>
<td>70</td>
<td>Multi-National Products</td>
</tr>
<tr>
<td>71</td>
<td>Client Server Applications</td>
</tr>
<tr>
<td>73</td>
<td>M&amp;D Complimentary Products</td>
</tr>
<tr>
<td>74</td>
<td>EMEA Localizations</td>
</tr>
<tr>
<td>74G</td>
<td>Greece</td>
</tr>
<tr>
<td>74H</td>
<td>Hungary</td>
</tr>
<tr>
<td>74I</td>
<td>Ireland</td>
</tr>
<tr>
<td>74L</td>
<td>Portugal</td>
</tr>
<tr>
<td>74N</td>
<td>Nordics</td>
</tr>
<tr>
<td>74P</td>
<td>Poland</td>
</tr>
<tr>
<td>74R</td>
<td>CIBS</td>
</tr>
<tr>
<td>74S</td>
<td>Spain</td>
</tr>
<tr>
<td>74T</td>
<td>Turkey</td>
</tr>
<tr>
<td>74Z</td>
<td>Czech Republic</td>
</tr>
<tr>
<td>75</td>
<td>Asia Pacific Localizations</td>
</tr>
<tr>
<td>75H</td>
<td>Thailand</td>
</tr>
<tr>
<td>75I</td>
<td>India</td>
</tr>
<tr>
<td>75K</td>
<td>South Korea</td>
</tr>
<tr>
<td>77</td>
<td>Taiwan</td>
</tr>
<tr>
<td>76</td>
<td>Latin American Localization</td>
</tr>
<tr>
<td>76A</td>
<td>Argentinean Localization</td>
</tr>
<tr>
<td>System Number</td>
<td>System</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>76C</td>
<td>Columbia</td>
</tr>
<tr>
<td>76H</td>
<td>Chile</td>
</tr>
<tr>
<td>77</td>
<td>Canadian Payroll</td>
</tr>
<tr>
<td>77Y</td>
<td>Canadian Payroll Year End Programs Reporting System Only</td>
</tr>
<tr>
<td>78</td>
<td>OBSOLETE CS Travel Expense Management</td>
</tr>
<tr>
<td>79</td>
<td>OBSOLETE Foreign Translation</td>
</tr>
</tbody>
</table>

Technical Foundation Systems

<table>
<thead>
<tr>
<th>System Number</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>General Back Office</td>
</tr>
<tr>
<td>80</td>
<td>Business Intelligence</td>
</tr>
<tr>
<td>81</td>
<td>DREAM Writer</td>
</tr>
<tr>
<td>82</td>
<td>World Writer</td>
</tr>
<tr>
<td>83</td>
<td>Finance Report Writer-FASTR</td>
</tr>
<tr>
<td>84</td>
<td>Distributed Data Processing</td>
</tr>
<tr>
<td>85</td>
<td>Custom Programming</td>
</tr>
<tr>
<td>86</td>
<td>Foreign Language Translation</td>
</tr>
<tr>
<td>87</td>
<td>JD Edwards World Internal</td>
</tr>
<tr>
<td>88</td>
<td>Cautious Purge System</td>
</tr>
<tr>
<td>89</td>
<td>Conversion Programs</td>
</tr>
<tr>
<td>91</td>
<td>Documentation</td>
</tr>
<tr>
<td>92</td>
<td>Computer Assisted Design</td>
</tr>
<tr>
<td>93</td>
<td>Computer Assisted Programming</td>
</tr>
<tr>
<td>94</td>
<td>Security Officer</td>
</tr>
<tr>
<td>95</td>
<td>Unattended Night Operations</td>
</tr>
<tr>
<td>96</td>
<td>Computer Operations</td>
</tr>
<tr>
<td>97</td>
<td>Software Load &amp; Install</td>
</tr>
<tr>
<td>98</td>
<td>Technical Aids</td>
</tr>
</tbody>
</table>
Examples of Program and File Names

The following is a detailed breakdown of program and file names. All of the file types have the same system code and component group.

**Data Files**

Account Master File

Component (File)  F  09  01
System Code (General Accounting)  
Component Group Type (Master)  

Account Master Alternate Logical

Component (File)  F  09  01  LA
System Code (General Accounting)  
Component Group Type (Master)  
Version Identification (Logical)  

**Videos (Screens)**

Component (Video)  V  09  01
System Code (General Accounting)  
Component Group Type (File Maintenance)  

**RPG Programs**

Component (RPG Program)  P  09  01
System Code (General Accounting)  
Component Group Type (File Maintenance)  

---

Optional Files Workbench

The Optional Files Workbench provides better access to optional files. When you delete optional files, they are logged. If you reinstall, those files are not put back into the system. Each file has an explanation about the circumstances that makes it optional. The Option Files Workbench is accessed via option 18 from menu G9645.

**Note:** If you need the deleted files, you can remove them from the logged optional files and copy them from JDFDATA.

---

**Logical Files**

- The Member ID for logical files ends with Lx, where x is the next available letter in alpha sequence.
- The Object Library is usually JDFDATA.
- The Description should list the key fields for the view.
- The Maint/RSTDSP is 1 for permanent system logicals.
- The Base Member Name is the physical file the logical view is over.
- If you are adding a custom logical, start at the bottom of the sequence for example L99. If JD Edwards World add a logical, it will be sequenced after the previous one they created.
Join Logical Files

- The Description lists the files over which the join is built.
- The Base Member Name is the primary file in the join.
- If the Physical files exist in separate libraries, the CRTLF command must be used since the SVR process can only create a logical if the joined files are in the same library.

Copy Modules

- The Member ID begins with C, D, E, or I.
- The Source File is JDECPY.
- The Description describes the function of the module.
- The Function Code is COPY or CPYL (RPG IV code).
Windows

- The Member ID begins with V, the system number, then an alphabetic identifier as shown in the example below.

- The Description describes the function of the form.

- Maint/ RSTDSP is left blank to allow the form to appear in front of text from the calling form.

Navigation Functions

The following Function keys facilitate navigation within the Software Versions Repository.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F6</td>
<td>Access Repository Services</td>
</tr>
</tbody>
</table>

You can access the Repository Services form using F6. This form provides access to the other repository services within JD Edwards World.
Sample F6 - Repository Services

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| F9       | Automatic Reinquiry  
          | Once the system has accepted the changes you made to a member and cleared the screen, you can automatically inquire on that member by pressing F9. |
| F17      | Position Cursor to Action Code  
          | When you inquire on a member, the system positions the cursor in the subfile for the screen. To reposition your cursor in the Action Code field, press F17. |
| F19      | Previous Member  
          | When you inquire on a member, the system positions the cursor in the subfile for the screen. To reposition your cursor in the Action Code field, press F17. |
| F20      | Next Member  
          | To access the member stored after the currently displayed member, press F20. |
Other Function Keys

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>Automatic Reinquiry</td>
</tr>
<tr>
<td></td>
<td>- To access a command line to enter a JD Edwards World or IBM command without having to exit to Command Entry or a menu.</td>
</tr>
<tr>
<td></td>
<td>- Calls a JD Edwards World program and not the IBM Command Entry.</td>
</tr>
<tr>
<td></td>
<td>If you are secured out of Command Entry or Menu Traveling, you still receive this command line but you cannot execute commands or menu travel.</td>
</tr>
<tr>
<td>F8</td>
<td>Optional Files</td>
</tr>
<tr>
<td></td>
<td>The system displays the optional files.</td>
</tr>
<tr>
<td>F10</td>
<td>Checklists</td>
</tr>
<tr>
<td></td>
<td>Displays a user defined checklist. Opt 1 displays additional job information.</td>
</tr>
<tr>
<td>F13</td>
<td>Displays additional category code information for each member. You can cross-reference category code values to the Software Versions Search program (23/G91).</td>
</tr>
<tr>
<td>F14</td>
<td>Member Parameter/Key List</td>
</tr>
<tr>
<td></td>
<td>Identifies the access path for keyed files.</td>
</tr>
<tr>
<td>F15</td>
<td>You can access the Where Used facility using Function key 15 on Software Versions Repository. Use this facility to determine every location that a particular member is used.</td>
</tr>
</tbody>
</table>

Below is an example form displaying every program that uses the Business Unit Master screen:

Use an * in Type on the Cross Reference screen to see the possible combinations.

```
Object: Name . . . W0006       Business Unit Master Revisions - Single
Type . . . E               All programs using file
Punct OD . ___

<table>
<thead>
<tr>
<th>O</th>
<th>Name</th>
<th>Description</th>
<th>Field Attr T Start Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>W0006</td>
<td>Business Unit Master Revisions - Single</td>
<td>___ Len_ Dec_ Y Loc_ X/N</td>
</tr>
</tbody>
</table>

Opt: 1=SVR 2=Create Object 3=Field Explanation 42=Print 41=Regenerate
```
Note: To use this facility, you must run the Cross Reference Rebuild.
(6/ G9642)

F23 - Flow Programs / Illustrate File Models

F23 - To display a flowchart if the member is a program or a Data Model if the member is a file.

This is only functional for programs and files.

Selection Exits from the Software Versions Repository

The following is a list of the options available from the Software Versions Repository. By referring to the form pictured in the beginning of the chapter, you can see that there are more options than can be displayed on the form.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Browse</td>
</tr>
<tr>
<td></td>
<td>Displays the source member in SEU browse mode.</td>
</tr>
<tr>
<td>2</td>
<td>Edit</td>
</tr>
<tr>
<td></td>
<td>Displays the source member in SEU update mode.</td>
</tr>
<tr>
<td>3</td>
<td>Copy the source member</td>
</tr>
<tr>
<td></td>
<td>Copies the source member to another source file member or copies the member to a new member name in the same file.</td>
</tr>
<tr>
<td></td>
<td>Adds SVR master and detail records if needed. Copies pre-compiler commands and Vocabulary Overrides. Copies program generator specifications if requested.</td>
</tr>
<tr>
<td>5</td>
<td>Work with SAR detail</td>
</tr>
<tr>
<td></td>
<td>Displays the SAR/ Work Order Detail Entry screen, defaulting to the members affected portion of the SAR/ Work Order.</td>
</tr>
<tr>
<td>8</td>
<td>Print source</td>
</tr>
<tr>
<td></td>
<td>Prints the member.</td>
</tr>
<tr>
<td>9</td>
<td>Delete/ remove source</td>
</tr>
<tr>
<td></td>
<td>Deletes the detail record and removes the source member from the source file.</td>
</tr>
<tr>
<td></td>
<td>The same IBM authority that applies to the command RMVM applies to this function.</td>
</tr>
<tr>
<td></td>
<td>Will delete the object if requested by the user. If you do not remove the source member, you will not be allowed to delete the object.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
</tbody>
</table>
| 10     | Exit to design aid  
Determines what type of member you are accessing and then exits to the correct JD Edwards World design tool; that is, SDA/ RDA/ FDA/ Program Generator. |
| 11     | Precompiler commands for JD Edwards World compiler.  
Accesses the source code for the precompiler commands associated with a program.  
A highlighted message, Precompiler Commands Exist, indicates there is a member with the same name in the F98CRTCMD file.  
Contains information the compiler will use when compiling the program.  
**Note:** Only one person can view the same pre-compiler commands. |
| 14     | Submit object creation  
Compiles the member and generates an object. The type of object determines what library the compiled object will be placed into. |
| 15     | Generate program source and help  
Submits the member to the program generator in order to generate source and related helps.  
Only applicable to CASE users. |
| 17     | Edit help instructions  
Accesses the help instructions for a particular program in update mode utilizing SEU. |
| 18     | Generate & rebuild help instructions  
Submits the helps for generation and rebuilds them into their final form once they have been entered. |
| 20     | Browse SDA/ RDA  
Accesses SDA or RDA in browse mode with Vocabulary Overrides. |
| 21     | Print help  
Prints the help instructions for the member. |
| 25     | Print illustration  
Prints an illustration of printer files, display files, or data base files. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>Source modifications editor&lt;br&gt;Allows you to view the source modifications made to the member through SEU after source was generated. Stored in the F93002 file.&lt;br&gt;Only applicable to CASE users.</td>
</tr>
</tbody>
</table>
CASE Profiles

About CASE Profiles

CASE profiles are user defined values that can pertain to individual users or to one *PUBLIC user profile. The profiles are stored in F98009. Parameters in these profiles are used when copying source members or when designing or compiling programs and files.

The following processing control parameters are defined by the user:

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

**Note:** You should immediately update the record for User ID *PUBLIC to reflect JOBQs, OUTQa and Libraries on your machine.

When entering information for *PUBLIC, all fields are required.

Default CASE Profile values are maintained in a record with the User ID *PUBLIC. You should enter CASE Profile values for individual users only if you need overrides to the *PUBLIC values.

When entering values for individual users, you can leave all fields blank except for the specific values being overridden.

This section contains the following:

- Accessing CASE Profiles
- Function Key Exits from the CASE Profiles Program
- Summary of CASE Profiles

Accessing CASE Profiles

**To access CASE Profiles**

To access CASE Profiles, choose one of the following methods:

1. From Computer Assisted Design (CAD) (G92), choose CASE Profiles.
2. From Run Time Setup (G90), choose Software Version Repository, press F6 to open up the Repository Services form, select CASE Profiles.

The CASE Profile form displays. The program attempts to automatically inquire on your User ID. If your ID is not set up, an error occurs. You can inquire on *PUBLIC.
Default Development Environment

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source File Name</td>
<td>The default source file name where source is to be stored within the source library. Must reside within the source library specified.</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. The value specified must reside within the source library specified.</td>
</tr>
<tr>
<td>Data File Library</td>
<td>Used to specify the test (or development) library for physical and logical files. Used as the default object library for the Software Versions Repository when copying source code for physical or logical files.</td>
</tr>
<tr>
<td>SAR Number</td>
<td>An abbreviation for software action request (SAR).</td>
</tr>
<tr>
<td></td>
<td>*NONE = the SAR number will not be validated in any of the CAD/CAP programs and can be left blank.</td>
</tr>
<tr>
<td></td>
<td>If a SAR number is entered, it is used in conjunction with the SAR Delivery Type of *DFT (default).</td>
</tr>
<tr>
<td>Version ID</td>
<td>The software version number to be defaulted in the Software Versions Repository file.</td>
</tr>
<tr>
<td>Status Code</td>
<td>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.</td>
</tr>
</tbody>
</table>

Caution: If you create a custom environment, put 2, 3, or 4 in user defined codes. If you have a ‘1’ (production), the system will think it is a JD Edwards World file and write over it during the Software Version Repository Merge in an upgrade.

Program Creation Options

You have the following options when you create a program.
### 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, RPGL, CBL, PLI, C and SYSC.

### Prog Gen Job Queue

Specifies which job queue will be used when submitting jobs from the Program Generator.

These jobs include the source code generation and the source code monitor from SEU.

### Compile Target Release

Used by various AS/400 compilers (RPG, RPGL, CLP, COBOL, C) to compile an object compatible with a specified target release.

- A value of *CURRENT compiles an object compatible to the release of the machine at compile time.
- A value of *PRV compiles an object compatible with both one release back and the current release.

### Print Option

Used to designate whether or not a report will be generated when an object is compiled.

- 0 no print
- 1 print
- 2 print and hold spool file
- 3 print only, does not generate execution object (applies to COBOL and RPG only)
- 4 print when compile or creation fails

### Cross-Reference Listing

Specifies whether a cross-reference listing will be generated for variables and fields in a program’s compile listing.

---

**Note:** The default job queue for compiles is called COMPIL. If you do not have a COMPIL JOBQ, enter the JOBQ name for compiles on the *PUBLIC CASE profile.

---

### SAR Options

The following fields provide you with options for the location of your SAR file and SAR logging.

### SAR File Library

Specifies the library where the Software Action Request (SAR) file for software development exists.

- If left blank, the user’s library list will be used.
- You can specify *NONE in the SAR number field if you do not want any SAR number editing.
**SAR Delivery Type**

Associated with SAR logging. SAR logging is a feature which tracks all activities related to modifying JD Edwards World™ software.

- **NONE** = no logging.
- **LOG** = log to SAR number 00000000 (no SAR number is used for logging).
- **DFT** = log to a default SAR number (specified in the SAR Number field).
- **PROMPT** = log and prompt the user for the SAR number to be used and allow the user to enter the revision notes.

**Miscellaneous**

The following fields are reserved for future use.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source Gen Opt (Future)</td>
<td>For future use.</td>
</tr>
<tr>
<td>Helps Maint Opt(Future)</td>
<td>For future use.</td>
</tr>
</tbody>
</table>

**Function Key Exits from the CASE Profiles Program**

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F6</td>
<td>Access Repository Services</td>
</tr>
<tr>
<td></td>
<td>This key will display a form that provides access to the other repository services including CASE profiles.</td>
</tr>
<tr>
<td>F9</td>
<td>Previous Member</td>
</tr>
<tr>
<td></td>
<td>Allows you to re-inquire on the last record viewed.</td>
</tr>
</tbody>
</table>

**Summary of CASE Profiles**

- The CASE Profiles file is F98009.
- You need to update the *PUBLIC record as well as add any additional individual records desired.
- You cannot delete the *PUBLIC record.
- When entering information for the *PUBLIC record, all fields are required.
The record for User ID *PUBLIC contains the values that are used as the defaults for all users unless individual user profiles have been set up.

When entering values for individual profiles, all fields are left blank EXCEPT for the specific values being overridden on the *PUBLIC profile.

SAR Number and SAR Delivery type work together to determine what type of SAR logging should occur.

- *NONE = no SAR logging at all.
- *LOG = no SAR number is included as part of the SAR logging.
- *DFT = the SAR number specified is used for the SAR logging.
- *PROMPT = you are prompted for a SAR number and revision notes when an entry is about to be made to the SAR log.
Working with SAR Log

About SAR Log

After you create the SARs, you must activate SAR logging, which automatically tracks the SARs as you develop the software.

The SAR Log Inquiry program allows you to review information in the SAR Log file (F9810).

You can also change the SAR Number and Revision Notes for individual log records.

Complete the following tasks:

- Setting Up User Input Options for SAR Logging
- Selecting Types of SAR Information to Log
- Accessing SAR Log Inquiry
- Summary of the SAR Log Inquiry

Before You Begin

- Create SARs before you activate SAR logging.

From the Version Control menu (G9261), choose CASE Profiles.
Setting Up User Input Options for SAR Logging

To set up user input options for SAR logging

On CASE Profiles

1. Complete the following fields:
   - SAR File Library
   - SAR Delivery Type

   **Caution:** The SAR file library contains the Work Order system files (F4801 and F4802). If you use the Work Order application, or if the F4802 file has different record types than what version control needs, you must create a library that contains new F4801 and F4802 files for version control purposes only. Specify this new library name in the SAR File Library field.

   **Note:** If you set the SAR Delivery Type field to *PROMPT, the Maintain User Default SAR Information form displays whenever you change a source code member or control table.

2. Complete the following optional field:

   If you provided a SAR number on CASE Profiles, it displays on this form. If you did not provide a SAR number, provide one on this form.

   If the Transfer field on Maintain User Default SAR Information is set to 1, a record will be created for the Version Control system. If it is set to 0, there will not be a record for the Version Control System.
SAR Number

What You Should Know About

SAR number and delivery type combinations

The information you provide for the SAR Number and SAR Delivery Type fields affects how the system handles SAR logging.

If you do not provide a SAR number, and set the SAR Delivery Type field to *PROMPT, the Maintain User Default SAR Information form prompts you for the SAR number whenever you change a source code member or control table.

If you provide a SAR number, and set the SAR Delivery Type field to *DFT, the system creates SAR log entries automatically without your input.

If you provide a SAR number, and set the SAR Delivery Type field to *PROMPT, the Maintain User Default SAR Information form prompts you to change the SAR number, if necessary, whenever you change a source code member or control table.

Invalid SAR delivery types

*LOG and *NONE are not valid for the SAR Delivery Type field when you use the Version Control system.

If you set the SAR Delivery Type field to *PROMPT, the Maintain User Default SAR Information form displays whenever you change a source code member or control table.

Selecting Types of SAR Information to Log

To select types of SAR information to log

In addition to setting up user input options for SAR logging, select the types of SAR information you want to log.

1. From the Version Control menu (G9261), access the processing options for Edit and Promote.

2. Make the following changes:
### Accessing SAR Log Inquiry

The SAR Log Inquiry includes several functions:

- Inquiry by user ID or SAR number with date range
- Exit to a maintenance program for the record type
- Exit to SAR detail
- Print option that allows for DREAM Writer selection

There are two ways to access the SAR Log Inquiry.

### To access the SAR Log Inquiry

1. To access the SAR Log Inquiry, select one of the following methods:
   - Choose SAR Inquiry from Menu G9362
   - Choose SAR Log Inquiry from the Repository Services form (F6 in SVR)
Working with SAR Log

1. The new SAR Log Inquiry form displays.

2. Complete one or more of the following fields:
   - User ID
   - SAR number
   - Date range

   Records matching the search criteria are displayed.
Field | Explanation
--- | ---
AC (Action) | The action that was taken on this record. The standard action code values apply.
Ty (Record Type) | The type of record that was updated. Use F1 to display all valid record types stored in User Defined Code 98/RT.
Item | The identification number (program number, file number, report number) assigned to any element of the software. These items are the members that reside in the Software Versions Repository or other repositories such as the Data Dictionary, Vocabulary Overrides, and the like.
SAR Number | The SAR number under which this change was made. This field can be updated on this video.
Revision Note | A user defined description field to further clarify the change made. This field can be updated on this video.
Time | The time at which the change was made.
Date | The date on which the change was made.
User | The user who made the change.

Selection Exits from the SAR Log Inquiry

The following is a list of selection exits from the SAR Log Inquiry form and an explanation of the effects of each selection.

Exit | Explanation
--- | ---
2 - Edit | Allows for maintenance of the record type. What program is accessed is based on the record type. For example, if the record type is ‘DD’, this exit will take the user to the Data Dictionary program.
5 - Work SAR | Exits to the SAR associated with the SAR log entry.
9 - Delete | Allows the user to delete entries from the SAR log.

Note: If you entered this program from the Repository Services form from the Software Versions Repository program, selection exit 2 does not function with record types “SV” or “PG” as these record types attempt to call the Software Versions Repository, which causes a recursive call error.
Function Key Exits from the SAR Log Inquiry

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F5</td>
<td>ASI Entry</td>
</tr>
<tr>
<td></td>
<td>Exits to Application Specific Instructions form for use during a software upgrade. You need the F0098 file to do this.</td>
</tr>
<tr>
<td>F6</td>
<td>Access Repository Services</td>
</tr>
<tr>
<td></td>
<td>Pressing this key displays a form that provides access to the other repository services, except for SAR Log Inquiry.</td>
</tr>
<tr>
<td>F21</td>
<td>Print</td>
</tr>
<tr>
<td></td>
<td>Allows you to print a SAR log report.</td>
</tr>
<tr>
<td></td>
<td>Exits to a DREAM Writer versions list.</td>
</tr>
</tbody>
</table>

Summary of the SAR Log Inquiry

The SAR Log Inquiry has the following features and restrictions:

- Uses the file F9810.
- If you so not want to use the SAR Logging feature at all, specify *NONE in the SAR Delivery Type field for all CASE Profile records.
- To use the SAR Logging feature, you must specify a value of *LOG, *DFT, or *PROMPT in the SAR Delivery Type field for all CASE Profile records.
- The SAR Logging feature records any changes that you make to the Data Dictionary, Vocabulary Overrides, User Defined Codes, and so forth.
- The SAR Log Inquiry program allows you to see what changes you make to any of the above.
- The SAR Log Inquiry program has Function Keys and Selection Exits which allow you to change the SAR Log records in the SAR Log file (F9810) or to exit to the maintenance program for the change you made.

For example, exit to the Data Dictionary program if the record indicates a Data Dictionary item was added or updated.
Work with Promotion Paths and Projects

Working with Promotion Paths and Projects

A promotion path defines how a project’s source code members and control file data will move from one environment to another. An environment consists of source code members and control file data. For source code members, the environment consists of:

- A source file
- A source library
- An object library

For control file data, the environment consists of a data library.

Perform the following tasks:

- Understanding Promotion Paths
- Defining a Promotion Path
- Defining a Project

Understanding Promotion Paths

A promotion path specifies the current locations of source code members and control file data and where they will be moved. For example, promoting a project’s source code members and control file data from a development environment to a test environment could look similar to the following illustrations.
9BWork with Promotion Paths and Projects

Each move between two environments requires that you define a unique promotion path.

A project is a collection of software and data you want to group together for promotion. A project is defined by the following characteristics:

- SARs that are associated with the project
- Promotion paths that determine the movement of the project software and data between environments
- Other projects that are attached to the project

<table>
<thead>
<tr>
<th>Source Code Members</th>
<th>Development Environment</th>
<th>Test Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLP</td>
<td>JDELSRC</td>
<td>JDELSRC</td>
</tr>
<tr>
<td>RPG</td>
<td>JDELSRC</td>
<td>JDELSRC</td>
</tr>
<tr>
<td>PF</td>
<td>JDELSRC</td>
<td>JDELSRC</td>
</tr>
<tr>
<td></td>
<td>DEVSRCS, DEVOBJ</td>
<td>TSTSRCS, TSTOBJ</td>
</tr>
<tr>
<td></td>
<td>DEVDTA</td>
<td>TSTDTA</td>
</tr>
</tbody>
</table>

Each move between two environments requires that you define a unique promotion path.
Before You Begin Defining a Promotion Path

- Verify that the SARs and promotion paths you want to associate with a project have been set up.
- The SAR system uses the Work Order files (F4801 and F4802). If your production environment uses these files, and if the F4802 file has different record types than what version control needs, define a separate library that contains these files for version control purposes only.

Defining a Promotion Path

Several steps are involved in defining promotion paths. Complete the following tasks:

- Locate a promotion path
- Add a promotion path
- Define a promotion path for source code members
- Define a promotion path for control tables

From the Version Control menu (G9261), select Manage Promotion Paths.
9BWork with Promotion Paths and Projects

To locate a promotion path

Select one of the following methods to locate a promotion path:

1. On a blank Manage Promotion Paths form, press Enter.
   The screen displays a complete list of promotion paths.

2. On Manage Promotion Paths, enter the path name in the Promotion Path field.
   The screen displays the path name. If the promotion path does not exist, the
   screen displays the path name that is closest alphabetically.

To add a promotion path

2. Complete the Promotion Path form:
   - Add a new path name, a path description, and a release level.
   - Use the Code 1 through 5 fields for additional classifications.
     - Code 1 through 5 fields are user defined in system 92, types E1, E2, E3, E4, and E5.
   - Specify the status of the promotion path in the Code 4 field.
     - Field-sensitive help (function key F1) provides valid values for the Code 4 and 5 fields.
   - Specify the type of promotion environment in the Code 5 field.

To define a promotion path for source code members

1. Locate Promotion Path Members using one of the following methods:
   - On the Manage Promotion Paths form, locate the promotion path you want to define.
   - Enter 2 (Members) in the OP (Option) field next to the promotion path name.
   - On the Promotion Path form, press F10 (Members).

The From Environment area on the Promotion Path Members form shows the current locations of the source and object code. The To Environment area shows the locations to which the code will be moved.

2. Specify source files and library names for each member type you list on this screen.
   - To display valid member types and their descriptions, press F1 while your cursor is in a Mbr Type field. The member types are defined in the Function Codes user defined code table (98/FN).
   - To copy source file and library names from one member type to another, type 1 (Copy) in the OP (Option) field next to the member type you want to copy. Type 2 (Target) in the OP fields next to the member types you want the information copied to, and press Enter. You can specify multiple targets.
The following chart shows some of the function keys available on this screen.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F9</td>
<td>Redisplays the record for the previously-changed path.</td>
</tr>
<tr>
<td>F11</td>
<td>Displays the Promotion Path Control Files form.</td>
</tr>
<tr>
<td>F13</td>
<td>Displays the CASE Profiles form.</td>
</tr>
<tr>
<td>F14</td>
<td>Retrieves the source file, source library, and object library from your CASE profile and fills in the From environment. This overwrites any information currently in the fields.</td>
</tr>
<tr>
<td>F15</td>
<td>Duplicates the source file and library names from the first member type to the remaining member types.</td>
</tr>
</tbody>
</table>
What You Should Know About

Copying an existing promotion path

If you copy an existing promotion path to create a new path, be sure that the source files and library names for the members are correct for the new path.

Changing library names

To change library names, enter the new library names over the current ones.

To define a promotion path for control files

1. Locate the Promotion Path Control Files using one of the following methods:
   - On Manage Promotion Paths (P92403), locate the promotion path you want to define and enter 3 (Ctl Files) in the OP (Option) field next to the promotion path name.
   - On the Promotion Path (F5 from P92403) form, press F11 (Ctl Files).
   - On the Promotion Path Members form (Opt 2 from P92403), press F11 (Ctrl Files).

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F9</td>
<td>Redisplays the record for the previously changed project.</td>
</tr>
<tr>
<td>F10</td>
<td>Displays the Promotion Path Members Files form.</td>
</tr>
<tr>
<td>F13</td>
<td>Displays the CASE Profiles form.</td>
</tr>
</tbody>
</table>
9B Work with Promotion Paths and Projects

F14 Retrieves the data library from your CASE profile and fills in the first From Data Libr field. This overwrites any information currently in the field.

F15 Duplicates the library names from the first record type to the remaining record types.

3. To copy library names from one record type to another, type 1 (Copy) in the OP (Option) field next to the record type you want to copy.

4. Type 2 (Target) in the OP fields next to the record types you want the information copied to, and press Enter. You can specify multiple targets.

Guidelines

- If you copy an existing promotion path to create a new path, be sure the library names for the control files are correct for the new path.
- To change library names, enter the new library names over the current ones.

Defining a Project

To define a project, complete the following tasks:

- Locate a project
- Add a project
- Assign promotion paths
- Assign project SARs

From the Version Control menu (G9261), choose Manage Projects.

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
<th>Client</th>
<th>Orig</th>
<th>Assigned</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>REINSTALL</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>TECH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>TECHRH</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>UPGRADE</td>
<td>256006</td>
<td>878411</td>
<td>878411</td>
</tr>
<tr>
<td>5</td>
<td>Version Control</td>
<td>875561</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Merge PO Display Level</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>4 BYTE STS</td>
<td>878411</td>
<td></td>
<td>2211696</td>
</tr>
</tbody>
</table>

Opt: 1=Details  2=Paths  3=SARs  F5=Add Project  F24=More Keys
9BWork with Promotion Paths and Projects

To locate a project

Locate a project using one of the following methods:

1. On a blank Manage Projects form, press Enter.
   - A complete list of projects displays.

2. On the Manage Projects form, enter the project name in the Project field.
   - The project name displays on the form. If the project does not exist, the project name that is closest alphabetically displays on the form.

To add a project

1. On Manage Projects, choose Add Project.

2. On the Software Development Project form, do the following:
   - Enter a new project name.
   - Enter a project description.
   - Enter any other information you want to associate with the project.
   - Complete the Code 1 through 5 fields for additional classifications.
     - The Code 1 through 5 fields are user defined in system 92, types P1, P2, P3, P4, and P5.

3. If you want to attach this project to a parent project, specify the parent project name in the Parent Project field.
The following chart shows some of the function keys available on this form.

<table>
<thead>
<tr>
<th>Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F9</td>
<td>Redisplays the record for the previously-changed project.</td>
</tr>
<tr>
<td>F10</td>
<td>Displays the Project Promotion Path form.</td>
</tr>
<tr>
<td>F11</td>
<td>Displays the Project Elements form.</td>
</tr>
<tr>
<td>F14</td>
<td>Displays the generic text associated with this project, and gives you access to text model selections.</td>
</tr>
</tbody>
</table>

You must assign promotion paths and SARs to the project you set up here. The following sections explain how to assign them.

**To assign promotion paths**

1. Locate the project to which you want to assign promotion paths using one of the following methods.
   - On Manage Projects selection (Option 4 on menu G9261), locate the project to which you want to assign promotion paths.
   - In the OP (Option) field next to the project name, enter 2 (Paths)
   - On Software Development Project (Manage projects, F5), press F10 (Promotion Paths).

2. Specify the promotion paths you want to assign to this project.
   - To display the available promotion paths, press F1 (Help) while the cursor is in a Promotion Path field.
   - Some of the function keys available on this form include:
     - F9 - Redisplays the record for the previously-changed project.
     - F11 - Displays the Project Elements form.
   - Options available on this form include:
     - 1 - Edit the promotion path details.
To assign project SARs

Project SARs are one element of a project; other projects can also be elements of a project.

1. Access the Project Elements form using one of the following methods:
   - On the Version Control menu (G9261), choose Edit and Promote.
   - On the Manage Projects form, locate the project to which you want to assign elements.
     - Enter 3 (SARs) in the OP (Option) field next to the project name.
   - On the Software Development Project form, press F11 (Project SARs).
   - On the Project Promotion Paths form, press F11 (Project SARs).

The Project Elements form displays the elements (usually SARs) assigned to the project.

2. Specify the elements (usually SARs) you want to assign to this project. You can also assign projects, which have SARs associated with them, as elements on this screen.

   In the TY (Type) fields, specify the corresponding element types (S for SARs, and P for projects). Some of the function keys available on this screen include:

   Key | Description
   --- | ---
   F9  | Redisplays the record for the previously-changed project.
   F10 | Displays the Project Promotion Paths screen.

Options available on this screen include:

- 1 - Displays or edits the SAR detail.
- 2 - Displays or edits the SAR log. The SAR Log Transfer screen appears, which lets you edit the SAR log and update the project SARs. For more
information about updating the SARs by using this log, see Update the SARs in this publication. For information about the SAR log, refer to the Computer Assisted Design Reference Guide.

- 3 - Displays the Pre-Promotion Edit History form. For information about this function, see Promote a SAR in this publication.
- 4 - Promotes a project. For information about this function, see Promote a SAR in this publication.
- 5 - Displays the promotion history of a SAR ('Z' record).
- 6 - Displays or edits notes associated with a SAR (for JD Edwards World environments only; '*' record).
Promote a Project

Promoting a Project

After you create a project, link promotion paths and SARs to it, and complete project development, you are ready to begin the promotion process.

The SAR system uses the Work Order files (F4801 and F4802). If your production environment uses these files, or if the F4802 file has different record types than is needed by version control, define a separate library that contains these files for version control purposes only.

Complete the following tasks:

- Update the SARs
- Validating a Promotion Path
- Promoting a Project

See Also

- Defining a Promotion Path

Update the SARs

The SARs, which are contained in the Work Order Header file (F4801), have detailed information in the Work Order Detail file (F4802). You must update the information in the SARs to reflect software developments that are recorded in the SAR log.

When you developed the software, the changes were recorded in the SAR log automatically. You now must update the F4802 file.
From the Version Control menu (G9261), choose Edit and Promote.

<table>
<thead>
<tr>
<th>Action Code</th>
<th>Project</th>
<th>SAR #</th>
<th>Description</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>G9261</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1. On the Edit and Promote form, inquire on the project you want to promote.
2. In the OP (Option) field next to the project SAR you want to update, enter 2 (SAR Log).

The SAR Log Transfer form displays, which lists all added or changed records logged in the SAR log (F9810) according to record type. The SAR Detail Status field shows whether the record has been updated in the F4802 file.

<table>
<thead>
<tr>
<th>SAR Number</th>
<th>Record Type</th>
<th>Record</th>
</tr>
</thead>
<tbody>
<tr>
<td>1079777</td>
<td>Multiple Jobs Submitted</td>
<td></td>
</tr>
</tbody>
</table>

If this screen lists many SAR log records, you can narrow your search by entering information in the Record Type and the and/or Member fields.

To view the details of a SAR log item, enter 1 (Details) in the OP (Option) field next to the item. On the form that displays, you can edit the SAR details. If the SAR logging system does not log an item that you want to include, press F5 (Add) from the SAR Log Transfer form to add it.

3. To update the F4802 file, press F10 (Update SAR).

Before you update a SAR, verify that each SAR log record should be transferred with the SAR. Change or delete those that are associated incorrectly with the SAR. To display all records with data that can be transferred (TR field value is 1).
Promote a Project

or with test data (TR field value is 0), press F16 (Display Update Capable/All Items). Update only those records that should be transferred with this SAR.

The system creates or updates the records in the SAR file that is located in the SAR library you indicated in the Edit and Promote processing options (not the SAR library appearing in your library list).

Validating a Promotion Path

Before you promote a SAR, you must perform a pre-promotion edit, or validation, against the promotion path that will be used for this SAR.

To validate a promotion path

1. From the Edit and Promote form, inquire on the project you want to promote.
2. In the Promotion Path field, type the name of the promotion path you want to use for your project.
3. In the OP (Option) field next to the project SAR you want to update, enter 3 (Edit).

If you did not choose a promotion path for the project, the Project Promotion Paths form lists all promotion paths defined for the project.

4. In the O (Option) field next to the promotion path you use to promote the project, enter 4 (Select). If you have run pre-promotion edits previously for this SAR, the Pre-Promotion Edit History form lists them. Otherwise, this form is blank.
5. To view the errors associated with a pre-promotion edit, enter 1 (Details) in the OP (Option) field next to the desired history record.

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Cause and Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>0020</td>
<td>Cause: You entered a “From” library that does not exist or you are not authorized to use. Resolution: Correct the library name, create the library, or get authorization to use it.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Cause and Resolution</td>
</tr>
<tr>
<td>------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>0092</td>
<td>Cause: A database table or member could not be opened because it did not exist, a conflicting lock state held by another job exists, or you are not authorized to open it. Resolution: Check your job log messages.</td>
</tr>
<tr>
<td>1046</td>
<td>Cause: An XJDE or ZJDE version was expected but not found. Resolution: If an XJDE or ZJDE version should exist, create it. If not, then change the processing option for form ID P926304.</td>
</tr>
<tr>
<td>1370</td>
<td>Cause: You entered a “From” table that does not exist or you are not authorized to use. Resolution: Review the “From” library for the promotion path control table. Either correct the library name or create the table.</td>
</tr>
<tr>
<td>1371</td>
<td>Cause: You entered a “To” table that does not exist or you are not authorized to use. Resolution: Review the “To” library for the promotion path control table. Either correct the library name or create the table.</td>
</tr>
<tr>
<td>1372</td>
<td>Cause: A key you wanted to copy from the “From” library does not exist. Resolution: Review the “From” library for the promotion path control table. Either correct the library name or re-enter the data record.</td>
</tr>
<tr>
<td>2892</td>
<td>Cause: A “From” library name is the same as the corresponding “To” library name. Resolution: Review the “From” and “To” libraries for the promotion path control table. Make the appropriate changes.</td>
</tr>
<tr>
<td>4395</td>
<td>Cause: No records exist in the Promotion Path Members file (F92401) for the promotion path you specified. Resolution: Complete the Promotion Path Members form for the promotion path.</td>
</tr>
<tr>
<td>4396</td>
<td>Cause: No records exist in the Promotion Path Control file Members file (F92402) for the promotion path you specified. Resolution: Complete the Promotion Path Control Files form for the promotion path.</td>
</tr>
<tr>
<td>4397</td>
<td>Cause: No records exist in the SAR Log table (F9810) for the project you specified. Resolution: In the project master record, change the based-on file for the Pre-Promotion Edit to the SAR Detail table (F4802), then manually update the SAR Detail records for the members and control file records updated by this project.</td>
</tr>
</tbody>
</table>
## Promote a Project

The promotion process involves transferring members and copying control file data.

### Before You Begin

- Before you promote the project, be sure you have edited all items that appear on the SAR Log Transfer screen. Otherwise, the SAR Log Transfer screen displays when you attempt to promote the project.
- You must update all SAR log records associated with the SAR before you promote it.
- You also must resolve all errors (not warnings) before you promote the SAR.

### To promote a project

1. On the Promote a Project form, inquire on the project you want to promote.
2. In the OP (Option) fields next to the project elements you want to promote, enter 4 (Promote).
   - To select all project elements automatically for promotion, press F14.
3. On the Project Promotion Paths form, enter 4 (Select) in the O (Option) field.

### Error Code Cause and Resolution

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Cause and Resolution</th>
</tr>
</thead>
</table>
| 4400       | Cause: No record exists in the Promotion Path Members file (F92401) for the function code of the member you want to promote.  
Resolution: For the specified promotion path, enter the environment for the function code of the member. |
| 4402       | Cause: No record exists in the Promotion Path Control Files file (F92402) for the control table you want to promote.  
Resolution: For the specified promotion path, enter the environment for the control file of the record. |
| 4439       | Cause: An error occurred while you attempted to copy a source code member.  
Resolution: Check for valid library, table, and member names, as well as options in the CPYF command. Check the job log for the error message ID. |
4. In the OP (Option) fields next to the member IDs, specify whether to transfer:
   - Both source and object code (option 1)
   - Source code only (option 2)
   - Object code only (option 3)

To override the From Environment and To Environment object libraries before you transfer the members, press F6 before you enter options 1, 2, or 3. Enter the names of the object libraries to which you want the members transferred.

The system transfers the members you selected to the target environment.

You can review the batch job that was submitted by this transfer program from the JD Edwards World command line. To display the command line, press F2.

If your promotion is successful, the system deletes all SAR log records for transferred items. It also creates a new SAR log record for each transferred item and associates it with the target library.

5. To copy control file data, press F5 (Control Files) from the Software Transfer form.

6. In the OP (Option) fields next to the items you want to copy, enter 1 (Copy to target library).
**Note:** Press F13 to select all items automatically for copying. The system copies the items you selected to the target environment.
Promote Project Updates

Promoting Project Updates

The version control process for project updates includes the following steps:

- To create the transfer library
- To prepare the SAR system
- To define promotion paths
- To define a project
- To update the SARs
- To validate the promotion path
- To promote the project
- To save the transfer library to tape
- To restore the transfer library from tape
- To print the transfer library report
- To load the transfer library
- To transfer individual control table records

To create the transfer library

1. From the Software Install menu (G9262), choose Build Transfer Library.

WARNING!!

If you specify a library that already exists on your system to be used as a software transfer library it will be cleared prior to use. All data and objects in that library will be lost.

If the library you specify does not exist it will be created for you.

(F6 - Execute )
2. After you read the warning message, press F6 (Execute).

```
98312  Build Transfer Library  Form ID... P92414
Build Skeleton Transfer Library  Version... EJD80001
Display Level  4

This job has various options described below. Enter the desired values and press ENTER to continue.

Enter the name of the transfer library MYLIBRARY

to be created. If the library already exists it will be CLEARED before use.
```

3. In the processing option field, enter a name for the transfer library you want to create.

To prepare the SAR system

To prepare your SAR system, see Prepare the SAR System in this guide.

To define promotion paths

From the Version Control menu (G9261), choose Manage Promotion Paths. Use the project update library name as your promotion path name. For information about defining a promotion path, see Define Promotion Paths in this guide.

To define a project

To define a project, see Define a Project in this guide.

To update the SARs

From the Software Install menu (G9262), choose Edit and Promote. For information about updating a SAR, see Update the SARs in Promote a SAR in this guide.

To validate the promotion path

From the Software Install menu (G9262), choose Edit and Promote. For information about validating a promotion path, see Validate the Promotion Path in Promote a SAR in this guide.

To promote the project

From the Software Install menu (G9262), choose Edit and Promote. For information about promoting a SAR, see Promote the Project in Promote a SAR in this guide.
To save the transfer library to tape

1. From the Software Install menu (G9262), choose Save Library to Tape.

   - **Library** field: Type the name of your transfer library.
   - **Device** field: Enter the name of your tape device.

To restore the transfer library from tape

1. From the Software Install menu (G9262), choose Restore Library from Tape.

   - **Saved Library** field: Type the name of your transfer library.
   - **Device** field: Enter the name of your tape device.

To print the transfer library report

1. From the Software Install menu (G9262), choose Print Transfer Report.

   A processing options form displays. Use the cursor keys to display additional processing options.
2. In the first processing option field, type the name of your transfer library.

3. In the remaining processing option fields, select the types of control files for which you want to print information.

4. To print the report, press Enter.
An example of the Print Install Records report (P924143) follows. It provides the total number of records for each type of control file. It also shows whether the transfer record already exists in your control file.

<table>
<thead>
<tr>
<th>Record Type</th>
<th>Item Type</th>
<th>Description</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td>Secondary</td>
<td>New/</td>
<td></td>
</tr>
<tr>
<td>Menus</td>
<td>G9261</td>
<td>Version Control</td>
<td>Changed</td>
</tr>
<tr>
<td>Menus</td>
<td>G9262</td>
<td>Software Install</td>
<td>Changed</td>
</tr>
</tbody>
</table>

DREAMwriter / PO P924143 ZJDE0024 Payee Control File Purge Changed
DREAMwriter / PO P92414 ZJDE0001 Manage Projects Changed
DREAMwriter / PO P92414 ZJDE0001 Build Skeleton Transfer Library Changed
DREAMwriter / PO P924147 ZJDE0001 Load Transfer Software Changed
DREAMwriter / PO P924901 ZJDE0001 SAR Inquiry by Reference Changed

To load the transfer library

Before you load the transfer library, you must create new target libraries for the objects, source code, and data files you will transfer. In your target source library, you must create the following multi-member source files:

- JDESRC
- JDECPY
- F98CRTCMD

Load the contents of your transfer library into your target libraries. The process merges control file records into your library files. You also can transfer control file records individually. For more information, see Transfer Individual Control File Records following this procedure.

1. From the Software Install menu (G9262), choose Load Transferred Library.

   **WARNING!!!**

   This program will transfer source code, objects and new data files into the libraries you name in the processing options. It will also add to or replace data in the control files in your current library list.

   It is recommended that you first run the ‘Print Transfer Report’ to view control file changes.

   { P6 - Execute }

2. After you read the warning message, press F6 (Execute).

   A processing options form displays. Use the cursor keys to display additional processing options.
3. In the first processing option field, specify the name of your transfer library.

4. In the next three fields, specify the libraries you created for the source code, objects, and data files you will transfer.

5. In the remaining fields, select the control files you want the system to transfer.

6. To begin the transfer, press Enter.

The system merges the control files into the target data library. For non-control files, the system adds the file if it currently does not exist in the target data library.
library. If the file does exist in the target data library, the system does not transfer the file or any data. After the transfer process completes, you must change these files manually based on information in the Print Install Records report (P924143).

Even though you can include next numbers in the transfer library and display information about them in the Print Install Records report, the system will not transfer them automatically. This protects your next number tables. After the transfer process completes, you must change them manually based on information in the report.

To transfer individual control table records

1. From the Software Install menu (G9262), choose Copy DD,VO,DW,UDC,SVR,Menus.

<table>
<thead>
<tr>
<th>From Library: MYLIBRARY</th>
<th>To Library: MYDATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dictionary Item: ______</td>
<td>Language: ___</td>
</tr>
<tr>
<td></td>
<td>Appl Ovr: ___</td>
</tr>
<tr>
<td></td>
<td>Form/Rpt: ______</td>
</tr>
<tr>
<td>Vocabulary Overrides:</td>
<td>Language: ___</td>
</tr>
<tr>
<td></td>
<td>Appl Ovr: ___</td>
</tr>
<tr>
<td>DREAM Writer Form: ____</td>
<td>Language: ___</td>
</tr>
<tr>
<td>User Def Codes Sys: ___</td>
<td>Language: ___</td>
</tr>
<tr>
<td>Type: ___</td>
<td></td>
</tr>
<tr>
<td>Software Versions Rep.</td>
<td></td>
</tr>
<tr>
<td>Menu Identification:</td>
<td></td>
</tr>
<tr>
<td>Generic Rate/Meg Sys:</td>
<td></td>
</tr>
<tr>
<td>Type: ___</td>
<td></td>
</tr>
<tr>
<td>F24-More</td>
<td></td>
</tr>
</tbody>
</table>

2. In the From Library, type the name of your transfer library.
3. In the To Library, type the name of the target data file library.
4. In the appropriate fields, enter information that is specific to the control file record you want to transfer.
4 Programming Tools
Overview to Programming Tools

About Programming Tools

Perform the following tasks:
- Work with Data Modeling
- Work with the Object Cross Reference Repository
- Work with Data Dictionary
- Work with Data File Design Aid
- Work with Screen Design Aid
- Work with Report Design Aid
Work with Data Modeling

Working with Data Modeling

The Data Modeling feature provides graphic representation of the relationships of different files. The important aspects of the JD Edwards World Data Modeling feature are:

- It is graphical in its presentation.
- It allows you to narrow the amount of information you view so you can better analyze the file and data relationships.
- It is integrated back to the Data Dictionary and other cross reference tools.

**Note:** To create a data model, you must run the Data Model rebuild, P9804DM from the DREAM Writer version list.

Accessing Data Modeling

To access data modeling

1. Use one of the following methods to access Data Modeling.
   - Inquire on the file through Software Versions Repository (SVR) and press F23.
   - Type the menu selection on the command line and press F23.
   - From the Model Relations form (G9241), select Data Modeling.

   The Data Model Diagrammer displays models from Base Files stored in the Entity Relationship Tracking file (P9804). When using the Data Model Diagrammer for the first time, rebuild the Cross Reference Index of the menu G9642. This rebuild will create data in the Entity Relationship Tracking file and allow file relationships to be built.

   ( F6 - Execute )

2. Press F6 to continue.
   - The Data Modeling form displays with the cursor positioned in the field where you enter a file name.
3. To view the Data Model, enter a file name and press Enter.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Levels</td>
<td>Determines what level of detail you want to view in terms of file relationships. Level 1 represents the highest level and level 10 represents the lowest level. The default value is level 08. Level 01 shows only those files that are directly related to the data model file.</td>
</tr>
<tr>
<td>Function Use</td>
<td>Displays the files that either match or have a function use less than the function use you specify.</td>
</tr>
<tr>
<td>Display Duplicate Relationships</td>
<td>Determines whether you want to display duplicate relationships or not. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>1 No duplicates (default value)</td>
</tr>
<tr>
<td></td>
<td>2 First logical only</td>
</tr>
<tr>
<td></td>
<td>3 All files</td>
</tr>
<tr>
<td>In Sys</td>
<td>Limits your model to only those files from the specified install or reporting system codes. To toggle to reporting system codes, you press F11, Install/Reporting.</td>
</tr>
</tbody>
</table>
4. To narrow the amount of file information displayed, specify values in the four fields appearing in the upper right of the form.

<table>
<thead>
<tr>
<th>Base File</th>
<th>Data Modeling</th>
<th>Max Levels: 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0006</td>
<td>Business Unit Master</td>
<td>Display Dupl. 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Sys 00 02 03 04</td>
</tr>
</tbody>
</table>
- M:1> P0010 F0010 Company Constants
- 1:M> P0901 P0901LB Account Master
- M:M> P4801 P4801LB Work Order Master File
- M:1> P0902 P0902LA Account Balances
- M:N> P0311 P0311LB Accounts Receivable Ledger
- M:N> P0411 P0411LA Accounts Payable Ledger
- M:N> P0911 P0911LA Account Ledger

Opt: 1=Move Top 5=Display 7=Where Used 9=Fields F11=Install/Reporting

Detailed Explanation of a Line

The following figure shows a portion of the Data Modeling form.

<table>
<thead>
<tr>
<th>Base File</th>
<th>Data Modeling</th>
<th>Max Levels: 08</th>
</tr>
</thead>
<tbody>
<tr>
<td>F0006</td>
<td>Business Unit Master</td>
<td>Display Dupl. 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In Sys 00 02 03 04</td>
</tr>
</tbody>
</table>
- M:1> P0010 F0010 Company Constants

Below is an explanation of the components displayed on the form.

- Business Unit Master is the primary file (F0006)
- Company Constants is the secondary file (F0010)
- <m:1> - There is a many to one, bi-directional relation between the files

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantifier</td>
<td>The quantifier notation indicates the following:</td>
</tr>
<tr>
<td></td>
<td>M:1 = many to one</td>
</tr>
<tr>
<td></td>
<td>1:M = one to many</td>
</tr>
<tr>
<td></td>
<td>M:M = many to many</td>
</tr>
<tr>
<td></td>
<td>M:N = many to zero or many</td>
</tr>
<tr>
<td></td>
<td>N:M = zero or many to many</td>
</tr>
<tr>
<td></td>
<td>1:N = one to zero or many</td>
</tr>
<tr>
<td></td>
<td>1:1 = one to one</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Direction</th>
<th>The three direction notations are as follows:</th>
</tr>
</thead>
<tbody>
<tr>
<td>-&gt;</td>
<td>refers to</td>
</tr>
<tr>
<td>&lt;</td>
<td>referred to</td>
</tr>
<tr>
<td>&lt;=&gt;</td>
<td>way relation</td>
</tr>
</tbody>
</table>
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Used to distinguish between prototype and permanent files.</td>
</tr>
<tr>
<td>Subfile portion of screen</td>
<td>Displays the key fields that relate these two files together.</td>
</tr>
</tbody>
</table>

### Function Key Exits from Data Modeling

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F11</td>
<td>Install/Reporting&lt;br&gt; Allows you to toggle between displaying install or reporting system codes.</td>
</tr>
<tr>
<td>F16</td>
<td>Rebuild A File Relationship&lt;br&gt; Exits to a DREAM Writer versions list.&lt;br&gt; The rebuild is fundamentally based upon the program finding a connection between data items. For example, if you create new data items in the Data Dictionary and use those data items when creating a new file, you do not get a graphic representation for that file because the data items do not exist in any other file. To create and present file relationships, there must be at least one data item in the primary file that also resides in some other file as well. Print the Data Model</td>
</tr>
<tr>
<td>F21</td>
<td>Print the Data Model&lt;br&gt; Prints the current data model</td>
</tr>
</tbody>
</table>

### Selection Exits from Data Modeling

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Move Top&lt;br&gt;To select a file in the current data model and move it to the top to view its data model.</td>
</tr>
<tr>
<td>5</td>
<td>Display&lt;br&gt;To view the file relationships. The Define a File Relationship form displays the relationship detail for the two files.</td>
</tr>
<tr>
<td>7</td>
<td>Where Used&lt;br&gt;Exits to the Object Cross Reference Repository and displays all the programs that access the particular file.</td>
</tr>
</tbody>
</table>
### Option 8: Fields

To access the File Field Description form for any file displayed in the Data Model. The File Field Description form presents all the fields in a file, the field type, their size and their position in the file.

#### Example Selection 7 - Where Used

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Field Attr T Start Upd</th>
<th>Len Dec T Loc Yr/M</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVMN</td>
<td>Jon Bugnet Test</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0006</td>
<td>Business Unit Master Revisions - Single</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>F0006A</td>
<td>Business Unit Structure Revisions</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>F0006CSZ</td>
<td>File Conversion - Plug the default value</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>F0006P</td>
<td>Business Unit Master Print</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0006QD</td>
<td>Update Bill Code If Business Unit Type = C</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0006A1</td>
<td>Job Master Revisions</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>F00062</td>
<td>Property/Building Revisions</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>F00071</td>
<td>Work Day Calendar</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0012</td>
<td>Automatic Accounting Instructions Revisi</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0013OD</td>
<td>Convert Amounts to Domestic Decimal</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0018</td>
<td>Tax File Revisions</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>F0018P3</td>
<td>Tax Detail Report by Tax Authority</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

Opt: 1=SVR 2=Create Object 3=Field Explanation F21=Print F16=Regenerate

#### Example Selection 8 - Fields

```
98042  Data Modeling  Max Levels  08
F0010  Company Constants  Display Duplicates  In Sys  00 04 01 04

<1:M> F0006  F0006LB  Business Unit Master

-N:1> F0901  F0901LB  Account Master

-N:M> F4201  F4201LB  Work Order Master File

-N:1> File and Libr: F0006  YENF  FF

-N:1> D006  - Business Unit Master File

-M:1> M006  - Business Unit Master

-M:1> M0200  X01 Business Unit  . . . A 12 1

-M:1> M0500  Description  . . . A 30 13

-M:1> M0600  Level of Detail  . . . A 1 68

-M:1> M0700  Address Number  . . . S 8 0 0 69

-M:1> M0800  Company  . . . A 3 77

-M:1> M0900  Type Business Unit  . A 2 80

-M:1> M1000  Division  . . . A 3 82

-M:1> M1100  Region  . . . . . . . A 3 85

Opt: 1=Move Top 5=Display 7=Where Used 8=Fields F11=Install/Reporting
```

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 4-7
Work with the Object Cross Reference Repository

Working with the Object Cross Reference Repository

The Object Cross Reference Repository locates all the objects associated with a particular member or object. The cross reference files are shipped empty. You must first run the Cross-Ref Index rebuild on menu G9642. If you add a new member to the Software Versions Repository, the Rebuild Cross Reference job must be run to ensure the new member is included in the display. You must have source code on your machine to run this rebuild.

Accessing the Object Cross Reference Repository

To access the Object Cross Reference Repository

Select one of the following methods:

- From the Master Directory (G), choose Hidden Selection 27. From Advanced and Technical Operations (G9), choose Documentation Services. From Documentation Services (G91), choose Object Cross Reference Repository.
- From Software Version Repository, press F15 to access the Object Cross Reference Repository.
- The Fast Path ‘XREF’ may also have been set up.

Example

The following form displays all programs using the file F0006.
The first four fields on this form relate to the object being cross referenced. The remainder of the form lists the members found during the cross reference search.
Conducting an Object Cross Reference Repository Search

All members of the Software Versions Repository are cross referenced, and you can search for these relationships in different ways.

To conduct an Object Cross Reference Repository search

1. From the Documentation Services menu (G91), choose Object Cross Ref. Repository.

   The following form displays the statistics for program P0006.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
<th>Field Attr</th>
<th>T Start</th>
<th>Udp</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0006661</td>
<td>Business Unit Master Conversion</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P06238</td>
<td>Report - Payroll Check Register</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P06371</td>
<td>Report - Certified Payroll Register</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P06111</td>
<td>Tip Credit Generation with Interim Check</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P06638</td>
<td>Sales Allocation Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26419</td>
<td>STAR - Columnar Spreadsheet</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P200</td>
<td>Submit Network Job</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P21250</td>
<td>Texas Z50 Report</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26911</td>
<td>Gas Balancing - Entitlement Extract</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26112</td>
<td>Gas Balancing - Entitlement Explosion to</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26113</td>
<td>Gas Balancing Statement by Sales Point</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26116</td>
<td>Gas Balancing Statement by Owner</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>P26119</td>
<td>Gas Balancing Master Subfile Display</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   Opt: 1=SVR 2=Create Object 3=Field Expl F21=Print P16=Regenerate

2. To conduct an object cross reference repository search, enter an object Name, Type code and To Display code. To narrow the search, enter a Funct Cd.

   Note: If you are unfamiliar with the Cross Reference Relationships codes, type an asterisk (*) in the Type field, as shown below.

3. Press Enter.

4. Enter a 4 to select the desired code. The Object Cross Ref. Repository form displays with the selected codes. The F/D combination shown is very valuable to document the fields in a file.
About the Data Dictionary Repository

The Data Dictionary is the most powerful element in the JD Edwards World software offerings. We define all data items used by JD Edwards World files and programs in the Data Dictionary. By requiring this up-front definition, the Data Dictionary enforces uniformity, consistency, and accuracy across all JD Edwards World applications.

The Data Dictionary represents a centralized repository of all:

- Field definitions
- Program error messages, both interactive and batch
- Menu messages
- Work fields
- User defined help instructions,
- Program and field descriptions accessed by the Help facility

This section has the following topics:

- Understanding the Data Dictionary Structure
- Locating A Data Item Name
- Working with the Data Dictionary
- Working with Data Item Alias Revisions
- Working with the Data Dictionary Glossary
- Working with User Defined Help Instructions
- Working with Data Field Descriptions
- Working with the Next Numbers Facility
- About the Field Reference File
- About the JD Edwards World Message File
- Locating the Rebuild FRF and JD Edwards World Msg File Form

Understanding the Data Dictionary Structure

The following files comprise the Data Dictionary Repository.

The following diagram illustrates the relationships between these files.
Data Item Master (F9200)

This is the master file for the Data Dictionary. Every data item has a record in this file.

Data Dictionary - OneWorld Attributes (F9210)

This file contains the base display and validation rules for all file and data items. It is also used in Enterprise 1 environments.

It also contains database fields (glossary group of D or S), categories (glossary group C), and work fields (glossary group U). In addition, the C alias for each data item is stored in this file.

Data Field Display Text (F9202)

This file lets you define multiple row descriptions and column titles for each data item, based upon language or reporting system (application override). You can add a language value for each language translation required for the row description and column title. The reporting system code allows the entry of jargon or company terminology that overrides the generic text supplied with the application.

Data Item Alpha Descriptions (F9203)

This file contains the alpha and compressed descriptions for all data items. This allows you to perform a Data Dictionary search by description. You can also specify separate alpha descriptions by language preference and reporting system. Every data item has a record in this file.
Data Item Aliases (F9204)

This file contains database fields (glossary group of D or S). It also contains COBOL aliases for each data item.

Error Message Information (F9207)

This file contains error messages that have a program, form, or report ID attached to them. You exit to this program, form, or report when you receive the error. For example, if you receive a user defined code error, you could exit to the User Defined Code Revisions program to modify a value.

Glossary Text File (F9816)

This file contains the glossary text for every data item. Each line of text in the glossary is one record.

Key Index File (F98163)

This file contains key information to link the data items to their glossary and to specific items.

Locating A Data Item Name

The system uses data items to define the parameters of a field or message. For example, AT1 defines the field Search Type. The system maintains each data item used in a file or retrieved for a form or report based on a data item name, such as AT1. To work with the Data Dictionary functions you need to know this name.
To locate a data item name

The JD Edwards World field-level help displays data item names. Position the cursor on any field and press F1. For example, position the cursor in the Search Type field on the Address Book Revisions form (option 3 from menu G01) and press F1. The User Defined Codes form displays for the Search Type field. In the upper right corner of this form is the data item name for the Search Type field, which is AT1.

The data item name is often in the upper right corner of the help screen or the User Defined Codes screen. It can also be seen by using F9 for the Glossary on other Help screens.

Working with the Data Dictionary

The Data Dictionary provides many useful features. You can create data item aliases for other programming languages, work with the glossary, add or change user defined help instructions, and locate data field descriptions.

To work with the Data Dictionary

From menu G92, choose Data Dictionary. The Data Dictionary form is displayed.
You find the Data Dictionary selection on several JD Edwards World menus and repository services.

You can also display the Data Dictionary form by entering the mnemonic DD in the Selection line of any JD Edwards World menu.

Use the following fields where applicable:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>RLs Last Chg</td>
<td>The software version number to be defaulted in the Software Versions Repository file.</td>
</tr>
<tr>
<td>Item Parent</td>
<td>Display only. A data item which becomes the template from which other data items are created. For example, AC (Category Codes) is the parent to AC01.</td>
</tr>
</tbody>
</table>
| Data Item     | The RPG data name. This data field has been set up as a 10-byte field for future use. Currently, it is restricted to 4 bytes so that, when preceded by a 2-byte file prefix, the RPG data name does not exceed 6 bytes. Within the Data Dictionary, all data items are referenced by this 4-byte data name. As they are used in database tables, a 2-character prefix is added to create unique data names in each table specification (DDS). Special characters are not allowed as part of the data item name, with the exception of #, @, $.

You can create protected data names by using $xxx and @xxx, where you define xxx.

Messages can contain up to 10 characters. Types of messages are further defined by glossary group. Form-specific information

Messages can contain up to 10 characters. Types of messages are further defined by glossary group.

Glossary Group | Differentiates data items into types. These types include primary and secondary types, error messages, and help text. See UDC 98/ GG for a complete listing of Glossary Groups.

See also 'What Are the Data Dictionary Glossary Groups?' within this 'Data Dictionary Repository' chapter.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Alpha Desc                 | Database text string that names the data item. Enter text in upper and lower case. The system uses this field to search for similar data items (F4). To enter an alpha description, follow these conventions:  

  Dates - Begin all Date fields with Date  
  Amounts - Begin all Amount fields with Amount  
  Units - Begin all Unit, Quantity, and Volume fields with Units  
  Name - Begin all 30-byte description fields with Name  
  Prompt - Begin any Y/N prompting field with Prompt  
  Address Number - Begin all address numbers (employee, customer, owner) with Address Number |
| Reporting System Code      | Designates the system number for reporting purposes. This rarely differs from the Install System. Exceptions occur for data items used by more than one system. |
| Product Code               | The application system code where the item is used.                                                                                         |
| Type                       | The RPG data type. Refer to UDC 98/DT.                                                                                                        |
|                            | Form-specific information                                                                                                                                 |
|                            | Note: When using the ‘O’ type, create the field as large as possible. This allows the use of ideographic (Double byte) languages such as Japanese. |
| Size                       | The length of the data item.                                                                                                                    |
| Data File Decimals         | The number of positions to the right of the decimal of the data item. This is usually zero. See Display decimals.                               |
| Data Item Class            | Defines the essential attributes and characteristics of a data item. There will be a data item for each of the Data Item Class entries. Data items are grouped into these Classes and the Data Item attributes are the same as the Class. F1 on this field will show all the defined Classes. |
| Item Occurrences           | In setting up a data item in the data dictionary, you may specify a number of array elements. This will cause the automatic creation of one additional data item for each array element.  

  The array data item names are restricted to certain lengths depending on the number of array elements:  
  3 bytes - 1 to 9 elements  
  2 bytes - 10 to 99 elements  
  1 byte - 100 to 999 elements |
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Decimals</td>
<td>Use this parameter to designate the number of decimals in the currency, amount, or quantity fields the system displays. For example, U.S. Dollars would be 2 decimals, Japanese Yen would be no decimals, and Cameroon Francs would be 3 decimals. Data Item Class groups will have the same display decimals as the Class definition. Most decimal data fields are stored in the data files as whole numbers and the Display Decimals value is used to display or print the data correctly.</td>
</tr>
<tr>
<td>Row Description</td>
<td>Stores the title on text and reports. It is used in a manner similar to the column description in the query facility. It should be less than 35 characters. Use abbreviations whenever possible. For example: U/M Units of measure, YTD Year-to-date, MTD Month-to-date, PYE Prior year end, QTY Quantity, G/L General ledger, A/P Accounts payable, DEPR Depreciation</td>
</tr>
<tr>
<td>Column Title</td>
<td>The first line of description that will be used in column headings on a report or screen. This description should be no larger than the data item size, if possible. If the column heading is only one line, it should be placed in this column. Use the second line of the Column Title when two are needed.</td>
</tr>
<tr>
<td>Default Value</td>
<td>Used as the default value on the data entry screen for the associated data item. This value will be entered into the field upon exit from the screen if no other value is entered. The value entered must be the exact same length as the data item size. Place single quotes around the value if it contains any embedded blanks. The keywords *BLANKS and *ZEROS can be used as the default value. When entering a numeric data item with default values, the redisplay of the data item suppresses all leading zeros. <strong>Caution:</strong> If a blank entry is allowed, default values should not be used.</td>
</tr>
<tr>
<td>Field</td>
<td>Explanation</td>
</tr>
<tr>
<td>----------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Data Display Rules</td>
<td>Keywords which describe an editing technique applied when data is displayed. Validation is applied to the data after Enter is pressed. The rules will be incorporated into the source code at program generation time. This is usually numeric editing or masking. The developer can override these rules at the time of program creation. The current list of these rules is kept in the User Defined Codes at SYSTEM = 98 and RECORD TYPE = DR.</td>
</tr>
<tr>
<td>Data Edit Rules</td>
<td>Keywords which control allowed values when data is entered. Validation is applied to the data after Enter is pressed. These controls will be incorporated into the program at generation time. The data may be required to be in a range, in a UDC table, or in a file. The developer can override these rules at the time of program creation. The current list of these rules is kept in the User Defined Codes at SYSTEM = 98 and RECORD TYPE = ER.</td>
</tr>
</tbody>
</table>
| Search Program       | The Help Text Program is used to call a program when the function key - F1 is pressed on its Data Item. When F1 is pressed, the program entered in this field will be executed. There are some fields where the Help Program is hard coded in the Help system such as AN8 in the Name Search program (P01200). If this field is left blank, the glossary will be used. If you wish the User Defined Code window to appear when F1 is pressed, enter ‘*UDC’ in this field (this is the default when ‘UDC’ is entered in the Data Edit Rules field). If you do not want the UDC window to appear and you have ‘UDC’ in the Data Edit Rules field, change this field to be blank. Program Requirements: For your text program to work correctly, you must allow it to accept three standard parameters:  
  - PARM 1 - Field Name, size 10, type alpha  
  - PARM 2 - Return Value, size 30, type alpha  
  - PARM 3 - Return Description, size 30, type alpha  |
| Next Nbr System      | Designates the system number for the Next Number retrieval. See User Defined Codes, system code '98', record type 'SY'. |
| Next Number Index    | The array element number retrieved in the Next Number Revisions program. There are ten NN array elements for each System Code. For example, the next Address Book number (AN8) is array element '01' of system '01'. |
What You Should Know About

Data Dictionary Security

Once a system is operational, you must be particularly careful to secure the integrity of the Data Dictionary. Two facilities are provided to aid you with the security:

- Operational systems coding - System numbers and names are defined in User Defined Codes, system code 98, record type SY. If you place an X in the second line of description for a particular system, it will be designated as operational. Once a system has been set up as operational, all data fields coded to this system are protected from modifications. This control, however, can be violated by removing the X in User Defined Codes.

- Action Code Security - A more prudent form of control is to assign add/ change/ delete authority to only one individual, the database administrator. If you choose to use this control, you should restrict access to the Data Dictionary program (P9201) in Action Code Security. See Working with Action Code Security. All users should be set up with Inquiry authority only. The database administrator would be set up with add/ change/ delete authority.

The Function Keys for the Data Dictionary

The following function keys are available from the Data Dictionary form.

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F4</td>
<td>A data item search facility. Enter the search text in the Alpha Name field on the Data Item Search screen. If you are a double-byte user, you must provide a search description for each data item you create or change in order for the search facility to function properly.</td>
</tr>
<tr>
<td>F6</td>
<td>Repository Services</td>
</tr>
<tr>
<td>F8</td>
<td>User Defined Code Tables</td>
</tr>
<tr>
<td>F9</td>
<td>Automatic Reinquiry</td>
</tr>
<tr>
<td>F10</td>
<td>Glossary</td>
</tr>
<tr>
<td>F15</td>
<td>A data item cross reference</td>
</tr>
</tbody>
</table>

Working with Data Item Alias Revisions

Use the Data Item Alias form to assign alias names to a data item that other programming languages use. When adding a data item of glossary group “D” or “S”, you must enter an alias for that field. This form automatically displays on an
Add function when the alias is not unique. The alias defaults from the alpha description.

To work with data item alias revisions

On Data Dictionary screen

1. Press F5. The Data Field Alias form displays.

2. Enter an alias type and name.

   An alias name must be unique to the system or the system does not let you exit from the Data Field Alias form.

   Current alias types required:
   
   - 1 = PL1 or COBOL
   - 2 = C language

   An alias must adhere to JD Edwards World’ syntax rules of the “C” language.

Working with the Data Dictionary Glossary

What are the Data Dictionary Glossary Groups?

The Data Dictionary consists of several glossary groupings that define the data item in the JD Edwards World software. All glossary groups typically have associated text. The glossary stores this text. The major glossary groups follow:

<table>
<thead>
<tr>
<th>Group</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>JD Edwards World Interactive error messages</td>
</tr>
</tbody>
</table>

   - JD Edwards World defines interactive error messages with numbers less than 5000 and with numbers from 000A to 999Z. For example, 0001
   - Client defines interactive error messages with numbers from 5001 to 9999
<table>
<thead>
<tr>
<th>Group</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| M     | Menu Messages  
|       | JD Edwards World defines menu message data items as MENUMSGxxx, where xxx represents a number. For example, MENUMSG044  
|       | Client defines menu message data items as MENUCLTxxx, where xxx represents a number  
| J     | JD Edwards World Batch error messages  
|       | JD Edwards World defines batch error messages with JDExxxx, where xxxx represents a number less than 7000. For example, JDE0001  
|       | Client defines batch error messages with JDExxxx, where xxxx represents a number greater than 7000 and less than 9000  
|       | The QJDEMSG message file contains batch error messages  
|       | The JD Edwards World program found on Rebuilds and Global Updates (G9642) builds the batch error messages file QJDEMSG.  
| C     | Data Field Function Categories (Data Item Class)  
|       | Groups common data elements  
|       | For example, CURRENCY, QTYINV.  
| D or S| Primary or Secondary Data Items  
|       | Used for validations  
|       | Text on Videos  
|       | Text on Reports  
|       | Field Reference Files - F98FRFA-Z $ and @  
|       | For example, AC for a D data item; AC01 for an S data item  
| F     | Files  
| G     | General Narrative. Used to add information about a specific data item  
| H     | User Defined program Helps  
|       | Client use only for adding custom helps for JD Edwards World programs  
|       | For example, U00MENU, U01051.  
|       | When HELP is keyed on command line, ‘F5’ on the help window indicates that there are customer helps available. Use F5 to view.  
| L     | Report Messages. Messages or warnings for certain procedures, or letters written and produced through DREAM Writer  

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 4-23
To work with the glossary


   If your glossary group is E, H, J, or M, this screen automatically displays when you press Enter on the main Data Dictionary form.
2. Do the following that applies:
   - Use the Language, Applic Override, and Scrn/ Rpt fields for jargon. See About Language and Jargon for details.
   - Use cursor keys to see additional text lines.
   - When entering an “E” glossary group item, which is an interactive error message, use F5 to define a program, form, or report to reference when the system displays the error message.
   - On double-byte machines, this form displays the Search Desc field. To ensure the data item search facility functions properly, you must enter a search description for each data item you create or change. You can enter it on this form or on the Data Dictionary form.

   **Note:** Always leave the last two character positions of each text line blank.

### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
</table>
| Data Item | The RPG data name. This data field has been set up as a 10-byte field for future use. Currently, it is restricted to 4 bytes so that, when preceded by a 2-byte file prefix, the RPG data name does not exceed 6 bytes. Within the Data Dictionary, all data items are referenced by this 4-byte data name. As they are used in database tables, a 2-character prefix is added to create unique data names in each table specification (DDS). Special characters are not allowed as part of the data item name, with the exception of #, @, $.
You can create protected data names by using $xxx and @xxx, where you define xxx.
Messages can contain up to 10 characters. Types of messages are further defined by glossary group.  |
| Form-specific information | If you are adding an error message, this field must be left blank. The system assigns the error message number using next numbers. The name displays on a successful add. You should assign interactive error message numbers greater than 5000. |
15B Work with Data Dictionary

Field | Explanation
--- | ---
Glossary Group | Differentiates data items into types. These types include primary and secondary types, error messages, and help text. See UDC 98/GG for a complete listing of Glossary Groups.

See also What Are the Data Dictionary Glossary Groups?.

Form-specific information

**Note:** If you need to assign your own error message numbers, use 4 digit numbers greater than ‘5000’.

For help text (glossary group H), the data dictionary “Inquiry/Revision Program” field may be used to specify the name of a follow-on item.

To create your own messages for the IBM message file (glossary group J), begin the data item name with your own three characters, for example CLT0001.

---

**Working with User Defined Help Instructions**

The easiest way to modify help instructions is to utilize the User Defined Instructions in Data Dictionary.

To work with user defined help instructions

On the Data Item Glossary Revisions screen

```
#01001  Data Item Glossary Revisions  Language  .  .  .  .  .  .  .
Applic Override  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Scrnn/Rpt  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

Action Code  .  .  .  .  .  .  .  .  .  D00MNU  Desc: Help - User Defined Instructions
Install System Code  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .
Glossary Group  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .

This is a sample of user defined instructions that may be entered by users for any given program in the system. If you wish to provided high-lighted or underlined or both high-lighted and underlined text refer to the special attributes section of help instructions. All user defined instructions may be entered directly into the data dictionary.
```

JD Edwards World provides an example record (U00MNU) in your system.

1. Enter a program name in the Data Item field, replacing the “P” with “U”. For example, for program P01051, create a data item U01051.

2. Enter H in the Glossary Group field. The H Glossary Group defines user defined help. JD Edwards World does not replace H Glossary Group data items during an upgrade.

3. Perform an add or change.
On the Help Task List form, “F5=User Inst” is displayed if you wrote your own User Defined Help instructions.

Working with Data Field Descriptions

To work with data field descriptions

1. From the Data Dictionary screen, press F11.

2. On the Data Field Descriptions screen, enter specific jargon or language descriptions for each data item. See About Language and Jargon in Technical Foundation for details.

Working with the Next Numbers Facility

The Next Number facility controls the automatic numbering for such items as new G/L account numbers, voucher numbers, address numbers. It allows you to specify what numbering system you want to use and gives you a method of incrementing numbers to reduce transpositions and keying errors.

Complete the following tasks:

- Locate the Next Numbers facility
- Work with Next Numbers by company and fiscal year
To locate the Next Numbers facility

From menu G00, choose Next Numbers.

What You Should Know About

Next Numbers

The next numbers file is F0002

- 10 element array
- 1 record per system
- Modulus 11 check digit optional

Once set, do not change

- Has an impact on system performance
- Will not duplicate numbers. When it reaches max, starts over
- Cannot change position of user or add new entry without programming modifications

Ties with the Data Dictionary

- Data Item in Data Dictionary points to the Next Number System. For example, Data Item AID has System Code 09/01.
To work with Next Numbers by company and fiscal year

Use Next Number by Company for these original documents:
- Journal Entries
- Accounts Payable Vouchers
- Accounts Receivable Invoices
- Sales Orders
- Purchase Orders

1. From Next Numbers, press F8.

2. Set the Next Number constant field to maintain next numbers by:
   - Company
   - Company and Fiscal Year

About the Field Reference File

The Field Reference File contains the specifications for each data item in the JD Edwards World Data Dictionary. Because the JD Edwards World Data Dictionary is different from the standard IBM Data Dictionary, each data item record needs to be translated from the JD Edwards World standard to the IBM standard.

When building the Field Reference File, JD Edwards World groups the data items. Items that begin with “A” are translated into the IBM-readable format and accessed through file F98FRFA. Data items that begin with “B” are accessed through F98FRFB. Each letter of the alphabet has a corresponding F98FRF file. Client data items are stored in F98FRFS and F98FRF@. You can rebuild one file at a time. You can also build the message file in alternative languages. Due to the IBM limitation on the number of fields allowed in a file, it may be necessary to have more than one “FRF” file. The field information from the World Data Dictionary is stored in separate physical files (F98FRFA1, F98FRFA2, and so on). F98FRFA is a logical which joins the physical files.
What Happens with the Rebuild?

The system does the following:

- Rebuilds F98FRFA-Z, $, and @ using Data Dictionary data item glossary groups D and S
- Rebuilds the message file (QJDEMSG) in QGPL
- Uses a processing option located on Form ID J98DDMSGF, to determine which library to build the QJDEMSG file. The default is QGPL
- Does not rebuild the JD Edwards World message file if entering a single field reference file to be built
- Builds a separate message file for each language installed. Enter ‘**’ for all languages installed on the system.

Always rebuild the files in the same library as previously built.

About the JD Edwards World Message File

The JD Edwards World Message (QJDEMSG) file contains all the messages that are coded Glossary Group J. The programs access the messages from this file. If a client adds messages with Glossary Group J, a rebuild is necessary to correctly add the new messages to the JD Edwards World Message (QJDEMSG) file.

Rebuilding only the JD Edwards World Message File?

The system does the following:

- Rebuilds the message file (QJDEMSG) in QGPL. Uses a processing option, Form ID J98DDMSGF, to determine which library to build the QJDEMSG file. The default is QGPL
- Picks up Data Dictionary data item glossary group J

Enter a value from UDC table 01/ LP to generate a message file for a single language. Only those translated error messages which can be seen on the Glossary screen with a language code will be included. Enter ‘**’ for all languages installed on the system.
Locating the Rebuild FRF and JD Edwards World Msg File Form

To locate the Rebuild FRF and JD Edwards World Msg File form

From menu G9642, choose FRF & JD Edwards World Msg File. The first and third fields are self-explanatory. The source referred to on the second parameter is not useful so enter QTEMP in that field.

<table>
<thead>
<tr>
<th>98FRF</th>
<th>Rebuild FRF &amp; JDE Msg File</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Field Reference Files are facsimiles of the J. D. Edwards Data Dictionary and are vital for the creation of all data base files. The version of the Data Dictionary upon which they are based determines the type and characteristics of all application data elements. This procedure will recreate these fields based upon the Data Dictionary files found in the library specified, placing the DDS source in the JDESSRC source file the Source Library selected, with the Field Reference Files being created in the Data Library selected.</td>
<td></td>
</tr>
<tr>
<td>Base Field Ref Files on Data Dictionary in Library __________</td>
<td></td>
</tr>
<tr>
<td>Create Field Ref source in Source Library __________</td>
<td></td>
</tr>
<tr>
<td>Create Field Ref Files in Data Library __________</td>
<td></td>
</tr>
<tr>
<td>Single field ref($, @, A-Z or blank=all) __</td>
<td></td>
</tr>
<tr>
<td>Language for message file (** for all) . __</td>
<td></td>
</tr>
<tr>
<td>NOTE: Generation of Field Reference and Message File is submitted to batch. No data files may be created during this generation process.</td>
<td></td>
</tr>
<tr>
<td>Press Enter to Rebuild Field Reference Files F3=Exit without Rebuild</td>
<td></td>
</tr>
</tbody>
</table>

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 4-31
About the Data File Design Aid

JD Edwards World Data File Design Aid provides a simple mechanism for creating Data Description Specifications (DDS) for physical and logical files.

To enforce standards, JD Edwards World recommends against file changes made through the Source Entry Utility (SEU). Changes should be done through File Design Aid. Non-JD Edwards World changes to a JD Edwards World file make support difficult if not impossible.

What You Should Know About

Enforced Prefixes

Data items are defined in the JD Edwards World Data Dictionary. Within files, these data item names are qualified with a prefix to make them unique. Every data file in JD Edwards World software is assigned a two-character prefix. For example:

- Business Unit Master file is MC
- Address Book Master is AB
- The data name MCU in the Business Unit Master file is MCMCU
- The data name in the Address Book file is ABMCU
- Use of prefixes ensures that data item names are both consistent and unique.

Enforced naming conventions

JD Edwards World file names begin with an F prefix and the format within that file begins with an I prefix.

Data Dictionary validation

- All data fields defined in files are verified against the Data Dictionary.
- Programmers cannot enter data names without first creating and documenting them in the Data Dictionary.
- Prefixes of $ and @ are reserved for client use.

Automatic reference to Field Reference Files

- JD Edwards World uses IBM’s Field Reference File (FRF) facility when creating files. When creating the DDS for a file, you need only enter the Data Dictionary data item name. The Data File Design Aid automatically enters the correct keywords for referring to the FRFs.
- If data items are added to the Data Dictionary, the user needs to run the rebuild for the Field Reference Files before using Data File Design Aid.
Resequencing

- A sequence number allows you to rearrange data items within a file while you are designing.

About Assigning the File Prefix

File prefixes are assigned through the Software Versions Repository. The information in this form comes from a logical file built over the Software Versions Repository file, F9801. The information in this form is updated automatically whenever the user adds, updates, or deletes software version repository records for files.

**Caution:** Programmers are responsible for not assigning the same prefix to two different files used in the same program.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action Code</td>
<td>I</td>
</tr>
<tr>
<td>Description</td>
<td>EDM Item Master File</td>
</tr>
<tr>
<td>Function Use</td>
<td>210</td>
</tr>
<tr>
<td>System Code</td>
<td>52</td>
</tr>
<tr>
<td>Reporting System</td>
<td>52</td>
</tr>
<tr>
<td>Base Member Name</td>
<td>F9801</td>
</tr>
<tr>
<td>File Prefix</td>
<td>OK</td>
</tr>
<tr>
<td>Q Source</td>
<td>Object Source</td>
</tr>
<tr>
<td>F</td>
<td>Library</td>
</tr>
<tr>
<td>JDFSEC71</td>
<td>JDFTDA71</td>
</tr>
</tbody>
</table>

**Note:** A file prefix can display in this list more than once if it is attached to more than one file.

To view all file prefixes currently in use:

- Press F1 on the File Prefix field.

**Note:** A file prefix can display in this list more than once if it is attached to more than one file.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>One or two character prefix for each field in a file</td>
</tr>
<tr>
<td>File Name</td>
<td>Member ID from SVR</td>
</tr>
<tr>
<td>File Description</td>
<td>The description of a file in the Software Versions Repository. The member description is consistent with the base member description.</td>
</tr>
</tbody>
</table>

**Note:** The Q series is reserved for clients.

If creating a new logical file, the prefix defaults from the based-on physical file.
Pressing F10 from this form displays all file prefixes that you should not use.

Entering Data File Design Aid

To enter Data File Design Aid

1. Inquire on a physical file.
2. Copy the production source down to a development environment.
3. Choose Option 10 to take you to the appropriate Design Aid form based on the members Function Code value.
   - A PF or LF value takes you to File Design Aid.

<table>
<thead>
<tr>
<th>Data Item</th>
<th>Data Field Desc.</th>
<th>R/S</th>
<th>Function Specifications</th>
<th>Seq No</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1T</td>
<td>Item ID</td>
<td>R</td>
<td>REFLED:XIT P99PREX</td>
<td>2.00</td>
</tr>
<tr>
<td>X94</td>
<td>Description</td>
<td>R</td>
<td>REFLED:X94 P99PREX</td>
<td>3.00</td>
</tr>
<tr>
<td>X74</td>
<td>Item Type</td>
<td>R</td>
<td>REFLED:X74 P99PREX</td>
<td>4.00</td>
</tr>
<tr>
<td>X2T</td>
<td>Date Last Ship</td>
<td>R</td>
<td>REFLED:X2T P99PREX</td>
<td>5.00</td>
</tr>
<tr>
<td>X8U</td>
<td>Business Unit</td>
<td>R</td>
<td>REFLED:X8U P99PREX</td>
<td>6.00</td>
</tr>
<tr>
<td>X0T</td>
<td>Quantity - On Hand</td>
<td>R</td>
<td>REFLED:X0T P99PREX</td>
<td>7.00</td>
</tr>
<tr>
<td>X0M</td>
<td>Item Unit of Measure</td>
<td>R</td>
<td>REFLED:X0M P99PREX</td>
<td>8.00</td>
</tr>
<tr>
<td>X01</td>
<td>Item Category Code</td>
<td>R</td>
<td>REFLED:X001 P99PREX</td>
<td>9.00</td>
</tr>
<tr>
<td>X02</td>
<td>Item Category Code</td>
<td>R</td>
<td>REFLED:X002 P99PREX</td>
<td>10.00</td>
</tr>
<tr>
<td>X03</td>
<td>Item Category Code</td>
<td>R</td>
<td>REFLED:X003 P99PREX</td>
<td>11.00</td>
</tr>
<tr>
<td>X04</td>
<td>Item Category Code</td>
<td>R</td>
<td>REFLED:X004 P99PREX</td>
<td>12.00</td>
</tr>
<tr>
<td>X05</td>
<td>Item Category Code</td>
<td>R</td>
<td>REFLED:X005 P99PREX</td>
<td>13.00</td>
</tr>
<tr>
<td>X1T</td>
<td>Item ID</td>
<td>R</td>
<td>REFLED:X1T P99PREX</td>
<td>14.00</td>
</tr>
</tbody>
</table>

P1-Exit/Save  P16-Search by File  P1-Search by Name  P4-Field Attributes

Field | Explanation
---|---
File Description | SVR member description for the file.
Unique Keys(Y/N) | Specifies if the data file contains unique keys.
  - If you say yes, Data File Design Aid puts the UNIQUE keyword in the DDS. As a result, no two records may have duplicate keys.
  - If you say no, Data File Design Aid leaves the UNIQUE keyword out of the file DDS.
Form-specific information | If a file can be organized so the key will uniquely identify only one specific record, define the Unique Keys field. Uniqueness can be specified for physical and logical files.
Most JD Edwards World physical files in the past have been defined as sequential and logicals were used for creating keyed sequences. More recently, however, physical files have been keyed.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member ID</td>
<td>The name assigned to the file. Defaults in from the Software Versions Repository.</td>
</tr>
<tr>
<td>File Prefix</td>
<td>This is the SVR field which indicates the prefix associated with a file.</td>
</tr>
<tr>
<td>Src Library</td>
<td>The library where the source for the data file resides. Defaults in from the Software Versions Repository.</td>
</tr>
<tr>
<td>Source File Name</td>
<td>The name of the file within the source library that contains the source member. Defaults in from the Software Versions Repository.</td>
</tr>
<tr>
<td>Based on File</td>
<td>Designates the physical file on which a logical file is based.</td>
</tr>
<tr>
<td></td>
<td>▪ Defaults in from the Software Versions Repository and only displays for logical files.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>For physical and logical files, the Based On File is the same as the physical file.</td>
</tr>
<tr>
<td></td>
<td>For join files, the Based On File is the name of the first physical file that the join is built over.</td>
</tr>
<tr>
<td>Data Item</td>
<td>The Data Dictionary name of the field or the record format name.</td>
</tr>
<tr>
<td></td>
<td>▪ The file prefix is added to create a unique data name for each field specified in this field.</td>
</tr>
<tr>
<td></td>
<td>▪ The record format line is automatically defaulted in.</td>
</tr>
<tr>
<td>K/ S</td>
<td>Identifies the DDS Type indicating whether the field is a format name (R), key field (K), select logic field (S) or omit logic field (O). It may be used in conjunction with information that appears in the Function Specifications field.</td>
</tr>
<tr>
<td>Function Specifications</td>
<td>Used with the DDS Type specified in the K/ S column.</td>
</tr>
<tr>
<td></td>
<td>▪ If it is a record format name this field will be blank.</td>
</tr>
<tr>
<td></td>
<td>▪ Contains the PFILE (Filename) statement for a logical file and you enter: JFILE (Filename Filename) statement for join files listing all the files involved in the join. Right below the JFILE statement, you use the JFLD (Field Field) statement to list the fields that are used to construct the join.</td>
</tr>
<tr>
<td></td>
<td>▪ If you are defining a normal data item and you want the FRF field designation pulled in, you leave it blank.</td>
</tr>
<tr>
<td></td>
<td>▪ If you are defining Select/ Omit logic on a field, you enter the logic itself.</td>
</tr>
<tr>
<td></td>
<td>▪ If you are defining a key data item, you may leave the Function Specifications field blank, or you may enter any valid DDS function keyword (DESCEND, RENAME, SIGNED, ZONE, and so forth).</td>
</tr>
</tbody>
</table>
16B Work with Data File Design Aid

Field Explanation

Seq No Determines the order of the fields in the file.

Form-specific information
When designing a physical, list the component fields in descending order of their importance to the file. The K entries for keyed fields must always be last in sequence number within the Data File Design Aid program itself.

Data Item Type A is for Alpha, S is for simple numeric, P is for Packed numeric, and O is for Open (any character can be entered).

Item Size Length of field taken from the Data Dictionary.

Display Decimals Usually Packed fields, the number of decimal places on a display. The data will actually be stored in the field in the file as a whole number.

Note: The detail area includes additional information: data item type, data item size, and number of display decimals. This information will be available after the enter key is pressed, at which time the information will be retrieved from the Data Dictionary along with the REFFLD keyword which indicates which F98FRF file was used to retrieve the field information.

Sample - Logical File

JD Edwards World logical files contain all fields from the PF, only keys are specified.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq No</td>
<td>Specifies the order of fields in the file.</td>
</tr>
<tr>
<td>Form-specific information</td>
<td></td>
</tr>
<tr>
<td>Data Item Type</td>
<td>Determines the type of data: A for Alpha, S for Simple Numeric, P for Packed Numeric, O for Open.</td>
</tr>
<tr>
<td>Item Size</td>
<td>Specifies the size of the field as taken from the Data Dictionary.</td>
</tr>
<tr>
<td>Display Decimals</td>
<td>Indicates the number of decimal places on a display. The data is stored as a whole number.</td>
</tr>
</tbody>
</table>

92102 Data File Design Aid

File Description ... X - Business Unit, Item ID
Unique Key (Y/N) ... X
File Prefix ... X
File Library ... X
Src File Name ... X
Source File Name ... X

<table>
<thead>
<tr>
<th>Data Item, Data Field Description</th>
<th>K/S Function Specifications</th>
<th>Seq No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Unit</td>
<td>E - FILE(F9280)</td>
<td>4.00</td>
</tr>
<tr>
<td>Item ID</td>
<td>E</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.00</td>
</tr>
</tbody>
</table>

F1-Exit/Save F16=Search by File F1=Search by Name P4=Field Attributes
Sample - Logical File with Selects

This example represents an AND condition for the selects.

```
<table>
<thead>
<tr>
<th>Data Item</th>
<th>Data Field Desc</th>
<th>K/S</th>
<th>Function Specifications</th>
<th>Seq No</th>
</tr>
</thead>
<tbody>
<tr>
<td>10911</td>
<td>Account ID</td>
<td>E</td>
<td>FILE(P0911)</td>
<td>1.00</td>
</tr>
<tr>
<td>72</td>
<td>Ledger Type</td>
<td>K</td>
<td></td>
<td>2.00</td>
</tr>
<tr>
<td>4031</td>
<td>DOT Sub</td>
<td>E</td>
<td></td>
<td>3.00</td>
</tr>
<tr>
<td>501</td>
<td>Subledger</td>
<td>E</td>
<td></td>
<td>4.00</td>
</tr>
<tr>
<td>5041</td>
<td>Date - Service/Tax</td>
<td>E</td>
<td></td>
<td>5.00</td>
</tr>
<tr>
<td>4085</td>
<td>Date - Service/Tax</td>
<td>E</td>
<td></td>
<td>6.00</td>
</tr>
<tr>
<td>4050</td>
<td>Date - Service/Tax</td>
<td>E</td>
<td></td>
<td>7.00</td>
</tr>
<tr>
<td>4051</td>
<td>Document Type</td>
<td>E</td>
<td></td>
<td>8.00</td>
</tr>
<tr>
<td>4500</td>
<td>Document (Voucher,</td>
<td>E</td>
<td></td>
<td>9.00</td>
</tr>
<tr>
<td>4510</td>
<td>Document Company</td>
<td>E</td>
<td></td>
<td>10.00</td>
</tr>
<tr>
<td>4920</td>
<td>G/L Posted Code</td>
<td>E</td>
<td></td>
<td>11.00</td>
</tr>
<tr>
<td>4930</td>
<td>Bill Code</td>
<td>E</td>
<td></td>
<td>12.00</td>
</tr>
</tbody>
</table>
```

Creating Join Files and Work Files

To create a join file or a work file, you should use the Source Entry Utility.

Function Keys From File Design Aid

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Using F1 in the Data Item field takes you to the Data Item Search form.</td>
</tr>
</tbody>
</table>
Function Key | Definition
--- | ---
F2 | Access the command line to enter a JD Edwards World or IBM command without having to exit to Command Entry or a menu. If you are secured out of Command Entry or Menu Traveling, you can still get to this command line but you cannot execute commands or menu travel.
F3 | Press to exit Data File Design Aid, the following form appears.

Example F3 - Data File Design Aid

On this form, you can choose to:

- Exit without saving the changes made.
- Exit and save the changes made.
- Save the changes made and return to the Design Aid form.

Function Key | Definition
--- | ---
F6 | This form provides access to other repository services within JD Edwards World.
F16 | Accesses the File Field Description form to view file formats and field descriptions for any file on the system.
What are the Data File Design Aid Standards?

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unique Keys</td>
<td>Specifies if the data file contains unique keys. If Yes, FDA puts the UNIQUE keyword in the DDS. No two records in the data file can have duplicate data in the key fields. If No, FDA leaves the UNIQUE keyword out of the file DDS. Records with duplicate keys will be allowed in the data file.</td>
</tr>
<tr>
<td>File Description</td>
<td>The description of a record in the Software Versions Repository file. The member description is often consistent with the base member description.</td>
</tr>
<tr>
<td>Based On File</td>
<td>Designates the physical file on which a logical file is based.</td>
</tr>
</tbody>
</table>

Form-specific information

- If a file can be organized so the key will uniquely identify only one specific record, define the Unique Keys field. Uniqueness can be specified for physical and logical files.  
- Most JD Edwards World physical files in the past have been defined as arrival sequence and logicals were used for creating keyed sequences. More recently, however, physical files have been keyed.

- Physical files should have a description that explains the purpose of the file.
- Logical files should be designated as follows: LF - fldname, fldname, fldname; where fldname is a key field.
- Join files should be designated as follows: JF - filename/ filename/ filename - fldname, fldname, fldname; where the filename is a file over which the join is built and fldname is the key field joining the files.
- Work files should be designated as follows: WF - filename; where filename is the file that the work file accesses.

- Defaults in from the Software Versions Repository and only displays for logical files.

- For physical and logical files, the Based On File is the same as the physical file.
- For join files, the Based On File is the name of the first physical file that the join is built over.
16

Work with Data File Design Aid

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordering of Fields</td>
<td>Determines the order of the fields in the file.</td>
</tr>
<tr>
<td></td>
<td>Form-specific information</td>
</tr>
<tr>
<td></td>
<td>When designing a physical, list the component fields in descending order of</td>
</tr>
<tr>
<td></td>
<td>their importance to the file. Keyed items must always be last in sequence</td>
</tr>
<tr>
<td></td>
<td>number within the Data File Design Aid program itself.</td>
</tr>
<tr>
<td>Logical Files</td>
<td>Logical files may include all fields; we do not define specific fields.</td>
</tr>
<tr>
<td>Recompiling</td>
<td>When recompiling a physical, you need to delete any logicals or joins from</td>
</tr>
<tr>
<td></td>
<td>the data file library and then recompile them after the physical has been</td>
</tr>
<tr>
<td></td>
<td>recompiled.</td>
</tr>
<tr>
<td>Record Format</td>
<td>It is a JD Edwards World standard that only one record format is defined for</td>
</tr>
<tr>
<td></td>
<td>each physical and logical file. Joins may contain more. Record format names</td>
</tr>
<tr>
<td></td>
<td>begin with I followed by the physical file number.</td>
</tr>
<tr>
<td>Field Reference Files</td>
<td>Used in all file creations to retrieve field descriptions.</td>
</tr>
</tbody>
</table>

**Merge Functions for Program Temporary Fix (PTF) Installations and Reinstallations**

The PTF installation or reinstallation does the following:

- A PTF installation prints a report that identifies all files that are in the PTF library but were not installed in the client’s production libraries. You must add the new files manually into the appropriate libraries.
- A reinstallation prints a report to add new files into appropriate libraries.
- Updates JDFDATA in a PTF installation; replaces JDFDATA in a reinstallation.
- May add new keys to both logical and physical files.
- May change the file format of logical or physical files.

**Note:** The Data Model displays relational models of the major files within the JD Edwards World environment.

**Data File Design Aid Summary**

In summary, the Data File Design Aid has the following features or restrictions:

- It has direct ties to the Data Dictionary and the Field Reference Files.
- It attaches a two-character prefix to each data item to create a unique field within the file.
- A record format must be defined for all files with a K/S value of R. This is the default record format.
- The PFILE keyword is automatically pulled in for logical files.
- Logical files must have a Based on File designated in the Software Versions Repository, which carries over to the design form.
- You must enter the data item names from the Data Dictionary.
- Perform these steps for creating a new file:
  - Data items must reside in the Data Dictionary.
  - You must rebuild the FRF files if new data items were added (from the Rebuilds menu, G9642).
  - A new file must have a file prefix specified on the Software Versions Repository record.
- Field Reference Files are characterized by the following:
  - They contain all the definitions for creating fields.
  - There are 28 in all (F98FRFA-F98FRFZ, F98FRF$, and F98FRF@).
  - Each field reference file points to all the data items beginning with the same character as the field reference file.
    For example: F98FRFA is a logical file which connects the F98FRFA1, F98FRFA2, and so on. physical files which contain all the Data Dictionary data items beginning with the letter A.
Work with Screen Design Aid

About Screen Design Aid

Screen Design Aid (SDA) is an interactive feature you use to design and maintain screens. This full-screen editor validates your work against the Data Dictionary and adds records to vocabulary overrides. You can work with multiple record formats simultaneously and you can move fields from one format to another.

Below is a list of some features of SDA:

- Design is conducted in a safe work environment. If you make a mistake you can exit without changing a screen’s Data Description Specifications (DDS).

- Screen specifications are stored in data structures in the QRECOVERY library. This is similar to the IBM recovery of SEU.

- You can create a screen in normal mode (80 columns by 24 rows) or wide mode (132 columns by 27 rows). You can also design wide screens on 80 column devices using a windowing facility.

- Answering initial yes/no options allows you to create a basic screen skeleton for a subfile, non-subfile or window-style screen.

- SDA is fully integrated with the Data Dictionary and vocabulary override files. You can place fields on the screen by referring to a Data Dictionary name and override default attributes, if necessary. You can place vocabulary override fields on the screen and, if desired, modify their contents through the full screen.

- SDA is fully integrated with the system database. You can select fields from the system database, create a pick list and then reorder fields in the pick list. You can place fields on the screen individually or all at once by pinpointing locations on the full screen with an ampersand (&) or asterisk (*).

- SDA has full screen capability. You can add, change, move, or delete fields by entering control characters directly on the screen.

- Unlike the IBM SDA, JD Edwards World SDA allows you to work with multiple record formats at one time. You can display and change any combination of formats simultaneously (as long as they do not overlap). You can also move fields from one format to another.

- SDA allows you to simulate a screen at program execution time. You can run the simulation for any set of conditioning indicators to represent a particular error condition or other program functions.
This section contains the following:

- Editing Commands
- Prefix Standards
- Field Name Standards
- Working with Screen Design Aid
- Function Key Exits
- Updating an Existing Field
- Accessing Fast Path Create for a New Screen
- Adding Fields without Using a Pick List
- Adding a Literal Field
- Using the *BOTH and *ALL Features
- Compiling Your Screen
- Screen Design Standards and Tips
- Adding Screen Fields Using Pick List
- Selecting Database Fields
- Placing Fields on a Screen Using a Pick List
- Function Key Exits from Screen Design Aid
- Changing Subfile Boundaries
- Process Overview - Placing Selected Fields
- Process Overview - Revising the Field Definition
- Process Overview - Revising Vocabulary and Function Keys
- Function Keys for Screen and Display Format Control
- Summary of Screen Design Aid

Editing Commands

Standard editing commands available in SDA include:

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>*DEL</td>
<td>Delete fields (used in Field Definition window)</td>
</tr>
<tr>
<td>d</td>
<td>Delete fields (used in Field Definition window)</td>
</tr>
<tr>
<td>(cannot be uppercase D)</td>
<td></td>
</tr>
<tr>
<td>&lt;&lt;, &gt;&gt;</td>
<td>Shift fields to the left or right</td>
</tr>
<tr>
<td>(xx...xx) 'xx...xx'</td>
<td>Literals (use apostrophes)</td>
</tr>
<tr>
<td>-</td>
<td>Move from position.</td>
</tr>
<tr>
<td>=</td>
<td>Move to position.</td>
</tr>
</tbody>
</table>
### Command Description

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>- -</td>
<td>Move block from position</td>
</tr>
<tr>
<td>=</td>
<td>Move block to position.</td>
</tr>
<tr>
<td>F7</td>
<td>Restore the screen if you accidentally press Field Exit or a power failure knocks you off.</td>
</tr>
</tbody>
</table>

### Precautions and automatic features of the SDA include:

- Do not use the INSERT or DELETE keys while in the actual design portion of SDA.
- SDA automatically assigns editing indicators.
  - Indicators 40 to 79 are reserved for editing.
  - Indicator 40 is reserved for the Action Code field.
  - Indicator 41 is reserved for the key fields.
  - If you use all available indicators, you will get an error message.
- Indicator 37 is used in subfile screens to highlight all fields on the last line of the subfile to indicate that no more records exist.

### Prefix Standards

Prefix standards for use in the SDA include:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Description</th>
</tr>
</thead>
</table>
| VD     | Screen display fields.  
  - VD fields display database information from the file being used for the screen and you can use them to enter database information.  
  - Default size is the size specified in the Data Dictionary for the data item being displayed.  
  - Reside in the based on file and can be input/output. |
| SF     | Subfile fields.  
  - Same as VD fields, but they are in a subfile.  
  - Default size is the size specified in the Data Dictionary for the data item being displayed plus editing characters. |
| SH     | Subfile Hidden fields.  
  - SH fields store data that is not displayed on a screen. |
# Field Name Standards

Field name standards for use in SDA include:

<table>
<thead>
<tr>
<th>Field</th>
<th>Standard</th>
</tr>
</thead>
</table>
| VC0 - Screen constants | VC0 (zero) fields display definitions or descriptions for a single piece of data or for a group of data.  
  VC0 fields are always output fields and the description that is loaded into the VC0 field is obtained from a separate file.  
  - For example, if creating a screen using the Item Master file (F92801), you need to take the Item Master Business Unit field and chain out to the Business Unit Master file (F0006) to get the description for that Business Unit.  
  - You enter *VC0 for the Field Name field in the Field Definition window when adding a new constant or description field.  
  - The default size for VC0 fields is 30. |
| VTX - Screen text | These fields display the row description or column headings from the Data Dictionary.  
  - The text that displays in the VTX fields is stored in the Vocabulary Overrides file (F9220).  
  - You can type directly over Vocabulary Override fields in SDA.  
  You enter *VTX in the Field Name field in the Field Definition window for the next sequential text field name.  
  - The default size for VTX fields is 16. |
| Line 24 is always VDL24 | You cannot change the text for Line 24 by using the Field Definition window because it is too large.  
  Type over the text in Line 24 to change it. |
| TTL@            | Uses the default title from Vocabulary Overrides if the screen program is called from another program.  
  Uses the menu selection text if the screen is called from a menu. |
| ACTION          | Action Code field.  
  - The name assigned by SDA.  
  - The default cursor keyword is assigned to the action code field. |
| *LITER - Literal fields | Literals are added by placing apostrophes around the text on the screen and pressing Enter. (For example, ’V928011’). |
Updating or Adding Fields through SDA

<table>
<thead>
<tr>
<th>Field Description</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>* - Field Definition Window</td>
<td>Allows you to update existing fields and add new fields without using the Pick List feature. Place the * one space to the left of the first character of the requested field to display the Field Definition window.</td>
</tr>
<tr>
<td>&amp; - Field Selection Window</td>
<td>Allows you to add new fields using the Pick List feature. Causes the Field Selection window to display.</td>
</tr>
</tbody>
</table>

To add a field, place an asterisk (*) on the SDA design area where you want to add the field.

To update a field, place an asterisk in the attribute character (first position to the left) of the field you want to update.

You can pull in the screen field, the Row Description/Column Headings (VTX), and a 30 character description field (VC0) all at the same time by making special entries in the Field Name field on the window (*BOTH or *ALL).

Working with Screen Design Aid

To work with Screen Design Aid you must have access to the source file.

To work with Screen Design Aid

1. Inquire on a screen in SVR.
2. Copy the production source code down to a development environment using selection 3.
3. Choose option 10 to access the appropriate Design Aid screen based on the member’s Function Code value.
Function Key Exits

**F12 - Return to Previous Panel**

F12 - Exits you out of the current screen or utility and returns to the screen you were on previously.

- Use F12 instead of F3; however, if you are calling another program outside of SDA (for example: F13, F24), you must use F3 to return to SDA.

Updating an Existing Field

To update an existing field

Place an asterisk (*) in front of the field (in the attribute character). This will bring up the Field Definition window.
<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dict Name</td>
<td>Identifies the four-byte data item name from the Data Dictionary.</td>
</tr>
<tr>
<td></td>
<td>This is the only required field for most data items, the rest will default.</td>
</tr>
<tr>
<td>Text</td>
<td>Describes the Dictionary Name.</td>
</tr>
<tr>
<td></td>
<td>VTX fields contain the soft coded description from the Data Dictionary that updates F9220 (Vocabulary Overrides).</td>
</tr>
<tr>
<td>Data Type</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>All JD Edwards World fields are defined as A.</td>
</tr>
<tr>
<td>Field Name</td>
<td>Identifies a screen field name.</td>
</tr>
<tr>
<td></td>
<td>*VTX (VTX001-VTX200) automatically assigns next available.</td>
</tr>
<tr>
<td></td>
<td>*VC0 (VC0001-VC0200) automatically assigns next available.</td>
</tr>
<tr>
<td></td>
<td>*LITER literal fields.</td>
</tr>
<tr>
<td></td>
<td>*BOTH or *ALL to bring in screen (VD), VC0, and/ or VTX fields.</td>
</tr>
<tr>
<td>Row/ Column</td>
<td>Two 3-digit fields that define the row and column location of field.</td>
</tr>
<tr>
<td>Field Use</td>
<td>How the data is to be used on the screen.</td>
</tr>
<tr>
<td></td>
<td>I</td>
</tr>
<tr>
<td></td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>M</td>
</tr>
<tr>
<td>Size</td>
<td>Two fields identify the length of the data item and for numeric fields, the decimal places.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Text Form</td>
<td>For VTX fields, identifies the field from the Data Dictionary that is used for headings.</td>
</tr>
<tr>
<td></td>
<td>R</td>
</tr>
<tr>
<td></td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>D</td>
</tr>
</tbody>
</table>
### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dft Cursor</td>
<td>Starting cursor position on a data entry screen, Y or N.</td>
</tr>
<tr>
<td>Edited</td>
<td>Should the field be checked for error conditions, Y or N.</td>
</tr>
<tr>
<td></td>
<td>- Will assign an indicator for error handling and default Condition Indicator information.</td>
</tr>
<tr>
<td></td>
<td>- Assigns error indicators 40-79.</td>
</tr>
<tr>
<td></td>
<td>- Key fields, K. Assigns indicator 41.</td>
</tr>
<tr>
<td>Lower Case</td>
<td>To allow lowercase, Y or N.</td>
</tr>
<tr>
<td>Change</td>
<td>CHANGE keyword is in effect, Y or N. The indicator will be set on whenever the value in this field is changed.</td>
</tr>
<tr>
<td>OVERDTA</td>
<td>OVRDTA keyword is in effect, Y or N. Used with PUTOVR to override data that is in a field already on the screen.</td>
</tr>
</tbody>
</table>

**Note:** You should edit all input capable fields. (There will be a “Y” or “K” in the “Edited” field).

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Duplicate</td>
<td>Duplicate the data. Only valid for a SFL format. Puts the DUP keyword in the screen/report DDS but the Program Generator does not generate any code to enable this.</td>
</tr>
<tr>
<td>OVRATR</td>
<td>OVRATR keyword is in effect, Y or N. Used with PUTOVR to override display attributes of a field on the screen.</td>
</tr>
<tr>
<td>Field Cond</td>
<td>Field Conditioning Indicators. Determines if the user can see the field or not.</td>
</tr>
<tr>
<td>Condition Indicators</td>
<td>To set a condition indicator on a field, enter a Y in the first blank to the right of the desired condition. You have the option of entering up to 3 indicators to be associated with the condition. Three spaces are provided to allow an N prior to the two digit indicator to create a negative condition. The allowed conditions are:</td>
</tr>
<tr>
<td></td>
<td>RI</td>
</tr>
<tr>
<td></td>
<td>HI</td>
</tr>
<tr>
<td></td>
<td>UL</td>
</tr>
<tr>
<td></td>
<td>ND</td>
</tr>
<tr>
<td></td>
<td>BL</td>
</tr>
<tr>
<td></td>
<td>PR</td>
</tr>
<tr>
<td></td>
<td>PC</td>
</tr>
<tr>
<td></td>
<td>A blank or N will deactivate the condition.</td>
</tr>
</tbody>
</table>
**Color**

F8 toggles to display the color attributes for the field. The first blank to the right of each color controls the order that multiple colors will appear in the DDS (1-7). If multiple colors are defined, the first enabled color appears and the remaining colors are ignored. A blank or N disables the color. The color values default based on whether you selected JD Edwards World or SAA colors in QJDF.

---

**Accessing Fast Path Create for a New Screen**

When you design the format for a new screen, you have the option to use Fast Path Create.

**To access Fast Path Create for a new screen**

1. Locate your screen and enter selection 10.

   If SDA cannot find the existing DDS for your screen, the Create New Screen screen displays:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Screen and Text Description</td>
<td>Taken from the SVR entry for this member.</td>
</tr>
<tr>
<td>Fast Path Create</td>
<td>Automatically create record formats, fields, file, and record level parameters.</td>
</tr>
<tr>
<td>Window</td>
<td>Screen is a window.</td>
</tr>
<tr>
<td>Wide Screen</td>
<td>Screen is in wide format (132 columns by 27 rows) or normal format (80 columns by 24 rows).</td>
</tr>
<tr>
<td>Subfile</td>
<td>Create subfile format.</td>
</tr>
</tbody>
</table>
17B Work with Screen Design Aid

### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subfile Fold</td>
<td>Create a fold area in the subfile using SFLDROP and SFLFOLD keywords.</td>
</tr>
<tr>
<td>Subfile Clear</td>
<td>Use SFLCLR (Y) OR SFLINZ (N).</td>
</tr>
<tr>
<td>Selection Exits</td>
<td>Create selection exits to allow the user to exit the program using selection codes.</td>
</tr>
<tr>
<td>PUTOVR</td>
<td>The screen record format uses the PUTOVR keyword. Causes the screen to be erased and redisplayed when a window is displayed.</td>
</tr>
<tr>
<td>OVERLAY</td>
<td>The screen record format uses the OVERLAY keyword. Will not erase and redisplay screen when a window is displayed. Most JD Edwards World screens use OVERLAY.</td>
</tr>
</tbody>
</table>

2. Press Enter and SDA begins the creation of your screen based on what you specified.

**Example - Screen with Action Code and No Subfile**

```
92700
Item Maintenance

Action Code. . . B

F24=More Keys
```
Example - Screen with Action Code and Subfile

```
92700                  Item Maintenance
Action Code . . B

SPLCTL
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD
DELETE THIS FIELD

P24=More Keys
```

Example - Screen with Action Code, Subfile, and Selection Exits

```
92700                  Item Maintenance
Action Code . . B

O
P
B
B
B
B
B
B
B
B
B
B
B
B
B
B

P24=More Keys
```
Adding Fields without Using a Pick List

To add a Screen Text Field (VTX)

1. Place an asterisk (*) on the SDA design area where you want to place the screen text field.

2. When the Field Definition window displays, do the following:
   - In the Dict Name field, enter the Data Dictionary item name.
   - In the Field Name field, specify *VTX.
     - The system assigns the next available VTX number.

3. Enter a value in the Text Form field to indicate whether the row description or a column heading from the Data Dictionary should be used as the text.
   - R - Row Description.
   - C - Column Heading 1.
   - D - Column Heading 2.

   **Note:** Text defaults from the Data Dictionary based upon the Text Form value. The default value is R for non-subfile formats.

4. Enter a value in the Size field only if you want to override the default length of 16 for the Row Description that will be brought in.

   **Note:** You should start your fields in column two (unless selection exits exist). This allows you to place an asterisk to the left of the first field in column one.

To add a Database Screen Field (VD)

1. Place an * on the SDA design area where you want the field to be placed.
2. On the Field Definition window, enter the Data Dictionary item name in the Dict Name field.

3. Specify a field use:
   - The default for field use is O for output.
   - Editing indicators are not assigned for output fields.

4. The Data Type, Size, and Text default from the Data Dictionary.

**To add a Screen Constant Field (VC0)**

1. Place an * on the SDA design area where you want to place the description or constant field.

   - The Field Definition window displays.

2. On the Field Definition window, specify *VCO in the Field Name field.
   - The system assigns the next available VC0 number.
3. To override the default length of 30, enter a value in the size field.
Adding a Literal Field

JD Edwards World standard is that the only literal on a screen is the program ID in the top left corner.

To add a literal field

1. Enter the literal text in the SDA Design area.
2. Enclose the text within single quotes.
3. Press Enter.

Using the *BOTH and *ALL Features

The Field Definition window allows you to enter some special keywords in the Field Name field. Two of these special keywords are *BOTH and *ALL.

This feature provides for placement of multiple fields with a single entry.

Using *BOTH

If you use the keyword *BOTH with a valid data dictionary item, screen design will place a VTX field and a screen (VD) field on the screen.
To use *BOTH

On Field Definition

![Screen Design Aid](image1)

Type "*BOTH" in the Field Name

When you enter *BOTH, the following screen displays in the SDA design area for the Unit of Measure field:

![Screen Design Aid](image2)

Using *ALL

If you use the keyword *ALL with a valid Data Dictionary item, screen design places a VTX field, a screen (VD) field, and a VC0 field on the screen.
To use "ALL"

On the Field Definition window

Type "ALL" in the Field Name.

When you enter "ALL", the following screen displays in the SDA design area for the Unit of Measure field:

Field Default Values

The following are field defaults in SDA:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VD - Screen Display</td>
<td>- Output only&lt;br&gt;You can enter a B in the Field Use field to override the default and change it to both input and output. &lt;br&gt;- No Editing&lt;br&gt;If you enter B in the Field Use field, the Edited field defaults to Y. The Condition Indicators default to Y and the next available editing indicator is assigned to that field.</td>
</tr>
</tbody>
</table>
**Field** | **Explanation**
---|---
VTX - Screen Text field | 16 bytes long
| Defaults to Row description rather than column description
VC0 - Screen Constant field | 30 bytes long

**Understanding the SDA Exit/Save Function Key**

**F3 - Design Aid Exit/Save**

F3 - Saves, exits, or does both from Screen Design Aid.

```
92590 Design Aid Exit/Save

Save DDS (Y/N) ... N
Member ID. . . . . . VS28200
File ID. . . . . . JTBREC
Src Library. . . . STRJTPR1LOAD
Description. . . . Item Search
Function Code. . . DSPF
Return to Edit (Y/N) N
```

**Field** | **Explanation**
---|---
Save DDS (Y/ N) | Saves the DDS and updates or creates Vocabulary Overrides and Function Key definitions.
Member ID | The record of the Software Versions Repository member to be copied.
Screen-specific information | Name of the screen.
File ID | Identifies the file that contains the source code.
Src Library | Identifies the library where the source code resides.
Description | The description of a record in the Software Versions Repository file. The member description is consistent with the base member description.
Function Code | Should be DSPF for a Screen.
Return to Edit (Y/ N) | EOJ or allows return to SDA.
Compiling Your Screen

To compile your screen

From the Software Versions Repository screen

Enter 14 next to the member in the subfile to create.

Screen Design Standards and Tips

Title

A screen title is limited to 30 characters and should match the Software Versions Repository (F9801). The title you enter in SDA updates the vocabulary overrides record for the screen. If you access the screen using a menu selection, the menu selection name overrides the screen title. If you access the screen using a selection option or function key, the vocabulary overrides title is used.

Line 24

You should document function keys on the right side of line 24 and you should document options on the left side. The following guidelines should also be followed:

- List both the options and function keys in numeric order.
- F24 should always appear and should say MORE KEYS or MORE.
- F4 should always read MORE DETAIL or DETAIL.
- Do not include standard exits of F3, F7, F22, Help, Rollup, Rolldown.
- Line 24 should be in reverse image during an error condition. Line 24 is conditioned to appear in reverse image on screens based on indicator 93.
- If you specify *SAME for the field “Error Text” for Line 24 in vocabulary overrides, then the text displayed is the same as the text specified for the normal Line 24.
Windows

Within a window, line 24 should include F3 and F24 when the window is initially displayed. When designing windows in SDA, fill in unused line space with literal fields to prevent data on the calling screen from showing through on the window. You can add the literal fields on the SDA screen as blanks with a single quote on each end or through the Field Definition window.

| 4038 Allen, Ray Jr. | 4072 Allen, Marilyn |
| 4039 Allen, Marilyn |
| 4031 Allen, Cindy |
| 4037 Allen, Cindy |
| 4038 Allen, Ray Jr. |
| 4037 Allen, Cindy |
| 4039 Allen, Marilyn |
| Date Of Birth: 06/13/48 |
| SSN. 422-51-4668 |
| Dep/Ben Status |
| Memo/Address: 7505 R. 11th Avenue |
| Denver, Colorado |
| Opt:1-Single Assignment |
| 2-Add/Rev D/E |
| F1-Detail F21-Print F34-More Keys |

Default Cursor

You set the default cursor attribute to Action Code for the input field closest to the upper-left corner of the screen.

Fold Area

Keep the number of Fold Area lines to no more than two lines to avoid excessive use of the cursor keys when the Fold Area is open.

Description Fields

Define all description input fields to allow for uppercase and lowercase letters DDS keyword CHECK(LC). Use VCO descriptions when a field’s value has no obvious meaning and you can retrieve a description from a master file or user defined codes.

Alpha Fields

Because of the dynamic nature of international currency, you must define every field as alpha. The only exception is that you can define hidden fields as numeric. JD Edwards World scrubbing routines handle the two-way conversion between numeric data in the file and formatted alpha screen fields.

General Aesthetics

The following are things you might want to consider when designing screens. They are guidelines that will give your screens a more professional look.
Alignment

Line up fields vertically. This includes row descriptions, input fields, and description fields. Fields on the left side of the screen should be in column space 2 (column 1 is needed for the attribute byte).

Grouping Fields

When entering a descriptive heading to group related fields, use up to 40 characters for the description (or as long as space permits). Highlight the heading and end it with a colon. Underneath the heading, indent the group of fields one space to the right.

Spacing

Use the following as your standards when spacing different screen elements:
- Separate column headings with one space.

- End row descriptions with at least one period followed by a single space before you begin associated input fields.

- Indent Fold Area fields one or more spaces to offset them from regular subfile.

- Use two or more spaces to separate Fold Area data fields from row descriptions that follow on the same line. End Fold Area row descriptions with a colon instead of periods to aid legibility.

- Insert a blank line between header and subfile information.
When possible, insert a blank line between the title and first field. Begin fields on line 3 unless you need to use the upper right corner of line 1 and 2.

Adding Screen Fields Using Pick List

To add screen fields

From the SDA screen:

1. Access the Records Formats List using the F10 key.
2. Complete the Record Formats List screen.
3. Press F10 to display the Record Formats List.

Use this screen to select database fields and maintain record formats, record types, fast path files, and record format keywords.

<table>
<thead>
<tr>
<th>Opt</th>
<th>Format Name</th>
<th>Type</th>
<th>Fast Path</th>
<th>Start / End</th>
<th>Related</th>
<th># Fields</th>
<th>Fld Pfx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>001 001</td>
<td>V9282005</td>
<td>000</td>
<td>VD</td>
</tr>
<tr>
<td>2</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>007 022</td>
<td>V9282005</td>
<td>000</td>
<td>SF</td>
</tr>
<tr>
<td>3</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>4</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>5</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>6</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>7</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>8</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>9</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
<tr>
<td>10</td>
<td>V9282005</td>
<td>SPL</td>
<td>P92801</td>
<td>024 024</td>
<td>V9282005</td>
<td>000</td>
<td>S0</td>
</tr>
</tbody>
</table>

Opt: 1-DB Field Selection  2-Field List  4-Delete  5-Format Keywords

### Field

<table>
<thead>
<tr>
<th>Opt</th>
<th>Explanation</th>
</tr>
</thead>
</table>

Enter the appropriate number to indicate you want to select one of the following values:

- File/ field pick list for ampersand functions.
- File/ field pick list for fast path functions.
- List of defined fields in the format.
- Delete format.
- Record format keywords.
**About Record Formats**

Several Record Format Types are valid for screens. Currently, they include:

- **SFLCTL - Subfile control**

  Present in all subfile screens. Contains all of the fields in the header or top portion of the screen, including the subfile column headings.

- **SFL - Subfile**

  Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 4-65
Contains all the fields in the subfile portion of the screen, including the fold area (if applicable).

- RECORD

Present in all screens. In subfile screens, contains VDL24 (line 24 text). In non-subfile screens, can contain all fields on the screen, including VDL24.

- SFLMSG - Subfile Message

Displays error message text. JD Edwards World does not use this format because errors are handled through RPG programs.

Selecting Database Fields

There are two methods of selecting database fields for placement on the screen:

- With Fast Path
- With the File Selection List

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fast Path</td>
<td>Type 1 next to the format on which you want to place the fields and enter a file name under the Fast Path File column.</td>
</tr>
<tr>
<td>File Selection List</td>
<td>Type 1 next to the format on which you want to place the fields but do not enter a file name. Accesses a file selection screen where you can specify multiple files and libraries from which to select database fields.</td>
</tr>
</tbody>
</table>

To select a database field using Fast Path

1. On the Record Formats List screen, enter a Fast Path File for the specified format.
2. For database field selection, choose option 1.
The Field Selection List displays.

3. Press F21 to select all the fields for the file instead of selecting them individually.

Based on the record format for which you are using the Field Selection List, the following information is the default:

- For a subfile control record format, the HDG field will default to R for the type of heading and the Use field will default to B for input/output.
- For a subfile record format, the HDG field will default to D for the type of heading and the Use field will default to B for input/output.
- For a non-subfile screen, the HDG field will default to R for the type of heading and the Use field will default to B for input/output.
- For a report, the HDG field will default to D for the type of heading and the Use field will default to O for output.

### Field Explanation

<table>
<thead>
<tr>
<th>Seq No</th>
<th>Field Name</th>
<th>Description</th>
<th>DT</th>
<th>Size</th>
<th>HDG</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>QXXIT</td>
<td>Item ID</td>
<td></td>
<td>S</td>
<td>8</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QXXDSC</td>
<td>Description</td>
<td></td>
<td>A</td>
<td>30</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXXTY</td>
<td>Item Type</td>
<td></td>
<td>A</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXXDT</td>
<td>Date Last Ship</td>
<td></td>
<td>S</td>
<td>6</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QXXCC</td>
<td>Business Unit</td>
<td></td>
<td>A</td>
<td>12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXXQT</td>
<td>Quantity On Hand</td>
<td></td>
<td>S</td>
<td>15</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>QXXUM</td>
<td>Unit of Measure</td>
<td></td>
<td>A</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX001</td>
<td>Item Code 001</td>
<td></td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX002</td>
<td>Item Code 002</td>
<td></td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX003</td>
<td>Item Code 003</td>
<td></td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX004</td>
<td>Item Code 004</td>
<td></td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QXX005</td>
<td>Item Code 005</td>
<td></td>
<td>A</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Field Name</td>
<td>The name given to a field for a screen, report, or database table. If the field is a key field in the file, K01, K02, etc will be displayed.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Description</td>
<td>The Data Dictionary row description.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Item Type</td>
<td>The type of data. The data item types are defined in User Defined Codes, system code '98', record type 'DT'.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Item Size</td>
<td>The field size of the data item.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Note:</td>
<td>All amount fields should be entered as 15 bytes, 0 decimals, and the data item type should be P (packed).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HDG</td>
<td>Which heading to use from the Data Dictionary.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Row Description (R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Column 1 heading (C)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Column 1 and 2 heading (D)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Used to indicate whether a 30 character VC field should be included for constant information.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Use</td>
<td>Specifies how the data field is to be used on the screen:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Input only (I).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Output only (O) (default).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Both input and output (B).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- IBM Message field (M).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- Subfile Hidden (H).</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**To select database fields using the File Selection List**

From the Record Formats List screen:

1. Choose option “1” but do not specify a file.
The File Selection List displays.

2. Enter the files from which you want to select fields.

Fields for files requested will be displayed through the Field Selection List screen.

3. Select fields using the same techniques as in the Fast Path method.

Note: If you select a key field, that field is edited as the key of the screen.
An edit indicator of 41 is assigned.
Placing Fields on a Screen Using a Pick List

To place fields on a screen using a Pick List

On the Item Master Information screen

1. Type either one or more ampersands (&) on the SDA screen where you want to place the fields from the pick list you created. If you place more than one &, make sure that you allow room for all of the fields that are returned to the screen, so that you do not overlap fields.

2. On the Field Selection screen, verify the information that is on the screen (VTX field - HDG, 30-character description - D, and field Use - USE), as well as the order that they will be brought back (the sequence number), and row description length.

3. To add a Fold Area to a subfile, place an asterisk (*) or ampersand (&) on the second line in the subfile format of your screen. If you need a second line in the Fold Area, you can place an asterisk (*) or ampersand (&) on the third line of the subfile format. HDG should be “R” when adding fields to the fold.

Function Key Exits from Screen Design Aid

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>Access a command line to enter a JD Edwards World or IBM command without having to exit to Command Entry or a menu. If you are secured out of Command Entry or Menu Traveling, you still get this command line, but you cannot execute commands or menu travel.</td>
</tr>
<tr>
<td>F5</td>
<td>Shows the Format Display Control window.</td>
</tr>
</tbody>
</table>

Example F5 - Format Display Control window.
The following table provides an explanation of the fields in the Format Display Control window.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selection</td>
<td>Controls the display of record formats.</td>
</tr>
<tr>
<td></td>
<td>1 Format is displayed.</td>
</tr>
<tr>
<td></td>
<td>Blank Format is not to be displayed.</td>
</tr>
<tr>
<td>Format</td>
<td>Lists the DDS format names for the screen screen.</td>
</tr>
<tr>
<td></td>
<td>All names begin with Screen name</td>
</tr>
<tr>
<td></td>
<td>- Subfile control formats end with C.</td>
</tr>
<tr>
<td></td>
<td>- Subfile formats end with S.</td>
</tr>
<tr>
<td></td>
<td>- Record (non-subfile) formats end with 1.</td>
</tr>
<tr>
<td>Type</td>
<td>Describes the format type.</td>
</tr>
<tr>
<td>Boundaries</td>
<td>Two 3-digit numbers that define the range (rows) for the DDS.</td>
</tr>
<tr>
<td>Window</td>
<td>Allows access to fields outside the boundaries.</td>
</tr>
<tr>
<td>Browse (Y/ N)</td>
<td>Allows user to enable/ disable the browse mode and view the screen as it</td>
</tr>
<tr>
<td></td>
<td>would appear when executed.</td>
</tr>
<tr>
<td></td>
<td>- Cannot change or access any item while in browse mode.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>F4</td>
<td>Provides the following:</td>
</tr>
<tr>
<td></td>
<td>- Toggle between displaying the Fold Area or not for a subfile screen.</td>
</tr>
<tr>
<td></td>
<td>- Must set Browse (Y) in Format Display Control Window (F5).</td>
</tr>
</tbody>
</table>

The table above provides an explanation of the fields in the Format Display Control window.

**Item Search**

<table>
<thead>
<tr>
<th>Item Number</th>
<th>Description</th>
<th>Date</th>
<th>Quantity On Hand</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>938290</td>
<td>Item Search</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Business Unit:** [Redacted]

**Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 4-71**
Function | Description
---|---
F6 | This screen provides access to other repository services within JD Edwards World.
F8 | Provides the following:
| ▪ Will display your screen in monochrome or color.
| ▪ If accessing the Field Definition window, will toggle between Condition Indicators and Color Attributes.
F10 | Displays the Format Keyword Maintenance screen.

Example F10 - Format Keyword Maintenance screen

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PUTOVR (Y/N)</td>
<td>The screen record format uses the PUTOVR keyword. Causes the screen to be erased and redisplayed when a window is displayed.</td>
</tr>
<tr>
<td>OVERLAY (Y/N)</td>
<td>The screen record format uses the OVERLAY keyword. Will not erase and redisplay screen when a window is displayed. Most JD Edwards World screens use OVERLAY.</td>
</tr>
<tr>
<td>Subfile Fold</td>
<td>Create a fold area in the subfile using SFLDROP and SFLFOLD keywords.</td>
</tr>
</tbody>
</table>
| Type (A/F) | Further identifies subfile fold area:
| | A Will lose modified data in the subfile when you press F4.
| | F Data is retained. |
| Subfile Clear | This option specifies if you want to use SFLCLR or SFLINZ. The default is SFLCLR. This option is ignored when designing non-subfile screens.
| | Y means you want SFLCLR
| | N will give you SFLINZ |
### Field Explanation

**Subfile Next Change**
Whether or not to use SFLNXTCHG (Y/N). Will require the user to correct any errors in the subfile before further execution of the program.

**Subfile Page**
Identifies the number of records on one subfile page, with the fold area open, if applicable.
- 1 to 27 inclusive (Number of lines in SFL area divided by number of lines in fold.)

**Subfile Size**
Identifies the total number of records in the subfile that will be loaded in one program cycle.
- 1 to 9999 inclusive

### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F13</td>
<td>Displays the Function Key/Opt Definition screen.</td>
</tr>
</tbody>
</table>
- Used to define the function keys for the screen. |
- Function Key Definition files (F9601 and F9611).

**Example F13 Function Key/Opt Definition screen**

```
#601 Function Key/Opt Definition
Action Code...I Video Screen...V92000.0
Video Title...I Item Search
Line 24
Opt:1=Item Master Information  F5=Item Maintenance  F24=More Keys

Include Description Key/Opt Field
Y Exit Program 03 $FEOJ
Y Clear Screen 22 $FCLR
Y Help Instructions NLI $FHELP
Y Roll Up/Next Record 2F $FROLJ
Y Roll Down/Previous Record 2D $FROLD
Y Field Descriptive Help 01 $FQRESH
Y Display Error Message(s) 07 $FERRRD
Y Display All Function Keys 24 $FERRD
Y Item Maintenance 01 $F01
Y Item Master Information 01 $F01

Include: Y/N  F16=Display All
```

### Field Explanation

**Action Code**
One character field used to indicate the action that the user wants to take on the record requested.
- Inquire on a record before you attempt to change it.

**Screen Screen**
The name of the screen or report record to be copied. All records for soft coding will be displayed.
### Field Explanation

**Screen Title**
The vocabulary overrides title used on screens and reports. On screens, the title is retrieved from the Menu file. If a record is not found, then the title is retrieved from the Vocabulary Overrides file. Report titles are retrieved from the DREAM Writer Version ID (F98301).

**Include**
Whether or not to include the function or option key on the screen.

**Description**
Describes the function or option exit. Cannot exceed 40 characters.

**Key/Opt**
Identifies the function key number or option.

- Special values:
  - Helps.
  - Roll Up.
  - Roll Down.

**Field**
Identifies the name of the function or option exit. Values always begin with a # (pound sign).

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F14</td>
<td>Displays the Indicator Control portion of a simulated program execution screen.</td>
</tr>
<tr>
<td></td>
<td>- Used with the Browse mode to simulate a screen at program execution.</td>
</tr>
<tr>
<td></td>
<td>- Desired indicators can be set on/off to test field conditioning.</td>
</tr>
</tbody>
</table>

Example F14 - Indicator Control portion of a simulated program execution screen.

```

```

---

4-74  JD Edwards World, A9.1
Function Description

F16 Displays the List of Defined Fields screen.

- Used to maintain the defined fields and add hidden fields.
- Only shows fields for the formats that are active.

Example F16 - List of Defined Fields screen.

<table>
<thead>
<tr>
<th>Opt</th>
<th>Fmt/Field</th>
<th>Description</th>
<th>Row/Col</th>
<th>Typ</th>
<th>Size</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VTX007</td>
<td>Item</td>
<td>006 013</td>
<td>A</td>
<td>30</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX009</td>
<td>Quantity</td>
<td>006 044</td>
<td>A</td>
<td>21</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX011</td>
<td>Ship</td>
<td>006 066</td>
<td>A</td>
<td>0</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX003</td>
<td>P</td>
<td>007 002</td>
<td>A</td>
<td>1</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX006</td>
<td>Number</td>
<td>007 004</td>
<td>A</td>
<td>5</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX008</td>
<td>Description</td>
<td>007 013</td>
<td>A</td>
<td>30</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX010</td>
<td>On Hand</td>
<td>007 044</td>
<td>A</td>
<td>21</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>VTX012</td>
<td>Date</td>
<td>007 066</td>
<td>A</td>
<td>8</td>
<td>O</td>
</tr>
<tr>
<td></td>
<td>V92801G</td>
<td>Record Format</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SMXIT</td>
<td>Item ID - Hidden Field</td>
<td>000 000</td>
<td>A</td>
<td>R</td>
<td>H</td>
</tr>
<tr>
<td></td>
<td>SFHEL</td>
<td>Selection Exits</td>
<td>008 002</td>
<td>A</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>SFXIT</td>
<td>Item ID</td>
<td>008 004</td>
<td>A</td>
<td>0</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>SFXDS</td>
<td>Description</td>
<td>008 013</td>
<td>A</td>
<td>30</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>SFXOT</td>
<td>Quantity On Hand</td>
<td>008 044</td>
<td>A</td>
<td>21</td>
<td>B</td>
</tr>
</tbody>
</table>

Opt: 4=Delete 5=Display/Update F3=Exit F12=Prev Screen

Hidden Fields

Used to store hidden field information.

To add a hidden field to a screen

1. Roll to the bottom blank line of the format that contains the field.
2. Choose option 5, Display/Update.
3. Enter the field with a prefix of SH, description, type, size, and press Enter.
   This information should be the same as the associated database field.
**Option 5 - Select Field Definition**

Accesses the Field Definition screen, just as if you had entered an asterisk (*) for the field.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F17</td>
<td>Used to define soft coding (Vocabulary Override) fields.</td>
</tr>
<tr>
<td></td>
<td>▪ To define VTX fields other than row and column headings on the screen. This is used to create a VTX field which stores a message which the application program uses for display purposes on a screen or report.</td>
</tr>
<tr>
<td></td>
<td>▪ You can specify the literal text that will be loaded into a *VC0 field.</td>
</tr>
<tr>
<td></td>
<td>▪ You must save your screen at least once in order to be able to update vocabulary overrides and/or function key definitions by this method. This is because when you are first defining a screen, the vocabulary override record and function key definition record are not created until you save the screen.</td>
</tr>
</tbody>
</table>

**Note:** After changing the size of a VTX field, you should execute the Rebuild Vocabulary Override Field Lengths program (11/ G9642). This displays the correct VTX field lengths in the Field Size field in Vocabulary Overrides.
An example F17 - Define Soft Coding Fields screen:

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>Description</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>CO</td>
<td>O</td>
<td>Business Unit</td>
<td>VTX001</td>
</tr>
<tr>
<td>CR</td>
<td>P</td>
<td>Item Description</td>
<td>VTX002</td>
</tr>
<tr>
<td>CR</td>
<td>E</td>
<td>Quantity On Hand</td>
<td>VTX003</td>
</tr>
<tr>
<td>CR</td>
<td>N</td>
<td>Number</td>
<td>VTX004</td>
</tr>
<tr>
<td>CR</td>
<td>Q</td>
<td>Date</td>
<td>VTX005</td>
</tr>
</tbody>
</table>

**Function Description**

- **F19**: Used to window left.
  
  **Note**: This function key is applicable only when designing wide screens (132 by 27 rows) on 80 column terminal.

- **F20**: Used to window right.
  
  **Note**: This function key is applicable only when designing wide screens (132 by 27 rows) on 80 column terminal.

### Changing Subfile Boundaries

Be careful when changing the size of a subfile through SDA. Consider using these processes to make such changes easier and less confusing.

**To make a subfile smaller**

1. To access the Record Formats List screen, press F10.
2. Change the starting line number for the subfile form at (VxxxxxS).
3. To return to SDA, press Enter.
4. To access the Record Formats List screen again, press F10.
5. Change the ending line number for the control format (VxxxxxC).
6. To return to SDA, press Enter.
7. Move or add headings.
To make a subfile larger

You can reverse the above steps if you want to make the subfile larger. You must move the control format fields up before changing the starting line number for the subfile format.

1. Move headings.
2. To access the Record Formats List screen, press F10.
3. Change the ending line number for the control format (VxxxxxC).
4. To return to SDA, press Enter.
5. To access the Record Formats List screen again, press F10.
6. Change the starting line number for the subfile format (VxxxxxS).
7. To return to SDA, press Enter.
8. In SDA, press F10 to alter the format.
9. Enter 5 on the control format field.
10. Change subfile page size, as needed.

**Note:** When you change the subfile, you must change the subfile page and subfile size to correctly reflect the size of the new subfile.

### Process Overview - Placing Selected Fields

Once you have established your field pick list, use the ampersand (&) to specify where you want to locate the field.

The ampersand (&) calls up the pick list in the Field Selection screen where you can order the fields and further define their specifications.

### Options

The following options are available:

- Override Row Description length
- Resequence fields in list
- Select headings (Row, Column headings) *VTX
- Description Field (*VC0)
- Usage (O=Output, B=Both Input and Output)

After you have sequenced the fields, they are retrieved from the file and placed on the design area.

```
  92S011  Item Master Information
  Action Code.    .  E
  Item ID.        .  RRRRRRRR
  Business Unit.  .  RRRRRRRRRRR RRRRRRRRRRR RRRRRRRRRRR RRRRRRRRRRR RRRRRRRRRRR
```

<table>
<thead>
<tr>
<th>Screen: V92S200</th>
<th>Field Selection List</th>
<th>Format: V92S200C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seq</td>
<td>Fields to select</td>
<td>Row Desc Length</td>
</tr>
<tr>
<td>No. Field Name</td>
<td>Description</td>
<td>C7 Size NC D Use</td>
</tr>
<tr>
<td>001</td>
<td>OGS17_01 Item ID.</td>
<td>12 E D B B</td>
</tr>
<tr>
<td>002</td>
<td>OGS17_BCA Business Unit.</td>
<td>12 E D B B</td>
</tr>
</tbody>
</table>

F3-Exit  F10-Formats  F13-Prev Screen  F16-Field List
Process Overview - Revising the Field Definition

SDA Design Area
Place asterisk in the field’s attribute byte to revise

Field Definition

List of Defined Fields
Use the field definition information to display or create attributes for the data item.

Option 3
Option 5 next to displayed

F16 OR

F10

F8

From the Field Definition information to display color attributes for the data item.
Process Overview - Revising Vocabulary and Function Keys

SDA Design Area

Function Key/Opt Definitions

F13 = Use the Function Key/Opt Definition screen to define the function keys for the form.

F17 = Use the Define Soft Coding Fields form to define VTX fields other than row and column headings.

Function Keys for Screen and Display Format Control

SDA Design Area

Format Display Control

Browse (Y/N)

F4 = The form will show a subfile form that has a fold area in its folded & unfolded formats. The form must be in browse mode for this to function.

F14

F5

User can turn indicators on and off to see how the form will look.

Format Display Control

- Put the form in browse mode
- Activate or deactivate formats
- Enable the F4 key to see a subfile form in the folded or unfolded format.
Summary of Screen Design Aid

Editing options:
- d, *DEL
- <<, >>
- ‘xx...xx’
- -, =
- - -, =
- * and &

You should not use the INSERT and DELETE keys while in SDA.

F7 restores a screen if you accidentally press Field Exit.

Standard prefixes:
- VD, SF, SH

Special Fields:
- ACTION
- VDL24
- TTL@

Error indicators 40 to 79 are automatically assigned to VD and SF fields that are defined as input or input/output.

Update fields by using *.

You have two methods of adding fields to a screen:
- * (non-pick list method)
- & (pick list method)

You can pull in VTX, VC, and the screen database fields all at the same time for one database field.

You have two methods of selecting database fields:
- Fast Path
- Non-Fast Path - Accesses File Selection screen

If you are changing subfile boundaries, you should use the outlined processes to make this process easier.

You must save a screen at least once before updating vocabulary overrides or Function Key Definitions because the exit from SDA creates these records.

You add hidden fields from the List of Defined Fields screen, which you access by pressing F16 from SDA:
- You add hidden fields one at a time.
- You must enter a selection exit 5 to actually add the field.
About Report Design Aid

Report Design Aid (RDA) is a powerful and versatile tool for designing reports. It uses the same process as the Screen Design Aid (SDA), except:

- It extends to column 227
- It has windowing capability

You need to identify only field names, field lengths, and field positions on the report.

JD Edwards World reports are externally defined, which means that all the Data Description Specifications are created and compiled as a printer file, separate from the program object. RDA automatically generates the DDS. It also incorporates the report information into the documentation and adds it to the cross reference facilities. You can print illustrations of each report.

RDA differs from SDA in that its parameters are targeted for print-based output, which includes page skipping, line skipping, and relative positioning.

This section includes the following tasks:

- Accessing Report Design Aid
- Updating a Field in RDA
- Compiling A Report
- Changing the Compile Option Defaults for Reports
Example - RDA and DREAM Writer

Comparing RDA and SDA - Field Definition Form

```
Report:R5540CX-------------------Field Definition-------------------Format:--DETAIL--
Dict Name: XXDS Text Description:-------------------Rule:---
Row/Column _  09 Field Use : O __________ Highlight __________
  Field Name: REXXDS __________ Underscore __________
  Size: i 12 __________ Text Form __________ Char per Inch __________
  Lines: Cond Ind __________ Edit Code __________
Space Before: __________ Field Cond __________
Space After: __________ Asterisk Pad __________
Skip Before: __________ Float Symbol __________
Skip After: __________
--F3=Exit  F12=Prev Screen  F17=Dictionary-------------------------------------
```

Screen: V5501S---------------------Field Definition---------------------Format: V5501SS
Dict Name: XXDS Text Description:-------------------Cond Ind
Data Type: A Field Name: TPXK01 __________
Row/Column _  13 Field Use: A __________ RI Y  44 __________
  Size: __________ Text Form __________ MI Y  44 __________
  Def: Cursor _ __________ Edited Y  44 __________ NL Y N44 __________
  Lower Case Y __________ Change __________ PN __________
  OVRDTA __________ Duplicate __________ PR __________
  OVRATR __________ Field Cond __________ PC __________
--F3=Exit  Fl2=Prev Screen  Fl7=Dictionary-------------------------------------
Work with Report Design Aid

<table>
<thead>
<tr>
<th>Item</th>
<th>FIELD POSITIONING</th>
<th>FIELD CONDITIONING</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDA</td>
<td>Row positions are relative to the other rows, not fixed. The location on the report is determined by Space and Skip designations. Column positions are fixed.</td>
<td>A field can optionally appear bold, underlined, and so forth. JD Edwards World does not typically use these features because they impact printer performance.</td>
</tr>
<tr>
<td>SDA</td>
<td>Both row and column positions are fixed. A field displays on the screen exactly where the Row and Column indicators specify.</td>
<td>A field can appear highlighted, underlined, in reverse image, and so forth JD Edwards World makes use of these attributes for marking fields in error.</td>
</tr>
</tbody>
</table>

Cover Page Fields

The figure below shows the fields used on the cover page of a report. These fields would indicate your company in a production environment.

The following table provides the field names and a description of each.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC0CO</td>
<td>Name of company 00000</td>
</tr>
<tr>
<td>TTL@</td>
<td>Line 1 of DREAM Writer Version ID if it exists, otherwise it is blank</td>
</tr>
<tr>
<td>TXT2</td>
<td>Line 2 of DREAM Writer Version ID, or blank</td>
</tr>
<tr>
<td>TXT3</td>
<td>Line 3 of DREAM Writer Version ID, or blank</td>
</tr>
</tbody>
</table>

Report Header Fields

The figure below shows the fields used on the report header. These fields would indicate your company in a production environment.
The following table provides the field names and a description of each.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>VC0CO</td>
<td>Name of company 00000</td>
</tr>
<tr>
<td>RRTTL@</td>
<td>Default Title from Vocabulary Override</td>
</tr>
<tr>
<td>RRTXT2</td>
<td>Line 2 of DREAM Writer Version ID, or blank</td>
</tr>
<tr>
<td>RRTXT3</td>
<td>Line 3 of DREAM Writer Version ID, or blank</td>
</tr>
</tbody>
</table>

What Are the Report Formats?

The first step in designing a new report is determining the format of the report. You should account for all lines of information on the report to correctly define the formats needed and their size.

<table>
<thead>
<tr>
<th>FORMAT</th>
<th>FIELD</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Any format</td>
<td>*VTX</td>
<td>Assigns the first available VTX name to the field and gets a description from the Data Dictionary that you can change.</td>
</tr>
<tr>
<td></td>
<td>*VC0</td>
<td>Assigns the first available VC0 field and assigns a default size of thirty.</td>
</tr>
<tr>
<td>FORMAT</td>
<td>FIELD</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td>HEADING1 - contains the standard fields to be printed on the top of every page</td>
<td>VTX001</td>
<td>The default VTX field which prints the row description, “Page.”</td>
</tr>
<tr>
<td></td>
<td>*PAGE</td>
<td>The default special field that inserts the DDS keyword PAGNBR in the source and retrieves the current page number on the report.</td>
</tr>
<tr>
<td></td>
<td>VTX002</td>
<td>The default VTX field which prints the row description, “Date -”.</td>
</tr>
<tr>
<td></td>
<td>*DATE</td>
<td>Special field that retrieves today’s date.</td>
</tr>
<tr>
<td></td>
<td>VC0CO</td>
<td>The name of the default company; it displays on the first line of each page.</td>
</tr>
<tr>
<td></td>
<td>RRTTL@</td>
<td>Default Title from Vocabulary Overrides.</td>
</tr>
<tr>
<td></td>
<td>RRTXT2 &amp; RRTXT3</td>
<td>DREAM Writer overrides that correspond to the second and third header lines of the report.</td>
</tr>
<tr>
<td>HEADING2 - contains the subheading fields used to describe the detail that is to follow</td>
<td>VC0ROW</td>
<td>Data Dictionary row description of the level break field.</td>
</tr>
<tr>
<td></td>
<td>VC0KEY</td>
<td>The value of the level break field.</td>
</tr>
<tr>
<td></td>
<td>VC0DSC</td>
<td>The description of the value of the level break field.</td>
</tr>
<tr>
<td>DETAIL1 - contains the data line fields</td>
<td>RRxxxx</td>
<td>The value of the data for these fields</td>
</tr>
<tr>
<td>TOTAL1 - contains the total line fields</td>
<td>VC1ROW</td>
<td>Data Dictionary row description of the level break field.</td>
</tr>
<tr>
<td></td>
<td>VC1KEY</td>
<td>The value of the level break field</td>
</tr>
<tr>
<td></td>
<td>VC1DSC</td>
<td>The description of the level break field</td>
</tr>
<tr>
<td></td>
<td>$$XXX</td>
<td>Value on total line.</td>
</tr>
</tbody>
</table>

**Note:** You can have as many formats as you can fit on one RDA form. Just remember to increment the suffix number for each format added as well as any VC fields you may be using.
Certain fields are used in RDA when generating reports that contain subheadings or dynamic (hierarchical) totaling. The following illustrates how these fields are used within a report.

What Are the Report Design Standards?

The following is a list of report design standards. Using these standards will give your reports a uniform appearance.

RDA Features

Some features of RDA are:

- Normal design range of 132 - 198 character reports
- Validates against the Data Dictionary
- Automatically adds records to the Vocabulary Overrides file

JD Edwards World Standards for Record Formats

Prefix standards:

- RR for output fields
- $$ for total fields
General Aesthetics

When possible, design your reports using the following set of rules:

Column Headings

Column headings should not be wider than the length of the data that appear below them.

Alignment

Begin fields in column space 2 and do not extend fields beyond column 132 unless necessary.

Spacing

Use the following as your guides when spacing different report elements:

- Separate column headings by one space
- Use both column headings when one heading is not clear enough

Special Effects

Some of the following recommendations can be ignored since most printers in use are not impact printers.

You should always use dashes below column headings instead of underlines. Underlines can impact the performance of printers. You enter dashes as literal fields.

Do not use highlight as it prints a line three times to achieve the highlighted (or boldface) effect, again impacting performance.

Format

To avoid overflow, limit the number of lines in any detail or total format to six or less.

Line and Page Skipping

To be consistent with other report programs, use SPACEB and SKIPB instead of SPACEA and SKIPA.

About Designing the Report

DDS statements are being created as you design the report.

- SPACEB and SPACEA are entered and removed as you add and move fields around.
- Multiple formats are relative to each other.
### Accessing Report Design Aid

You must have access to the source file to enter RDA.

#### To access Report Design Aid

From Software Versions Repository

```
9801  Software Versions Repository
Action Code. . . I
Member ID . . . PD8400
Description. . . Inventory by Cost Center w/o Subheadings
Function Code . . FRTF Printer Files
Function Use . . 164 Simple Reports
System Code . . 32 Computer Assisted Design
Reporting System 32 Computer Assisted Design
Base Member Name PD8400 File Prefix . .
Maint/RESDSP . 1 Omit Option . . 5 Generation Serv.
   Source Object Source SAR Version S D User Date
   UDPS0171 UDPS0171 12345678 1/1 QUARLES 10/26/94
   ----------------- ------------------ -----------------------
   1-Browse 2-Edit 3-Copy 5-SAR 6-Print 9-Dlt 10-Design 14-Crt  F24-More
```

1. Inquire on a report.
2. Copy the production source code down to a development environment.
3. Choose option 10 on the Software Versions Repository form to go to the appropriate Design Aid form based on the members Function Code value.
   - To go to Report Design Aid, enter “PRTF” or “PRTS” in the Function Code field.

### Updating a Field in RDA

The field definition form in RDA is slightly different from SDA.
To update a field in RDA

From Software Versions Repository, choose the design option.

1. Enter "*" in the field you wish to update.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Space Before</td>
<td>Specifies the number of lines a printer device is to space before printing the next lines.</td>
</tr>
<tr>
<td>Space After</td>
<td>Specifies the number of lines a printer device is to space after printing the next lines.</td>
</tr>
<tr>
<td>Skip Before</td>
<td>Specifies that the printer device is to skip to a specific line number before it prints the next lines.</td>
</tr>
<tr>
<td>Skip After</td>
<td>Specifies that the printer device is to skip to a specific line number after it prints the next lines.</td>
</tr>
<tr>
<td>Field Cond</td>
<td>Indicates whether the field conditioning (to print this field or not) is in effect.</td>
</tr>
<tr>
<td>Char per Inch</td>
<td>Specifies the horizontal printing density. JD Edwards World specifies this at the report level and this field is not used.</td>
</tr>
<tr>
<td>Edit Code</td>
<td>Used to specify output formatting of numeric data. Used in conjunction with *DATE, *TIME, *PAGE.</td>
</tr>
<tr>
<td>Asterisk Fill</td>
<td>Optionally specify asterisk fill for edit codes 1-4, A-D, and J-M. An asterisk will print for each zero suppressed in the edited field.</td>
</tr>
</tbody>
</table>
**Field** | **Explanation**
---|---
Float Symbol | Specify a currency symbol (corresponding to the system value QCURSYM) that will be printed immediately to the left of the left-most digit of an edited field. Valid for a numeric field that has an edit code of 1-4, A-D, or J-M.

---

**Understanding the Report Design Aid Function Keys**

**Function Key** | **Description**
---|---
F5 | Shows the Format Display Control portion of a form.

---

Example F5 - Format Display Control

<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Description</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>028400</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**Field** | **Explanation**
---|---
Sel | Selection. Controls the display of record formats.
Format | Lists the DDS format names. Valid format names are:
- HEADINGn
- HEADINGn+1
- DETAILn
- TOTAL1
Type | Describes the DDS format type. Always REPORT or SFORMS in RDA.
18B Work with Report Design Aid

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boundaries</td>
<td>Two 3-digit numbers that define the range (rows) for the DDS.</td>
</tr>
<tr>
<td></td>
<td>- HEADING1 is usually rows 1 to 8</td>
</tr>
<tr>
<td></td>
<td>- DETAIL1 is row 9 (Only one detail line is defined.)</td>
</tr>
<tr>
<td></td>
<td>- TOTAL1 is rows 10 to 11 (Leave one line for the dashes above the Total field.)</td>
</tr>
<tr>
<td>Window</td>
<td>Allows you to access fields outside the boundaries.</td>
</tr>
<tr>
<td>Browse (Y/N)</td>
<td>Indicator that allows you to enable/disable the browse mode.</td>
</tr>
<tr>
<td>Form Width</td>
<td>Width of the form in print positions.</td>
</tr>
</tbody>
</table>

**Caution:** RDA might automatically adjust displayed formats with those formats that are not displayed.

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F6</td>
<td>Shows the Repository Services portion of a form.</td>
</tr>
</tbody>
</table>

**Example F6 - the Repository Services portion**

<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Description</th>
<th>It</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>985001</td>
<td>Repository-Services</td>
<td></td>
<td>&quot;IC Available Services&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Data Dictionary</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Menus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Vocabulary Overrides</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Function Key Definitions</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Processing Options</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- User Defined Codes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Edit System Help</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- CASE Profiles</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- SAR Log Inquiry</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Copy ED, VO, DM, BDC, GL, Menus</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Sel:=&quot;1&quot;=Select...............F12=Previous----------</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F10</td>
<td>Displays the Record Formats List form.</td>
</tr>
</tbody>
</table>
Example F10 - the Record Formats List form.

```
<table>
<thead>
<tr>
<th>Opt</th>
<th>Format Name</th>
<th>Type</th>
<th>Fast Path File / End Lines</th>
<th>Related Record</th>
<th># Fields</th>
<th>Fld Pfx</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HEADING</td>
<td>REPORT</td>
<td>001 006</td>
<td>000</td>
<td>000</td>
<td>RR</td>
</tr>
<tr>
<td>2</td>
<td>DETAILS</td>
<td>REPORT</td>
<td>002 008</td>
<td>000</td>
<td>000</td>
<td>$$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

The Record Formats establish the arrangement of fields on your report and in what segment of the page they are to print.

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Opt</td>
<td>Enter the appropriate number to indicate you want to select one of the following values:</td>
</tr>
<tr>
<td></td>
<td>File/field pick list of ampersand functions.</td>
</tr>
<tr>
<td></td>
<td>File/field pick list for fast path functions.</td>
</tr>
<tr>
<td></td>
<td>List of defined fields in the format.</td>
</tr>
<tr>
<td></td>
<td>Delete format.</td>
</tr>
<tr>
<td></td>
<td>Report record format keywords.</td>
</tr>
<tr>
<td>Format Name</td>
<td>Report record format.</td>
</tr>
<tr>
<td></td>
<td>The format name will be the Heading, Detail, or Total. If additional formats are required, add 1 to the previous format with the same name.</td>
</tr>
<tr>
<td>Type</td>
<td>Record format type, usually REPORT.</td>
</tr>
<tr>
<td>Fast Path File</td>
<td>The data base file you want to select fields from.</td>
</tr>
<tr>
<td>Start/End Lines</td>
<td>Specifies the line number range of the format.</td>
</tr>
<tr>
<td>Related Record</td>
<td>Field that ties a subfile to a control record format. Required in all SFLCTL record formats.</td>
</tr>
</tbody>
</table>
| Fld Pfx | Screen field prefix to be used for the video fields: RR, $$.

**Note**: There should be no gaps between the end line of one format and the start line of the next format. If you make changes to the positioning of a format and leave a gap between formats, RDA will automatically adjust the end lines for you.
Function Key | Description
--- | ---
F14 | Displays the Indicator Control form.

Example F14 - the Indicator Control form

```
<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9</td>
<td>Inventory by Business Unit</td>
</tr>
<tr>
<td>1 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
<tr>
<td>21 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
<tr>
<td>51 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
<tr>
<td>71 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
<tr>
<td>81 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
<tr>
<td>91 0 0 0 0 0 0 0 0</td>
<td>Description</td>
</tr>
</tbody>
</table>

Ignore all conditioning _ (Y/N)
Reset all indicators ___ ( y/ n )
F3=Exit--F12=Prev Screen-----
```

Use this form to turn on/ off selected or all indicators to see the resulting print image.

Function Key | Description
--- | ---
F16 | Displays the List of Defined Fields form.

Example F16 - the List of Defined Fields form.

```
<table>
<thead>
<tr>
<th>Opt/Field</th>
<th>Description</th>
<th>Row/Col</th>
<th>Type</th>
<th>Size</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>READING1</td>
<td>Record Format</td>
<td>REPORT</td>
<td>001 003</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>*LITBR</td>
<td>Company Name</td>
<td>003 046</td>
<td>A</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>VTX001</td>
<td>Page No.</td>
<td>001 112</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>*PAGE</td>
<td></td>
<td>001 125</td>
<td>A</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>RETTXT1</td>
<td>Inventory by Business Unit</td>
<td>002 046</td>
<td>A</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>VTX002</td>
<td>Date</td>
<td>002 112</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>*DATE</td>
<td></td>
<td>002 125</td>
<td>A</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>RETTXT2</td>
<td>Processing Option Text</td>
<td>003 046</td>
<td>A</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>RETTXT3</td>
<td>Processing Option Text</td>
<td>004 046</td>
<td>A</td>
<td>40</td>
<td>0</td>
</tr>
<tr>
<td>VTX003</td>
<td>Cost</td>
<td>006 002</td>
<td>A</td>
<td>12</td>
<td>0</td>
</tr>
<tr>
<td>VTX006</td>
<td>Item</td>
<td>006 046</td>
<td>A</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>VTX009</td>
<td>Item</td>
<td>006 080</td>
<td>A</td>
<td>8</td>
<td>0</td>
</tr>
<tr>
<td>VTX011</td>
<td>Ship</td>
<td>006 109</td>
<td>A</td>
<td>8</td>
<td>0</td>
</tr>
</tbody>
</table>

Opt: 4=Delete  5=Display/Update  F3=Exit  F12=Prev Screen
```
### Function Key Description

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F17</td>
<td>Used to maintain vocabulary override fields.</td>
</tr>
</tbody>
</table>

#### Example F17

```
<table>
<thead>
<tr>
<th>Business Unit</th>
<th>Dict</th>
<th>Text Description</th>
<th>Screen Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCCCCCCCCC</td>
<td>PAGE E</td>
<td>Page No.</td>
<td>VT0001 0</td>
</tr>
<tr>
<td>ACC</td>
<td>Date</td>
<td></td>
<td>VT0002</td>
</tr>
<tr>
<td>XXX</td>
<td>Business</td>
<td></td>
<td>VT0003 0</td>
</tr>
<tr>
<td>YYY</td>
<td>Unit</td>
<td></td>
<td>VT0004</td>
</tr>
<tr>
<td>ZZZ</td>
<td>Description</td>
<td></td>
<td>VT0005</td>
</tr>
<tr>
<td>AAA</td>
<td>Item</td>
<td></td>
<td>VT0006</td>
</tr>
<tr>
<td>BBB</td>
<td>Type</td>
<td></td>
<td>VT0007</td>
</tr>
<tr>
<td>CCC</td>
<td>Description</td>
<td></td>
<td>VT0008</td>
</tr>
<tr>
<td>DDD</td>
<td>Item</td>
<td></td>
<td>VT0009</td>
</tr>
<tr>
<td>EEE</td>
<td>Number</td>
<td></td>
<td>VT0100</td>
</tr>
<tr>
<td>FFF</td>
<td>Ship</td>
<td></td>
<td>VT0111</td>
</tr>
<tr>
<td>GGG</td>
<td>Description</td>
<td></td>
<td>VT0112</td>
</tr>
</tbody>
</table>
```

You must save your report at least once to update vocabulary overrides by this method. This is because when you are first defining a report, the vocabulary override record is not created until you save the report.

#### Function Key Description

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F19</td>
<td>Window Left.</td>
</tr>
<tr>
<td>F20</td>
<td>Window Right.</td>
</tr>
</tbody>
</table>

### Compiling A Report

#### To compile a report

From Software Versions Repository

```
<table>
<thead>
<tr>
<th>Source</th>
<th>Object</th>
<th>Source</th>
<th>SAR</th>
<th>Version</th>
<th>SID</th>
<th>User</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Library</td>
<td>File</td>
<td>Number</td>
<td>ID</td>
<td>C</td>
<td>P</td>
<td>ID</td>
</tr>
<tr>
<td>JUPER71</td>
<td>JUPER71</td>
<td>JUPER71</td>
<td>22451 L81</td>
<td>1</td>
<td>JYUERLE</td>
<td>10/26/94</td>
<td></td>
</tr>
<tr>
<td>21 YBRO00</td>
<td>YBRO00</td>
<td>YBRO00</td>
<td>24189 A71</td>
<td>A</td>
<td>STUDENTS</td>
<td>07/19/95</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

- Opt: 1=Browse 2=Edit 3=Copy 5=SAR 8=Print 9=Olt 10=Design 14=Oxt 24=More
1. Enter 14 next to the member in the subfile that you want to create and press Enter.

   A form of printer file parameters displays.

   ![Printer File Parameters](image)

2. Do one of the following:
   - Accept the defaults.
   - Change the defaults, as necessary.

   **Note:** Make sure Copies is non-zero.

---

### Changing the Compile Option Defaults for Reports

You must compile reports through the JD Edwards World compiler by this method so that R98COVER and R98RPTH are pulled in for the cover page and help instructions when the Function Code is PRTF. Only one PRTF can be included in an RPG program. Second print files or Special form print files must have a Function Code of PRTS in SVR. Compiling PRTF items through the Production Development Manager (PDM) or some other method will not bring the additional formats in automatically.

### To change compile option defaults for reports

The Data Dictionary default values were set for 8 ½ by 14 printer paper.

Change the Data Dictionary defaults for the following data items for your purposes:

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#FLN</td>
<td>Forms Length</td>
</tr>
<tr>
<td>WDTH</td>
<td>Forms Width</td>
</tr>
<tr>
<td>LPI</td>
<td>Lines Per Inch</td>
</tr>
<tr>
<td>#CPI</td>
<td>Characters Per Inch</td>
</tr>
<tr>
<td>#OVF</td>
<td>Overflow Line Number (Usually forms length minus one inch.)</td>
</tr>
<tr>
<td>Item</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>#ALN</td>
<td>Alignment (Y/ N)</td>
</tr>
<tr>
<td>#FTY</td>
<td>Form Type</td>
</tr>
<tr>
<td>#CPY</td>
<td>Number of Copies</td>
</tr>
<tr>
<td>#SPG</td>
<td>Number of Separator Pages</td>
</tr>
</tbody>
</table>

**Note:** Some severity level 10 errors can occur when your report compiles because of R98COVER (DREAM Writer cover page) and R98RPTH (DREAM Writer help instructions). These are only warning errors.
5 Programming Standards
Overview to Programming Standards

Programming Standards

The Program Generator serves as the primary enforcer of JD Edwards World programming standards. These standards include subroutines and consistent formats that ease the maintenance process. The following areas are covered in the programming standards.

- Program Specifications
- Program Overview
- Program Structure
- Performance Issues
- User Spaces
- User Indices
- File Servers
- Functional Servers
- Group Jobs
- JD Edwards World Source Debugger
Program Specifications

About Program Specifications

There are several kinds of RPG/400 specifications as described in IBM’s Languages: RPG/400 User’s Guide. When your source program is compiled, these specifications are arranged in the following sequence:

- What Are Header (Control) Specifications?
- What Are File Description Specifications?
- What Are Extension Specifications?
- What Are Input Specifications?
- What Are Calculation Specifications?
- What Are Output Specifications?

An RPG/400 program does not have to use all specifications. A typical JD Edwards World program contains Header, File Description, Extension, Input, and Calculation specifications. Few programs have Output specifications.

Note: There are a few programs which have been converted to, or written in, RPGIV (ILE). RPGIV has been available since 1994. RPGIV programs have Data specification entries and no Extension specification entries. Consult IBM reference manuals for details.
What Are Header (Control) Specifications?

The Header (control) specifications (H specs) include the name of the program.

- The first line identifies the program, P55011X, including its description, Item Information Update.
- The next fourteen lines are comments that are included in JD Edwards World programs for copyright purposes and reproduction restrictions.

What Are File Description Specifications?

File description specifications (F Specs) describe all the files that your program uses. The information for each file includes:

- The name of the file
- How the file is used (for example, input)
- The size of records in the file for internal files or if the file has an external definition
- Whether or not the file is keyed
- Input or output device used for the file
- If the file will have records added to it

When the Program Generator generates a program, it arranges the included files in alphabetical order within the F Specs.
When a program runs, it opens the files in bottom-to-top order. Some general rules, in regard to files, are the following:

- Place the files that have the most I/Os at the bottom of the F specs.
- Place any small usage files or files that are closed after first use at the top of the F specs.
- Place the display or print files at the bottom of the list.

Line 35 shows a request for the compiler to copy in F specs from JDECPY. All F spec copy names begin with D.

What Are Extension Specifications?

Extension specifications describe all tables and arrays used in the program. The information includes:

- Name of the file, table, or array
- Number of entries in a table or array input record
- Length of the table or array entry
- Optional comment text

Lines 44 through 47 are used in this program to facilitate error handling and field editing.

- The first line defines an array called EMK which has a maximum of 64 entries, each with a length of 4 characters.

Line 52 requests that the compiler program copy in a specific set of E Specs.

- The E Specs, E0001, are used in any program that executes the common subroutine, C0001.

What Are Input Specifications?

Input specifications describe the records, fields, data structures, and named constants used by the program. The information in the input specifications includes:

- The name of the file
- The sequence of record types
20B Program Specifications

- Whether record-identifying indicators, control-level indicators, field-record relation indicators, or field indicators are used
- Whether data structures, look-ahead fields, record identification codes, or match fields are used
- The type of each file (alphanumeric or numeric; packed-decimal, zoned decimal, or binary format)
- The location of each field in the record
- The name of each field in the record
- All named constants

Lines 73 through 83 are used to define some of the vocabulary overrides that appear on this screen.

- The ending lengths change from program to program, and the program retrieves the values for each field at the time it executes the housekeeping subroutine, S999.
- In JD Edwards World RPGIV (RPGL) programs, there may be D specs and no I or E specs.

What Are Calculation Specifications?

Calculation specifications describe the calculations to be done on the data and the order of the calculations. Calculation specifications can also be used to control certain input and output operations. The information includes:

- Control-level and conditioning indicators for the operation specified (generally not used in JD Edwards World software)
- Fields or constants to be used in the operation
- The operation to be processed
- Whether resulting indicators are set after the operation is processed
The C Specs are the heart of the processing of a program. JD Edwards World programs are designed with a MAINLINE portion which is a select set of C Specs that call other subroutines.

What Are Output Specifications?

Output specifications describe the records and fields in the output files and the conditions under which output operations are processed. They include information such as:

- Name of the file
- Type of record to be written
- Spacing and skipping instructions of printer files
- Output indicators that condition when the record is to be written
- Name of each field in the output record
- Location of each field in the output record
- Edit codes and edit words
- Constants to be written
- Format name for a workstation file

JD Edwards World uses the RPG EXCPT operation to release locks on data records. The O Specs inform the program which record format is to be released when the
EXCPT UNLOCK calculation is performed. Additional formats can be identified with a name such as UNLCKA or UNLCKB.

**Note:** Typically, JD Edwards World does not perform reporting functions using O Specs.

You can use the Opcode “UNLCK” instead of EXCPT/ O-SPECS.
6 Program Overview
Program Overview

About the Program Overview

The program overview provides a basic overview of the standards used in a program. It includes the following:

- Subroutines
- Error Handling
- Indicator Usage
- Documentation
- Miscellaneous Items

Subroutines

The Program Generator uses two categories of subroutines:

- Standard Subroutines
- Common Subroutines

Standard Subroutines

The Program Generator includes the required standard routines in the Calculation Specifications at the time it generates a program. It arranges them in alphanumeric order.

If you must enter your own standard subroutine, name it in such a way that it will be executed in the necessary order. For example, if you need your subroutine to be executed after the scrub and edit subroutine (S005) but before the update files subroutine (S010), begin the name with an S and then use a three to four character suffix that fits in logically, such as S005A or S006.

Standard subroutine code lines are identified in positions 7 and 8 with SR. Their names always begin with an S. Subroutines are separated by a single line of asterisks. Major blocks of code within a subroutine are separated by a single line of dashes.
Place an END tag on the ENDSR statement. The TAG name should start with END. The subroutine name is added as a suffix. For example, END001 would be the used for subroutine S001. Do not use the end tag for anything else. Use a T tag if the code needs to be executed prior to the ENDSR statement. For example, T001 would be used for subroutine S001 if the tag is used in the middle of the subroutine.

Common Subroutines

Common subroutines are maintained outside the program and are included at the appropriate times using the COPY statement. Common subroutines are also referred to as copy modules for that reason. JD Edwards World stores all common subroutines in the file JDECPY.

At compile time, the compiler copies in source code for all instances of the COPY statement. The included code appears only once and can be called from anywhere within the program.

The statement that instructs the compiler to copy in the source code is shown below. Single lines of asterisks separate common subroutines.
This example shows how the COPY statement in the source (above) brings in additional code to the compiled source.

The following user defined code contains an online listing and specifications:

- **Install System Code**: 93
- **User Defined Code**: C

### Error Handling

JD Edwards World has devised an efficient means of handling errors by way of arrays.
The EMK array holds the four byte data dictionary name of every error that could occur in this program. The array is loaded in Housekeeping (S999).

- The @MK array maintains a flag setting for each error identified in EMK. If one of the errors occurs, the flag is set on.
- The @ER array loads the related error messages when the user presses F7 to view the errors that actually occurred.
- A program may have up to 64 errors.

The call to the error message handling program is shown below.

If any error flag is set to one, then the program moves the corresponding data item from the array of all possible errors (EMK) into the array of the errors that have actually occurred (@ER). P0000E is called to display the errors when the function key is pressed.

The next example of code shows how a flag is set in the @MK array.

**Indicator Usage**

There are 99 indicators available for use. They are grouped by purpose. The chart on the next page lists the available indicators and their description.

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Causes the Invalid Function Key Pressed message to appear</td>
</tr>
<tr>
<td>02</td>
<td>Dictates the color palette to be used</td>
</tr>
<tr>
<td>Indicator</td>
<td>Explanation</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>15</td>
<td>Indicates a function key was pressed</td>
</tr>
<tr>
<td>20</td>
<td>Handles the clear screen action code</td>
</tr>
<tr>
<td>21</td>
<td>Handles the add action code</td>
</tr>
<tr>
<td>22</td>
<td>Handles the change action code</td>
</tr>
<tr>
<td>23</td>
<td>Handles the delete action code</td>
</tr>
<tr>
<td>24</td>
<td>Handles the inquire action code</td>
</tr>
<tr>
<td>25</td>
<td>Handles the inquire action code ‘P’ for print (payroll)</td>
</tr>
<tr>
<td>31</td>
<td>Used in conjunction with subfile processing to initiate the INVITE or SFLCLR keyword. Using INVITE will slow processing</td>
</tr>
<tr>
<td>32</td>
<td>Used in conjunction with subfile processing initiating the keyword SFLNXTCHG</td>
</tr>
<tr>
<td>37</td>
<td>Used in conjunction with subfile processing to avoid display of an empty subfile (used only with inquiry subfiles)</td>
</tr>
<tr>
<td>38</td>
<td>Used in conjunction with subfile processing to highlight the last record in the display (keyword SFLDSP) and avoid display of an empty subfile</td>
</tr>
<tr>
<td>40-79</td>
<td>Used for error processing to indicate which fields are in error and need to be highlighted</td>
</tr>
<tr>
<td>40</td>
<td>Reserved for errors in the Action Code field</td>
</tr>
<tr>
<td>41</td>
<td>Reserved for errors in the key fields</td>
</tr>
<tr>
<td>80-89</td>
<td>General reusable one-time indicators. Use them as needed</td>
</tr>
<tr>
<td>93</td>
<td>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 or file error</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>
<tr>
<td>RT</td>
<td>Indicates that a temporary or final halt in the program should take place. Returns to calling program leaving files open</td>
</tr>
</tbody>
</table>

**Documentation**

In the F specifications the program contains several comment lines that are to serve as the program revisions log. The log should list all programmers who have revised
the program, the date the revisions were made, and the SAR outlining the change that was made.

| 0016.00 | F* | PROGRAM REVISION LOG |
| 0017.00 | F* | --------------------- |
| 0018.00 | F* | Date | Programmer | Nature of Revision |
| 0019.00 | F* | -------------------------------------------------- |
| 0020.00 | F* | 01/10/78 | RAPSTIN | SAR # 00000005 | 1A67/006 A/0 |
| 0023.00 | F* | 05/01/93 | RUPPEY | SAR # 00167542 |

When entering comment lines, use the following conventions:

- An asterisk in column seven specifies that the line is a comment line only.
- The asterisk should be followed by four blank spaces before the comment begins.
- Precede and follow the comment lines with a blank line.

The example below shows how these conventions are observed.

| 0034.00 | F* |
| 0035.00 | F* |
| 0036.00 | F* | Copy Member for Composite Common Subroutine - 00001 |
| 0037.00 | F* |

Guidelines

Common sense should be your guide when documenting your programs. Be thorough and descriptive. Put yourself in the place of the next programmer who will inherit your work. Use English and not “programmerese” to specify the action occurring. For example, for the code shown below:

| 0130.00 | C* |
| 0131.00 | C | $998 | CASBO’’ | $998 |
| 0132.00 | C* |
| 0133.00 | C | END |

DON’T WRITE: If $998 is blank, execute S998.

INSTEAD WRITE: Load data field dictionary parameters (one cycle only).

Include a line of dashes beneath any line of code that branches to another line of code (CASxx, CABxx, GOTO, EXSR, CALL, BEGSR). The receiving tag statement should also be followed by a line of dashes as shown in the example below.

| 0275.00 | C* |
| 0276.00 | C | EXSR $998 |
| 0277.00 | C* | **** **** |
Miscellaneous Items

The following represent miscellaneous items that you should keep in mind when writing your own code.

Naming Conventions

Use the following first character to distinguish different item names:

- @ Array names
- $ Program created field names (flags and work fields)
- # Fields defined in common subroutines

Key List (KLIST)

Key lists should all be defined 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, the successive files are noted in the last character space. For example, for three key lists for the physical file F0101, the key lists would be: ABKY01, ABKY02, and ABKY03.
- XXKY0x for logical files where XX is equal to the file prefix and x is equal to the last letter of the logical file name. For example: ABKY0A for F0101LA, ABKY0B for F0101LB.
- When a logical file needs to have more than one key list in a program, the successive files are noted in the second to last character space. For example for three key lists for the logical file F0101LA, the key lists would be: ABKY0A, ABKY1A, and ABKY2A.

Work Fields

Define work fields only once within a program. The use of the *LIKE DEFN command is highly recommended for defining work fields when their attributes are directly tied to those of database fields.

If the work field needs to have the same attributes as a field that exists in a file, for example:

```
MOVE ABANS $$ANS
```

Then define $$ANS:

```
*LIKE DEFN ABANS $$ANS
```

The advantage of this method is that the work field and database field will retain the same attributes even if the database field changes.
When using work fields as a flag, you should assign them the prefix $ and have the remainder of the name be descriptive. In the example below, the work field name is $GLOBL. This name is more descriptive than a field name such as $G.

Optional Files

If a program uses files which are dependent upon your particular setup, you should designate those files as user control open (UC) in the file specifications and then write the program such that they are opened, if needed, in the Housekeeping subroutine. This eliminates the need to open files unnecessarily and conserves resources.

The lines that perform the open are shown below.

If your program performs a user-controlled open for a file that is part of another system, you will also need to provide pre-compiler commands in the event the user has not purchased that system. The example below illustrates the necessary pre-compiler commands designed to address this situation.

In the example, if a Payroll client has not purchased Human Resources, the code specifies a file override and then substitutes an empty file (identified with the suffix E) which all Payroll clients receive.
The user-controlled opens in the program allow the program to run in the absence of certain files, whereas the precompiler commands allow the program to be compiled in the absence of those files.
Program Structure

About Program Structure

The JD Edwards World program generator produces several types of programs:

- Subfile program with selection exits
- Interactive non-subfile program
- Report program without subheadings
- Report program with subheadings
- Maintenance program without a subfile

Internal RPG Subroutines Within JD Edwards World Programs

Standard subfile names make program maintenance easier. The system calls these subfles primarily from the Mainline.

JD Edwards World program structure uses several types of subroutines, including:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| S00EX | Processes all function key exits.  
- Calls J96012 if F24 was pressed  
- Calls X96CCX if F1 was pressed  
- Calls subroutine S00VL if F1 was pressed after X96CCX was called  
- Calls P0000E if F7 was pressed  
- Calls P00HELP if the HELP key was pressed  
- Calls subroutine S001 if F22 was pressed  
- Calls all programs to process all user defined function keys  
| S00VL | Values returned with Cursor Sensitive Help.  
Is called from the subroutine S00EX after the program X96CCX is called  
| S00OP | Subfile Selection Exits (Options)  
| S001 | Clears all database and form fields.  
- Usually only clears key fields and VC0 fields if F22 (Clear) is pressed |
<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
</table>
| S002 | Checks for level breaks for reports.  
  • Turns on level break flags.  
  • Retrieves total line description |
| S003 | Validates the key fields.  
  Calls S998 subroutine if auto inquire was invoked  
  Sets the file pointer.  
  • Performs a SETLL or CHAIN if a single record maintenance program  
  • Performs a SETLL for subfile programs  
  Calls a subroutine S004 to load form or report fields  
  Monitors for no subfile records loaded if a subfile  
  Loads unused subfile records with blanks |
| S004 | Display or load form or report fields. |
| S005 | Scrubs and edits form fields.  
  • Moves screen data to database fields  
  • Turns on error indicators if a field is in error  
  • Updates or writes records to the database file if a subfile  
  • Updates the subfile |
| S010 | For reports with level breaks it:  
  • Prints the total  
  • Clears the level break totals  
  • Prints the grand total (if it has reached the end of the file)  
  • Prints the detail  
  • Adds to the new level break totals  
  • Calls subroutine S020 if it is a report with subheadings  
  If it is not a report, it updates, adds, or deletes records from the database file  
  • Turns on F22 (Clear) to force S001 to be executed 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 |
### Program Structure

<table>
<thead>
<tr>
<th>Name</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  
- Performs file opens  
- Current date retrieval  
- Work fields defined using *LIKE  
- Prints cover page and Helps in a report |

### Interactive Non-Subfile Program

![Interactive Non-Subfile Program Diagram](image)
Subfile Program with Selection Exits

Report Program without Subheadings
Review an RPG Program’s Source

The following pages illustrate a maintenance program without a subfile. Some of the more important areas and commonly used fields are highlighted and explained.
22B Program Structure

Data Structure to Load Video Screen Text

Data structure for commonly used indexes

Program status data structure

Each VTX field is 40 bytes long but may not use all 40.

Dialog in text from Vocabulary Overides.

One time only function

If information is passed to this program, it will automatically inquiry on the screen.
Sets the message for line 24

Values assigned in the Function Key Definitions program

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

Parameters passed identifying where the cursor was when F1 was pressed
22B Program Structure

```
206.00 C* If Display errors press, exit to error messages.
                --------------------------------------------
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*                
228.00 C*                
229.00 C*                
230.00 C*                
231.00 C*                
232.00 C*                
233.00 C*                
234.00 C*                
235.00 C*                
236.00 C*                
237.00 C*                
238.00 C*                
239.00 C*                
240.00 C*                
241.00 C*                
242.00 C*                
243.00 C*                
244.00 C*                
245.00 C*                
246.00 C*                
247.00 C*                
248.00 C*                
249.00 C*                
250.00 C*                
251.00 C*                
252.00 C*                
253.00 C*                
254.00 C*                
255.00 C*                
256.00 C*                
257.00 C*                
258.00 C*                
259.00 C*                
260.00 C*                
261.00 C*                
262.00 C*                
263.00 C*                
264.00 C*                
265.00 C*                
266.00 C*                
267.00 C*                
268.00 C*                
269.00 C*                
270.00 C*                
271.00 C*                
272.00 C*                
273.00 C*                
274.00 C*                
275.00 C*                
276.00 C*                
277.00 C*                
278.00 C*                
279.00 C*                
280.00 C*                
281.00 C*                
282.00 C*                
283.00 C*                
284.00 C*                
285.00 C*                
286.00 C*                
287.00 C*                
288.00 C*                
289.00 C*                
290.00 C*                
291.00 C*                
292.00 C*                
293.00 C*                
294.00 C*                
```

Access JDE program level
Help information

JD Edwards World, A9.1
Program Structure

C******************************************************************************
466.00 C* Copy Common Subroutine - Cost Center Security Check
467.00 C******************************************************************************
468.00 C* Returned value. If the format is a suffix, the record
469.00 C* to change is found in A$R$.
470.00 C******************************************************************************
471.00 C* SUBROUTINE EDGVL - Cursor Control Return Values
472.00 C******************************************************************************
492.00 C* For cursor sensitive help. Information was retrieved in program X96CCX.
493.00 C* The retrieved information is returned to the
494.00 C* video fields in this subroutine.

<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction 1</th>
<th>Instruction 2</th>
<th>Action 1</th>
<th>Action 2</th>
<th>Action 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>512.00</td>
<td>CER</td>
<td>EICD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>514.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>515.00</td>
<td>CER</td>
<td>MOVELLYVAL</td>
<td>V0302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>516.00</td>
<td>CER</td>
<td>GOTO</td>
<td>EICDVL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>518.00</td>
<td>C’</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>520.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>525.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>526.00</td>
<td>CER</td>
<td>MOVELLYVAL</td>
<td>V0302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>527.00</td>
<td>CER</td>
<td>GOTO</td>
<td>EICDVL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>528.00</td>
<td>C’</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>530.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>532.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>534.00</td>
<td>CER</td>
<td>MOVELLYVAL</td>
<td>V0302</td>
<td></td>
<td></td>
</tr>
<tr>
<td>535.00</td>
<td>CER</td>
<td>GOTO</td>
<td>EICDVL</td>
<td></td>
<td></td>
</tr>
<tr>
<td>536.00</td>
<td>C’</td>
<td>----</td>
<td>----</td>
<td></td>
<td></td>
</tr>
<tr>
<td>538.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>540.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>542.00</td>
<td>C’</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

22BProgram Structure

---

Clears all the fields in the record format for 92001

---

Clears the video fields

---

These fields will only be cleared if the user presses the function key to clear the screen. We want to save certain information like key fields and descriptions so they don’t get deleted every time 9001 is executed.
Program Structure

22B

---

SUBROUTINE SUB03 - Edit Key
---

Sets the file pointer and edits the key.

---

Processing:
1. Clear error indicators and arrays.
2. Load input keys.
3. Validate master file key.
4. Release master file record lock.
5. Load video screen output on inquiry.

---

Load data field dictionary parameters (one cycle only).

---

Reset error indicators and arrays.

---

Load video input field for - Item ID

---

Automatic Next Number for - Item ID

---

Automatic Next Number for - Item ID

---

Automatic Next Number for - Item ID

---

Automatic Next Number for - Item ID

---

Checks cost center security.

---

If security violation, set error condition.
Program Structure

22B

MOVE ' ' schön 1
GOTO END003

----- ------
END

Move result of read and action code.

If indicator 41 on, invalid key for action code.

If indicator 99 on, record in use.

If indicator 49 on, release record lock.

If errors, skip remainder of subroutine.

Move data base information to video screen.

 Moves information to the video/report fields

Copy common subroutine - right justify numeric fields

Copy JNDBK, JNDBK, etc.

Copy common subroutine - load video screen data

Processing: Move data base information to video screen.

All video screen fields are alpha and therefore numeric information must be processed through subroutine save4 to set proper variables and provide editing for display on screen.

Data fields must be converted from their internal format of month, day and year or cut-out to the system format using program
22B Program Structure

```
055.00  C1T  MOVE@40000  VD001
056.0  C1T  END
057.0  C1T  *-----------------------------------------------*
058.0  C1T  * Description display for - Item Category Code 003*
059.0  C1T  *-----------------------------------------------*
061.0  C1T  C1T  CLDATA10000
062.0  C1T  C1T  MOVE@40000  VD001
063.0  C1T  C1T  MOVE@40004  VD001
064.0  C1T  C1T  MOVE@40005  VD001
065.0  C1T  C1T  CALL@'00000'  V11
066.0  C1T  * ---- ------
067.0  C1T  C1T  PARM  @00000
068.0  C1T  C1T  move 'blank'  VD0007
069.0  C1T  C1T  C1T  IPFQ 'O'
070.0  C1T  C1T  MOVE@40001  VD0006
071.0  C1T  C1T  END
072.0  C1T  *-----------------------------------------------*
073.0  C1T  * Description display for - Item Category Code 004*
075.0  C1T  *-----------------------------------------------*
076.0  C1T  C1T  CLDATA10000
077.0  C1T  C1T  MOVE@40000  VD001
078.0  C1T  C1T  MOVE@40004  VD001
079.0  C1T  C1T  MOVE@40005  VD001
080.0  C1T  C1T  CALL@'00000'  V11
081.0  C1T  * ---- ------
082.0  C1T  C1T  PARM  @00000
083.0  C1T  C1T  move 'blank'  VD0007
084.0  C1T  C1T  C1T  IPFQ 'O'
085.0  C1T  C1T  MOVE@40001  VD0007
086.0  C1T  C1T  END
087.0  C1T  *-----------------------------------------------*
089.0  C1T  * Description display for - Item Category Code 005*
090.0  C1T  *-----------------------------------------------*
091.0  C1T  C1T  CLDATA10000
092.0  C1T  C1T  MOVE@40000  VD001
093.0  C1T  C1T  MOVE@40004  VD001
094.0  C1T  C1T  MOVE@40005  VD001
095.0  C1T  C1T  CALL@'00000'  V11
096.0  C1T  * ---- ------
097.0  C1T  C1T  PARM  @00000
098.0  C1T  C1T  move 'blank'  VD0008
099.0  C1T  C1T  C1T  IPFQ 'O'
100.0  C1T  C1T  MOVE@40001  VD0008
101.0  C1T  C1T  END
102.0  C1T  *-----------------------------------------------*
103.0  C1T  Move to output - Cost Center
104.0  C1T  *-----------------------------------------------*
105.0  C1T  C1T  NOTE 'blank'  @00000
106.0  C1T  C1T  MOVE@40000  @00000
107.0  C1T  C1T  MOVE@40004  @00000
108.0  C1T  C1T  MOVE@40005  @00000
109.0  C1T  C1T  CALL@'00000'  V11
110.0  C1T  * ---- ------
111.0  C1T  C1T  PARM  @00000
112.0  C1T  C1T  move 'blank'  VD0008
113.0  C1T  C1T  C1T  IPFQ 'O'
114.0  C1T  C1T  MOVE@40001  VD0008
115.0  C1T  *-----------------------------------------------*
116.0  C1T  Move to output - Description
117.0  C1T  *-----------------------------------------------*
118.0  C1T  C1T  NOTE 'blank'  +00000
119.0  C1T  C1T  MOVE@40000  +00000
120.0  C1T  C1T  MOVE@40004  +00000
121.0  C1T  C1T  MOVE@40005  +00000
122.0  C1T  C1T  CALL@'00000'  V11
123.0  C1T  * ---- ------
124.0  C1T  C1T  PARM  +00000
125.0  C1T  C1T  move 'blank'  VD0008
126.0  C1T  C1T  C1T  IPFQ 'O'
127.0  C1T  C1T  MOVE@40001  VD0008
128.0  C1T  C1T  END
129.0  C1T  *-----------------------------------------------*
130.0  C1T  Move to Output - Data Last Ship
131.0  C1T  *-----------------------------------------------*
```

**Program Structure**

```assembly
000.00 CER MOVE QEDIT $EDIT 6
022.00 CER MOVE *BLANK $EDIT 2
024.00 CER MOVE "$BLANK" $EDIT 7
035.00 CER MOVE "BLANK" $EDIT 7
040.00 CER MOVE "$BLANK" $EDIT 7
042.00 CER MOVE "$BLANK" $EDIT 7
044.00 CER MOVE "$BLANK" $EDIT 7
046.00 CER MOVE "$BLANK" $EDIT 7
048.00 CER MOVE "$BLANK" $EDIT 7
049.00 CER CALL $EDIT 12
050.00 CER CALL $EDIT 12
051.00 CER MOVE *BLANK $EDIT 7
052.00 CER MOVE *BLANK $EDIT 7
053.00 CER MOVE "BLANK" $EDIT 7
054.00 CER MOVE "BLANK" $EDIT 7
055.00 CER MOVE "BLANK" $EDIT 7
056.00 CER MOVE "BLANK" $EDIT 7
057.00 CER MOVE "BLANK" $EDIT 7
058.00 CER MOVE "BLANK" $EDIT 7
059.00 CER MOVE "BLANK" $EDIT 7
060.00 CER MOVE "BLANK" $EDIT 7
061.00 CER MOVLE *BLANK $EDIT 7
062.00 CER MOVLE "BLANK" $EDIT 7
063.00 CER MOVLE "$BLANK" $EDIT 7
064.00 CER MOVLE "$BLANK" $EDIT 7
065.00 CER MOVLE "$BLANK" $EDIT 7
066.00 CER MOVLE "$BLANK" $EDIT 7
067.00 CER MOVLE "$BLANK" $EDIT 7
068.00 CER MOVLE "$BLANK" $EDIT 7
069.00 CER MOVLE "$BLANK" $EDIT 7
070.00 CER MOVLE "$BLANK" $EDIT 7
071.00 CER MOVLE "$BLANK" $EDIT 7
072.00 CER MOVLE "$BLANK" $EDIT 7
073.00 CER MOVLE "$BLANK" $EDIT 7
074.00 CER MOVLE "$BLANK" $EDIT 7
075.00 CER MOVLE "$BLANK" $EDIT 7
076.00 CER MOVLE "$BLANK" $EDIT 7
077.00 CER MOVLE "$BLANK" $EDIT 7
078.00 CER MOVLE "$BLANK" $EDIT 7
079.00 CER MOVLE "$BLANK" $EDIT 7
080.00 CER MOVLE "$BLANK" $EDIT 7
081.00 CER MOVLE "$BLANK" $EDIT 7
082.00 CER MOVLE "$BLANK" $EDIT 7
083.00 CER MOVLE "$BLANK" $EDIT 7
084.00 CER MOVLE "$BLANK" $EDIT 7
085.00 CER MOVLE "$BLANK" $EDIT 7
086.00 CER MOVLE "$BLANK" $EDIT 7
087.00 CER MOVLE "$BLANK" $EDIT 7
088.00 CER MOVLE "$BLANK" $EDIT 7
089.00 CER MOVLE "$BLANK" $EDIT 7
090.00 CER MOVLE "$BLANK" $EDIT 7
091.00 CER MOVLE "$BLANK" $EDIT 7
092.00 CER MOVLE "$BLANK" $EDIT 7
093.00 CER MOVLE "$BLANK" $EDIT 7
094.00 CER MOVLE "$BLANK" $EDIT 7
095.00 CER MOVLE "$BLANK" $EDIT 7
096.00 CER MOVLE "$BLANK" $EDIT 7
097.00 CER MOVLE "$BLANK" $EDIT 7
098.00 CER MOVLE "$BLANK" $EDIT 7
099.00 CER MOVLE "$BLANK" $EDIT 7
1000.00 CER MOVE *BLANK $EDIT 7
1001.00 CER MOVE *BLANK $EDIT 7
1002.00 CER MOVE *BLANK $EDIT 7
1003.00 CER MOVE *BLANK $EDIT 7
1004.00 CER MOVE *BLANK $EDIT 7
1005.00 CER MOVE *BLANK $EDIT 7
1006.00 CER MOVE *BLANK $EDIT 7
1007.00 CER MOVE *BLANK $EDIT 7
1008.00 CER MOVE *BLANK $EDIT 7
1009.00 CER MOVE *BLANK $EDIT 7
```

---

*External program used to edit dates*

---

**22BProgram Structure**
22BProgram Structure

1097.00 CES MOVE *ELANK REDSER
1098.00 CES MOVE *CMID09 REDSER
1099.00 CES MOVE T40000 RDOT7
1100.00 CES MOVE *CMID07 REDSER
1101.00 CES MOVE B40000 RED
1102.00 CES MOVE T40000 RDOT7
1103.00 CES MOVE G40000 RDOT7
1104.00 CES MOVE *CMID07 REDSER
1105.00 CES MOVE ' ' REDSER
1106.00 CES MOVE ' ' REDSER
1107.00 CES EZRE C30161
1109.00 CES ALAX IPFU 'L'
1110.00 CES RETL,DIHR V50005
1111.00 CES EZLE
1112.00 CES MOVE REDSER V50005
1113.00 CES ENDF
1114.00 CES C*-----------------------------------------------------------------------
1115.00 CES C* Copy Common Subroutine - Format Numeric Fields for Output with Overide
1116.00 CES C*/---------------------------------------------------------------------
1117.00 CES C* SUBROUTINE 3055 - Scrub input
1118.00 CES C* Validates and edits data
1119.00 CES C* entered by the user
1120.00 CES C*-----------------------------------------------------------------------
1121.00 CES C* Processing: 1. Validate all video input.
1122.00 CES C* All numeric fields must be processed
1123.00 CES C* thru subroutine 30002 and 30012 in order
1124.00 CES C* to scrub the alpha input field and convert
1125.00 CES C* to degree and minutes.
1126.00 CES C* Data fields must be converted from system
1127.00 CES C* format to their internal format of month,
1128.00 CES C* day and year or julian using program X0020.
1129.00 CES C* a. System date records fields from video.
1130.00 CES C* Only performs this subroutine if action code is
1131.00 CES C* added or change.
1132.00 CES C*-----------------------------------------------------------------------
1133.00 CES C* SUBROUTINE 3055 - Scrub input
1134.00 CES C* Validates and edits data
1135.00 CES C* entered by the user
1136.00 CES C*-----------------------------------------------------------------------
1137.00 CES C* Processing: 1. Validate all video input.
1138.00 CES C* All numeric fields must be processed
1139.00 CES C* thru subroutine 30002 and 30012 in order
1140.00 CES C* to scrub the alpha input field and convert
1141.00 CES C* to degree and minutes.
1142.00 CES C* Data fields must be converted from system
1143.00 CES C* format to their internal format of month,
1144.00 CES C* day and year or julian using program X0020.
1145.00 CES C* a. System date records fields from video.
1146.00 CES C* Only performs this subroutine if action code is
1147.00 CES C* added or change.
1148.00 CES C*-----------------------------------------------------------------------
22BProgram Structure

Common subroutine to convert screen fields to numeric data

Work fields used in the RPG program begin with $ $

Work fields used in a copy module begin with &

Convert to numeric

Adjust for display decimals
<table>
<thead>
<tr>
<th>Line</th>
<th>Instruction</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1242.00</td>
<td>C*</td>
<td>Program Structure</td>
</tr>
<tr>
<td>1243.00</td>
<td>GSR MOVE PMeIT WDGUI</td>
<td></td>
</tr>
<tr>
<td>1244.00</td>
<td>GSR MOVE GMEIT WDGUI</td>
<td></td>
</tr>
<tr>
<td>1245.00</td>
<td>GSR REEB CIVISS</td>
<td></td>
</tr>
<tr>
<td>1246.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1247.00</td>
<td>GSR MOVE #8963 QREXIT</td>
<td></td>
</tr>
<tr>
<td>1248.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1249.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1250.00</td>
<td>C*</td>
<td>*Edit upper and lower range - Item ID</td>
</tr>
<tr>
<td>1251.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1252.00</td>
<td>GSR LAEXIT IFPE *BLANK</td>
<td></td>
</tr>
<tr>
<td>1253.00</td>
<td>GSR MOVE *BLANK QEXIT 15</td>
<td></td>
</tr>
<tr>
<td>1254.00</td>
<td>GSR MOVE '1' QEXIT</td>
<td></td>
</tr>
<tr>
<td>1255.00</td>
<td>GSR MOVE.CQEXIT QEXIT</td>
<td></td>
</tr>
<tr>
<td>1256.00</td>
<td>GSR XEXIT IFPE LAEXIT</td>
<td></td>
</tr>
<tr>
<td>1257.00</td>
<td>GSR XEXIT ANCEMEXIT</td>
<td></td>
</tr>
<tr>
<td>1258.00</td>
<td>GSR MOVE '1' TYPESET</td>
<td></td>
</tr>
<tr>
<td>1259.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1261.00</td>
<td>GSR MOVE '1' 165,07</td>
<td></td>
</tr>
<tr>
<td>1262.00</td>
<td>GSR ARROW 4190</td>
<td></td>
</tr>
<tr>
<td>1263.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1264.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1265.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1266.00</td>
<td>C*</td>
<td>*Scrub and edit - Quantity - On Hand</td>
</tr>
<tr>
<td>1268.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1269.00</td>
<td>GSR MOVEAVEXT 35M</td>
<td></td>
</tr>
<tr>
<td>1271.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1272.00</td>
<td>GSR REEB CIVISS</td>
<td></td>
</tr>
<tr>
<td>1273.00</td>
<td>GSR MOVE PMeIT WDGUI</td>
<td></td>
</tr>
<tr>
<td>1274.00</td>
<td>GSR REEB CIVISS</td>
<td></td>
</tr>
<tr>
<td>1275.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1276.00</td>
<td>GSR MOVE #8963 QREXIT</td>
<td></td>
</tr>
<tr>
<td>1277.00</td>
<td>C*</td>
<td>*Set default value - Quantity - On hand</td>
</tr>
<tr>
<td>1278.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1279.00</td>
<td>GSR VISIT IFPE *BLANK</td>
<td></td>
</tr>
<tr>
<td>1280.00</td>
<td>GSR DWQT ANCEM-BLANK</td>
<td></td>
</tr>
<tr>
<td>1282.00</td>
<td>GSR MOVEAVEXT 35M</td>
<td></td>
</tr>
<tr>
<td>1283.00</td>
<td>GSR REEB CIVISS</td>
<td></td>
</tr>
<tr>
<td>1284.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1285.00</td>
<td>GSR MOVE PMeIT WDGUI</td>
<td></td>
</tr>
<tr>
<td>1286.00</td>
<td>GSR MOVE '1' QEXIT</td>
<td></td>
</tr>
<tr>
<td>1287.00</td>
<td>GSR MOVE.CQEXIT QEXIT</td>
<td></td>
</tr>
<tr>
<td>1288.00</td>
<td>GSR XEXIT IFPE LAEXIT</td>
<td></td>
</tr>
<tr>
<td>1289.00</td>
<td>GSR XEXIT ANCEMEXIT</td>
<td></td>
</tr>
<tr>
<td>1290.00</td>
<td>GSR MOVE '1' TYPESET</td>
<td></td>
</tr>
<tr>
<td>1291.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1292.00</td>
<td>C*</td>
<td>*Edit upper and lower range - Quantity - On hand</td>
</tr>
<tr>
<td>1293.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1294.00</td>
<td>GSR LAEXIT TYPE *BLANK</td>
<td></td>
</tr>
<tr>
<td>1295.00</td>
<td>GSR MOVE *BLANK QEXIT 15</td>
<td></td>
</tr>
<tr>
<td>1296.00</td>
<td>GSR MOVE '1' QEXIT</td>
<td></td>
</tr>
<tr>
<td>1297.00</td>
<td>GSR MOVE.CQEXIT QEXIT</td>
<td></td>
</tr>
<tr>
<td>1298.00</td>
<td>GSR XEXIT TYPE LAEXIT</td>
<td></td>
</tr>
<tr>
<td>1299.00</td>
<td>GSR XEXIT ANCEMEXIT</td>
<td></td>
</tr>
<tr>
<td>1300.00</td>
<td>GSR MOVE '1' TYPESET</td>
<td></td>
</tr>
<tr>
<td>1301.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1302.00</td>
<td>GSR TYPESET IFPE '1'</td>
<td></td>
</tr>
<tr>
<td>1303.00</td>
<td>GSR MOVE '1' 165,07</td>
<td>9500</td>
</tr>
<tr>
<td>1304.00</td>
<td>GSR ARROW 4600</td>
<td></td>
</tr>
<tr>
<td>1305.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1306.00</td>
<td>GSR END</td>
<td></td>
</tr>
<tr>
<td>1307.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1308.00</td>
<td>C*</td>
<td>*Scrub and edit - Item Type</td>
</tr>
<tr>
<td>1309.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1310.00</td>
<td>GSR MOVETYPE QREXIT</td>
<td></td>
</tr>
<tr>
<td>1311.00</td>
<td>GSR</td>
<td></td>
</tr>
<tr>
<td>1312.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1313.00</td>
<td>C*</td>
<td>*Set default value - Item Type</td>
</tr>
<tr>
<td>1314.00</td>
<td>C*</td>
<td></td>
</tr>
<tr>
<td>1315.00</td>
<td>GSR QREXIT IFPE *BLANK</td>
<td></td>
</tr>
<tr>
<td>1316.00</td>
<td>GSR QREXIT IFPE *BLANK</td>
<td></td>
</tr>
<tr>
<td>1317.00</td>
<td>GSR MOVETYPE 940</td>
<td></td>
</tr>
<tr>
<td>1318.00</td>
<td>GSR MOVETYPE QREXIT</td>
<td></td>
</tr>
</tbody>
</table>
22B Program Structure

```
1550.00 CLR #UERR 1F00 1E0000
1551.00 CLR MOVE 'Y' @MK,09
1552.00 CLR MOVE 'Y' @MK,09
1553.00 CLR GETM 4933
1554.00 CLR END
1555.00 CLR END
1556.00 C*------------------------------------------
1557.00 C* Srch and edit - Item Category Code 002
1558.00 C*------------------------------------------
1559.00 CLR MOVE/COORD 002
1560.00 CLR MOVE/COORD 002
1561.00 C*------------------------------------------
1562.00 C* Set default value - Item Category Code 002
1563.00 C*------------------------------------------
1564.00 CLR @X002 1FP# *BLANK
1565.00 CLR @X002 1FP# *BLANK
1566.00 CLR MOVEAB@X002 @X002
1567.00 CLR MOVEAB@X002 @X002
1568.00 CLR MOVE 43A 3FP# 3FP#'
1569.00 CLR MOVE 'Y' @MK,09
1570.00 CLR ADD 1 @MK
1571.00 CLR END
1572.00 CLR MOVEAB@X002 @X002
1573.00 CLR ADD 1 @MK
1574.00 CLR END
1575.00 CLR ADD 1 @MK
1576.00 CLR END
1577.00 CLR MOVEAB@X002 @X002
1578.00 CLR END
1579.00 CLR END
1580.00 CLR END
1581.00 C*------------------------------------------
1582.00 C* Edit allowed values - Item Category Code 002
1583.00 C*------------------------------------------
1584.00 CLR AER@X002 1FP$ *BLANK
1585.00 CLR AER@X002 1FP$ *BLANK
1586.00 CLR AER@X002 1FP$ *BLANK
1587.00 CLR MOVE 'Y' @MK,09
1588.00 CLR GETM 4933
1589.00 CLR ADD 1 @MK
1590.00 CLR MOVEAB@X002 @X002
1591.00 CLR MOVEAB@X002 @X002
1592.00 CLR MOVE 43A 3FP$ 3FP$'
1593.00 CLR MOVE 'Y' @MK,09
1594.00 CLR MOVE 'Y' @MK,09
1595.00 CLR MOVE 'Y' @MK,09
1596.00 CLR MOVEAB@X002 @X002
1597.00 CLR MOVEAB@X002 @X002
1598.00 CLR MOVEAB@X002 @X002
1599.00 CLR MOVEAB@X002 @X002
1600.00 CLR MOVE 43A 3FP$ 3FP$'
1601.00 CLR MOVE 'Y' @MK,09
1602.00 CLR MOVE 'Y' @MK,09
1603.00 CLR MOVE 'Y' @MK,09
1604.00 CLR GETM 4933
1605.00 CLR END
1606.00 CLR END
1607.00 CLR END
1608.00 CLR END
1609.00 C*------------------------------------------
1610.00 C* Edit upper and lower range - Item Category Code 002
1611.00 C*------------------------------------------
1612.00 CLR AER@X002 1FP$ *BLANK
1613.00 CLR AER@X002 1FP$ *BLANK
1614.00 CLR AER@X002 1FP$ *BLANK
1615.00 CLR AER@X002 1FP$ *BLANK
1616.00 CLR AER@X002 1FP$ *BLANK
1617.00 CLR AER@X002 1FP$ *BLANK
1618.00 CLR AER@X002 1FP$ *BLANK
1619.00 CLR AER@X002 1FP$ *BLANK
1620.00 CLR AER@X002 1FP$ *BLANK
1621.00 CLR AER@X002 1FP$ *BLANK
1622.00 CLR AER@X002 1FP$ *BLANK
1623.00 CLR AER@X002 1FP$ *BLANK
1624.00 CLR AER@X002 1FP$ *BLANK
1625.00 CLR AER@X002 1FP$ *BLANK
1626.00 CLR AER@X002 1FP$ *BLANK
1627.00 CLR AER@X002 1FP$ *BLANK
1628.00 CLR AER@X002 1FP$ *BLANK
```

6-36  JD Edwards World, A9.1
22B Program Structure

```
1627.00  CAR  CLEAR100500
1629.00  CAR  MOVE@@1002  #DIS
1629.00  CAR  MOVE @1002  #DIS
1632.00  CAR  MOVU QA1002  MOVU
1633.00  CAR  CALL 'X300'  01
1634.00  CAR  DRZ  --------
1635.00  CAR  CAR  KARU  D00100
1636.00  CAR  @TPRX  IFPE '1'
1638.00  CAR  MOVU '1'  4992
1639.00  CAR  SETON  END
1640.00  CAR  END
1640.00  CAR  C'  Scrub and edit - Item Category Code 000
1642.00  CAR  C'  
1643.00  CAR  MOVU9003  QA0003
1644.00  CAR  C'  Set default value - Item Category Code 000
1645.00  CAR  C'  
1646.00  CAR  C'  
1647.00  CAR  QA0003  IFPE 'BLANK'
1648.00  CAR  QA0003  IFPE 'BLANK'
1649.00  CAR  MOVU@QA0003  #400
1650.00  CAR  MOVU@QA0003  QA0003
1651.00  CAR  #40,1  IFPE
1652.00  CAR  MOVU ' '  #40,1
1653.00  CAR  MOVU ' '  #40,1
1654.00  CAR  MOVU ' '  #40,1
1655.00  CAR  MOVU ' '  #40,1
1656.00  CAR  MOVU ' '  #40,1
1657.00  CAR  MOVU ' '  #40,1
1658.00  CAR  MOVU ' '  #40,1
1659.00  CAR  MOVU ' '  #40,1
1660.00  CAR  MOVU@QA0003  QA0003
1661.00  CAR  MOVU@QA0003  QA0003
1662.00  CAR  MOVU@QA0003  QA0003
1663.00  CAR  MOVU@QA0003  QA0003
1664.00  CAR  C'  
1665.00  CAR  C'  Edit allowed values - Item Category Code 000
1666.00  CAR  C'  
1667.00  CAR  QA0003  IFPE 'BLANK'
1668.00  CAR  QA0003  IFPE 'BLANK'
1669.00  CAR  QA0003  IFPE 'BLANK'
1670.00  CAR  MOVU ' '  #40,1
1671.00  CAR  SETON  5092
1672.00  CAR  SETON  5092
1673.00  CAR  SETON  5092
1674.00  CAR  SETON  5092
1675.00  CAR  SETON  5092
1676.00  CAR  MOVU ' '  SETINT
1677.00  CAR  MOVU ' '  SETINT
1678.00  CAR  MOVU ' '  SETINT
1679.00  CAR  MOVU@QA0003  QA0003
1680.00  CAR  MOVU@QA0003  QA0003
1681.00  CAR  MOVU@QA0003  QA0003
1682.00  CAR  MOVU@QA0003  QA0003
1683.00  CAR  MOVU@QA0003  QA0003
1684.00  CAR  MOVU@QA0003  QA0003
1685.00  CAR  MOVU@QA0003  QA0003
1686.00  CAR  MOVU@QA0003  QA0003
1687.00  CAR  MOVU@QA0003  QA0003
1688.00  CAR  MOVU@QA0003  QA0003
1689.00  CAR  MOVU@QA0003  QA0003
1690.00  CAR  MOVU@QA0003  QA0003
1691.00  CAR  MOVU@QA0003  QA0003
1692.00  CAR  MOVU@QA0003  QA0003
1693.00  CAR  MOVU@QA0003  QA0003
1694.00  CAR  MOVU@QA0003  QA0003
1695.00  CAR  MOVU@QA0003  QA0003
1696.00  CAR  MOVU@QA0003  QA0003
1697.00  CAR  MOVU@QA0003  QA0003
1698.00  CAR  MOVU@QA0003  QA0003
1699.00  CAR  MOVU@QA0003  QA0003
1700.00  CAR  MOVU@QA0003  QA0003
1701.00  CAR  MOVU@QA0003  QA0003
1702.00  CAR  MOVU@QA0003  QA0003
1703.00  CAR  MOVU@QA0003  QA0003
```

6-37
22B Program Structure

```
1650.00 C*  Edit upper and lower range - Item Category Code 005
1659.00 C*  
1660.00 C*  
1661.00 CSE  &XW005  IPHE 'BLANK'  SBSETX
1662.00 CSE  &XW005  IPHE '0'  SBSETX
1664.00 CSE  &XW005  ANKLOBD005
1665.00 CSE  MOVE '0'  SBSETX
1666.00 CSE  END
1667.00 CSE  SBSETX  IPHE '1'
1669.00 CSE  MOVE '1'  $MEM, 07
1670.00 CSE  GOTO  6290
1671.00 CSE  END
1672.00 C*  
1673.00 C*  Edit from User Defined Codes - Item Category Code 005
1674.00 C*  
1675.00 CSE  &XW005  IPHE 'BLANK'
1676.00 CSE  &XW005  IPHE '0'  $MEM
1677.00 CSE  MOVE &XW005  $MEM
1678.00 CSE  MOVE &XW005  $MEM
1679.00 CSE  CALL 'XW005',  01
1681.00 C*  
1682.00 CSE  $MEM  100000
1683.00 CSE  $MEM, IPHE '1'
1684.00 CSE  MOVE '1'  $MEM, 09
1685.00 CSE  GOTO  5290
1686.00 CSE  END
1687.00 CSE  END

```

JD Edwards World, A9.1
22B Program Structure

Program Structure

1290.00 C* Clear data field for next transaction
1290.00 C* SIMulates use pressing the Clear Screen function key.
1300.00 C* Forces clear of everything before processing next record.
1300.00 C* Retrieves all of the Data Dictionary parameters for necessary data items used in the program and moves the information into constant fields
1310.00 C* Data Dictionary file server
1320.00 C* Dictionary parameters for - Cost Center
1330.00 C* Dictionary parameters for - Description
1340.00 C* Data Dictionary file server
1350.00 C* Dictionary parameters for - Cost Center
1360.00 C* Dictionary parameters for - Description
1370.00 C* Data Dictionary file server
1380.00 C* Dictionary parameters for - Cost Center
1390.00 C* Dictionary parameters for - Description
1400.00 C* Data Dictionary file server
1410.00 C* Dictionary parameters for - Cost Center
1420.00 C* Dictionary parameters for - Description
1430.00 C* Data Dictionary file server
1440.00 C* Dictionary parameters for - Cost Center
1450.00 C* Dictionary parameters for - Description
1460.00 C* Data Dictionary file server
1470.00 C* Dictionary parameters for - Cost Center
1480.00 C* Dictionary parameters for - Description
1490.00 C* Data Dictionary file server
1500.00 C* Dictionary parameters for - Cost Center
1510.00 C* Dictionary parameters for - Description
1520.00 C* Data Dictionary file server
1530.00 C* Dictionary parameters for - Cost Center
1540.00 C* Dictionary parameters for - Description
1550.00 C* Data Dictionary file server
1560.00 C* Dictionary parameters for - Cost Center
1570.00 C* Dictionary parameters for - Description
1580.00 C* Data Dictionary file server
1590.00 C* Dictionary parameters for - Cost Center
1600.00 C* Dictionary parameters for - Description
1610.00 C* Data Dictionary file server
1620.00 C* Dictionary parameters for - Cost Center
1630.00 C* Dictionary parameters for - Description
1640.00 C* Data Dictionary file server
1650.00 C* Dictionary parameters for - Cost Center
1660.00 C* Dictionary parameters for - Description
1670.00 C* Data Dictionary file server
1680.00 C* Dictionary parameters for - Cost Center
1690.00 C* Dictionary parameters for - Description
1700.00 C* Data Dictionary file server
1710.00 C* END
1720.00 C* END
1730.00 C* END
1740.00 C* END
1750.00 C* END
1760.00 C* END
1770.00 C* END
1780.00 C* END
1790.00 C* END
1800.00 C* END
1810.00 C* END
1820.00 C* END
1830.00 C* END
1840.00 C* END
1850.00 C* END
1860.00 C* END
1870.00 C* END
1880.00 C* END
1890.00 C* END
1900.00 C* END
1910.00 C* END
1920.00 C* END
1930.00 C* END
1940.00 C* END
1950.00 C* END
1960.00 C* END
1970.00 C* END
1980.00 C* END
1990.00 C* END
2000.00 C* END
2010.00 C* END
2020.00 C* END
2030.00 C* END
2040.00 C* END
2050.00 C* END
```
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>2237</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN003 4</td>
</tr>
<tr>
<td>2239</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2239</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2240</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2240</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2241</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 4</td>
</tr>
<tr>
<td>2241</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 4</td>
</tr>
<tr>
<td>2242</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 4</td>
</tr>
<tr>
<td>2243</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2244</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2245</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2246</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2247</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2250</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2250</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2251</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2252</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2253</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2254</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2255</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2256</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2257</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2258</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2259</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2260</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2261</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2262</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2263</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2264</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2265</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2266</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2267</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2268</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2269</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2270</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2271</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2272</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2273</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2274</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2275</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2276</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2277</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2278</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2279</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2280</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2281</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2282</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2283</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2284</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2285</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2286</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2287</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2288</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2289</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2290</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2291</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2292</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2293</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2294</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2295</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2296</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2297</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2298</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2299</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2300</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2301</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2302</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2303</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2304</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2305</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2306</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2307</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2308</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2309</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2310</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2311</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2312</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
<tr>
<td>2313</td>
<td>00</td>
<td>MOVENVFMT</td>
<td>DBN002 2</td>
</tr>
</tbody>
</table>
```
2396.00 C*  Load error message array.
2397.00 C*  Error message numbers from Data Dictionary
2398.00 C*  Move
2399.00 C*  'ER0001'  ERN.01  Inv Action
2400.00 C*  'ER0002'  ERN.02  Inv Key
2401.00 C*  'ER0003'  ERN.03  Inv Blank
2402.00 C*  'ER0004'  ERN.04  Inv Date
2403.00 C*  'ER0005'  ERN.05  Inv Heart Wbr
2404.00 C*  'ER0006'  ERN.06  In Use
2405.00 C*  'ER0007'  ERN.07  Inv Volvo
2406.00 C*  'ER0008'  ERN.08  Inv MCU
2407.00 C*  'ER0009'  ERN.09  Inv Decr Ttl
2408.00 C*  'ER0010'  ERN.10  Inv Volvo
2409.00 C*  Load invalid action code array.
2410.00 C*  Lockout action code function used with the Program Generator
2411.00 C*  Move
2412.00 C*  'M0012'  M0012
2413.00 C*  'M0013'  M0013
2414.00 C*  'M0014'  M0014
2415.00 C*  'M0015'  M0015
2416.00 C*  Load system date.
2417.00 C*  Use the TIME feature to allow for all data formats
2418.00 C*  Move
2419.00 C*  'SW0013'  SW0013
2420.00 C*  'SW0014'  SW0014
2421.00 C*  'SW0015'  SW0015
2422.00 C*  'SW0016'  SW0016
2423.00 C*  'SW0017'  SW0017
2424.00 C*  Call 'S0026'
2425.00 C*  Method of releasing master file record locks
2426.00 C*-----------------------------
User Spaces

About User Spaces

User spaces are IBM Operating System objects managed by Application Program Interfaces (APIs) to store data. User Object APIs create, manipulate, and delete user spaces and indexes. An API provides you with:

- A faster method of retrieving information
- A means of dynamically modifying sizes
- A means of manipulating user objects

You should place your user spaces in library QTEMP so that it is deleted automatically when the user signs off.

In this chapter you will learn the following about user spaces.

- What Is a User Space?
- What Are the Advantages of Using a User Space?
- How Does a User Space Function?
- Creating a User Space
- Writing to a User Space
- Reading from a User Space

What Is a User Space?

A user space is an object made up of a collection of bytes that are used for storing any user defined information.

When you use a user space, there is no key to retrieve the information placed in the space. Therefore, the information in the user space is in the order that it was entered. A user space can store up to 16 megabytes of information.

**Note:** To see the contents of a user space, enter the command DMPOBJ (Dump Object) from any command line after the space has been loaded.

What Are the Advantages of Using a User Space?

The main advantage of using a user space is its speed. Because a user space consists of a string of bytes instead of elements like an array, you can write and retrieve records faster using a user space than an array.
In addition to speed, a user space provides you with more flexibility. A user space does not have a fixed record length. When you write a record to a user space, you define the length of that record. Therefore, each record you write to your user space can be a different size. In addition, it is possible to dynamically increase the size of your user space by calling the Enter User Space program (X00SPC) after creating the user space.

For example: @EX 999 30

The array @EX has a fixed record length of 30, therefore no record smaller or larger than 30 bytes can be written to this array.

User spaces are also used when communicating between two programs. The space can carry information loaded in one program to another program for retrieval.

For example: Program A creates the user space and loads information into a user space. Then Program A calls Program B and passes the name of the user space to it. Program B can retrieve information from the user space that was loaded by Program A.

How Does a User Space Function?

Remember that a user space is nothing more than a collection of bytes used to store information.

You write information to a user space, as well as retrieve information from it. Since there is no key associated with a user space, the information contained in a user space is in a user-defined order. The order is based on program controlled offset and length values.

Creating a User Space

To create a User Space

1. Determine if a user space already exists by using the JD Edwards World program J98CKOBJ.

For example:

<table>
<thead>
<tr>
<th>CALL</th>
<th>‘J98CKOBJ’</th>
<th>81</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSOBJ</td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSLIB</td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSTYPE</td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSMID</td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSAUT</td>
<td></td>
</tr>
<tr>
<td>PARM</td>
<td>PSERR</td>
<td></td>
</tr>
</tbody>
</table>
PARM (Length) | Description
---|---
PSOBJ (10) | The name of your user space.
PSLIB (10) | The name of the library in which you wish to check for the existence of the user space. Generally, this is *LIBL to check all of the libraries in the library list.
PSTYPE (8) | The type of object you are checking for. Generally, this is *USRSPC for a user space.
PSMID (10) | The member ID if you are checking for a database file. Generally, this is *NONE.
PSAUT (10) | The authority or authorization list to be checked for the user. Generally, this is *NONE.
PSERR (1) | The error parameter that will indicate an error while checking your object. Generally, this is initialized with *BLANK.

0 | No authority
1 | Not found
3 | No library
4 | Member not found
5 | No authority to library
6 | Cannot assign library

2. Do one of the following:

- If a user space does exist you should clear it and write your new information over the old.
- If the user space does not exist and no errors occurred, you can create your user space. To create a user space, use the QUSCRTUS (Create User Space) command.

For example:

```
CALL 'QUSCRTUS' 81
```

```
*SPNAM
*SPATT
*SPSIZ
*SPVAL
*SPAUT
*SPTXT
```
## User Spaces

### Description

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SPNAM (20)</td>
<td>The first 10 characters contain your user space name, and the second 10 characters contain the name of the library where your user space is located. Remember; place your user space in library QTEMP to automatically delete your space when you sign off.</td>
</tr>
<tr>
<td>#SPATT (10)</td>
<td>The extended attribute of your user space. You may use this field to classify your user space. For example, JD Edwards World uses this field to label all of the user spaces with JD Edwards World.</td>
</tr>
<tr>
<td>#SPSIZ (4 binary)</td>
<td>The initial size of your user space. Any value from 1 byte to 16 megabytes.</td>
</tr>
<tr>
<td>#SPVAL (1)</td>
<td>The initial value of all bytes in the user space. Generally, this is *BLANK.</td>
</tr>
<tr>
<td>#SPAUT (10)</td>
<td>The authority you give users to your user space. Generally, this is *ALL.</td>
</tr>
<tr>
<td>#SPTXT (50)</td>
<td>The text description of your user space.</td>
</tr>
</tbody>
</table>

To dynamically increase the size of your user space when maximum allocation is reached, call the Enlarge User Space program (X00SPC).

```
For example: CALL 'X00SPC' 81
             ---- ---- ---- ----
             PARM #XSPCN
             PARM #XRQSZ
             PARM #XERR
```

### Description

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#XSPCN (20)</td>
<td>The first 10 characters contain your user space name, and the second 10 characters contain the name of the library where your user space is located.</td>
</tr>
<tr>
<td>#XRQSZ (15,0)</td>
<td>The requested size to increase your space.</td>
</tr>
<tr>
<td>#XERR (1)</td>
<td>An error flag:</td>
</tr>
<tr>
<td></td>
<td>1 Space not found</td>
</tr>
<tr>
<td></td>
<td>2 Not authorized</td>
</tr>
<tr>
<td></td>
<td>3 Error</td>
</tr>
</tbody>
</table>

**JD Edwards World, A9.1**
Writing to a User Space

To write to a User Space

Use either the QUSCHGUS or the X98CHGUS (Change User Space) command.

For example:

```
CALL 'QUSCHGUS' 81
```

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SPNAM (20)</td>
<td>The first 10 characters contain your user space name, and the second 10 characters contain the name of the library where your user space is located.</td>
</tr>
<tr>
<td>#SPPOS</td>
<td>The starting position in your user space where the information will begin. It must be the first byte and must have a value greater than 0.</td>
</tr>
<tr>
<td>#SPLGH</td>
<td>The length of the information that is being written to your user space. This field is user-defined, but it must be greater than 0.</td>
</tr>
<tr>
<td>#SPVAL</td>
<td>The actual information to be written to your user space. The field must be at least as long as the length parameter. (* user defined)</td>
</tr>
<tr>
<td>#SPAUX (1)</td>
<td>Used to force changes made to your user space to auxiliary storage, such as a disk. The valid values are: 0 do not force changes 1 write changes 2 write changes immediately</td>
</tr>
</tbody>
</table>

_Note:_ The X98CHGUS program, JD Edwards World’s version of the IBM command QUSCHGUS, will perform a transfer control to QUSCHGUS.
Tracking Information if Writing Variable Length Records

Method 1

During the process of writing information to your user space, you should keep track of a pointer. This will ensure that you will not overwrite information or retrieve incorrect information.

One way to keep track of the pointer is to initialize it to 1. After you write information to your user space, add the length of the information to your pointer. The pointer is now set at the next starting point and ready for you to enter new information.

If the information you are writing to your user space contains various lengths, you should maintain the length of each piece of information in save fields. You can use the save fields when you wish to retrieve the information from your user space.

<table>
<thead>
<tr>
<th>Initialize pointer to 1</th>
<th>Add 30 bytes to pointer and to save field</th>
<th>Add 41 bytes to pointer and to another save field</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>A - 30 bytes - &quot; &quot;</td>
<td>B</td>
</tr>
<tr>
<td>&quot; &quot;</td>
<td></td>
<td>A - 41 bytes - B</td>
</tr>
</tbody>
</table>

Method 2

You can also reserve the first 2 or 3 bytes of every record for the size of that record. Then you would only have to load that part of the record with its length. When you read the record from the user space, the first 2 or 3 bytes will tell you how long the record is.

Reading from a User Space

Once you have loaded information into your user space, you are ready to retrieve it. Do not forget that your pointer must be set to the proper starting position to ensure the correct information is retrieved.

To read from a User Space

Use the QUSRTVUS (Retrieve User Space) command.

For example:

```
CALL 'QUSRTVUS' 31
       . . . . . . . .
       PARM #SPNAM
       PARM #SPPOS
       PARM #SPLOH
       PARM #SPREC
```
<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SPNAM (20)</td>
<td>The first 10 characters contain your user space name, and the second 10 characters contain the name of the library where your user space is located.</td>
</tr>
<tr>
<td>#SPPOS</td>
<td>The starting position in your user space where the information will begin. It must be the first byte and must have a value greater than 0.</td>
</tr>
<tr>
<td>(4 binary)</td>
<td></td>
</tr>
<tr>
<td>#SPLGH</td>
<td>The length of the information that is being retrieved to your user space. This field is user-defined, it must not be larger than the variable that will receive the information, and it must be greater than 0.</td>
</tr>
<tr>
<td>(4 binary)</td>
<td></td>
</tr>
<tr>
<td>#SPREC</td>
<td>The variable that will receive the information from your user space.</td>
</tr>
<tr>
<td>(* user defined)</td>
<td></td>
</tr>
</tbody>
</table>
User Indices

About User Indices

A user index is an object that will:

- Store data
- Allow search functions
- Automatically sort data based on its value

When you use a user index you must have a key to retrieve the information placed in the index. The key must be unique and you can only retrieve data using the key in ascending or descending order.

When you enter data into a user index, it is placed in order according to its key value.

A user index can store up to 4 gigabytes of information. Each key and record within a user index can be 1 to 999 bytes long.

**Note:** To see the contents of a user index, enter the command DMPOBJ (Dump Object) from any command line after the index has been loaded. You should place your user indices in library QTEMP so that it will be deleted automatically when you sign off.

To work with user indices, perform the following tasks:

- Create a User Index
- Write to a User Index
- Retrieve from a User Index

This section contains the following:

- **What Are the Advantages of Using a User Index?**
- **How Does a User Index Function?**
- **Creating a User Index**
- **Writing to a User Index**
- **Appearance of Records**
- **Retrieving Data from a User Index**
What Are the Advantages of Using a User Index?

When you load data into your user index, it is automatically sorted for you. Based on your key for the index, the information is arranged according to its value. This will help streamline table searches, cross referencing, and the ordering of data.

The size flexibility of a user index is much better than an array because arrays have a fixed size. A user index is only as big as the information it contains at one time. User indices expand as you add data to them.

For example: @EX  999 30

The array @EX has a fixed size of approximately 3 kilobytes. Each record must be 30 bytes long and up to 999 records can be loaded. If you have 300 records loaded into @EX, you will waste approximately 2 kilobytes. On the other hand, if you have 1500 records to load, the program will error when record number 1000 is loaded. A user index would be able to accommodate both situations.

- A user index is able to retrieve records faster than an array.
- Although a user index may expand to hold more records, it will not contract when records are removed. If you load 100 records into a user index and then remove 50 of them, the user index will remain at the 100 record level size.

You may retrieve data from a user index in ascending order or descending order. When data is loaded into a user index, it is loaded in ascending order. This does not restrict you to retrieving it in this order.

How Does a User Index Function?

A user index stores data and allows you to retrieve it by a key, which must be unique. The data it stores is made up of a data structure that consists of several fields that you wish to store. A user index is capable of expanding when you add data to it.

JD Edwards World leaves the first byte in the user index blank for clearing purposes.

```
A - - - - - - - - Key - - - - - - ”    A - - - - - - Data - - - - ”
```

When using a user index you can create it, add data to it, remove data from it, and delete it.

User indices, like user spaces, should be created in your QTEMP library so you do not have to worry about deleting them.
Creating a User Index

To check for user index

Before you actually create a user index, check to see if one already exists using the JD Edwards World program J98CKOBJ.

For example:

```
CALL 'J98CKOBJ' 81
       ---- ----
       FARM  PSOBJ
       FARM  PSLIB
       FARM  PSTYPE
       FARM  PSMID
       FARM  PSAUT
       FARM  PSERR
```

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSOBJ (10)</td>
<td>The name of your user index.</td>
</tr>
<tr>
<td>PSLIB (10)</td>
<td>The name of the library in which you wish to check for the</td>
</tr>
<tr>
<td></td>
<td>existence of the user index. Generally, this is *LIBL to</td>
</tr>
<tr>
<td></td>
<td>check all of the libraries in the library list.</td>
</tr>
<tr>
<td>PSTYPE (8)</td>
<td>The type of object you are checking for. Generally, this is</td>
</tr>
<tr>
<td></td>
<td>*USRIDX for a user index.</td>
</tr>
<tr>
<td>PSMID (10)</td>
<td>The member if you are checking for a database file.</td>
</tr>
<tr>
<td></td>
<td>Generally, this is *NONE.</td>
</tr>
<tr>
<td>PSAUT (10)</td>
<td>The authority or authorization list to be checked for the</td>
</tr>
<tr>
<td></td>
<td>user. Generally, this is *NONE.</td>
</tr>
</tbody>
</table>
If a user index exists, clear it and write your new information over the old.

For example:

```
    CALL  "X00IDX"  81
    
    PARM   #0XNAM
    PARM   #0XACT
    PARM   #0XRLN
    PARM   #0XKEY
    PARM   #0XRLN
    PARM   #0XREC
    PARM   #0XSTA
```

**Note:** The above code deletes (clears) every record with a key of *Blank in position one.

If the user index did not exist, you can now create your user index.

**To create a User Index**

Use the QUSCRTUI (Create User Index) command.

For example:

```
    CALL  "QUSCRTUI"  81
    
    PARM   #IDNAM
    PARM   #IDATT
    PARM   #IDENT
    PARM   #ILLEN
    PARM   #IDINS
    PARM   #IDKEY
    PARM   #IDUPD
    PARM   #IDOPT
    PARM   #IDAUT
    PARM   #IDTXT
```
<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#IDNAM (20)</td>
<td>The first 10 characters contain your user index name, and the second 10 characters contain the name of the library where your user index is located. Remember to place your user index in library QTEMP to automatically delete your index when you sign off.</td>
</tr>
<tr>
<td>#DATT (10)</td>
<td>The extended attribute of your user index. You may use this field to classify your user index. For example, JD Edwards World uses this field to label all of the user indexes with JD Edwards World.</td>
</tr>
<tr>
<td>#IDENT (1)</td>
<td>Whether the records you are loading into your user index are Fixed-length (F) or Variable-length (V). Generally, this is set to ‘F’.</td>
</tr>
<tr>
<td>#IDLEN (4 binary)</td>
<td>The length of the records to be entered into your user index. For fixed-length records valid values are 1 to 999. For variable-length records, enter 0 for a key length of 1 to 120, or 1 for a key length of 1 to 999.</td>
</tr>
<tr>
<td>#DINS (1)</td>
<td>Whether you are loading your user index by a key or not. Generally, this is set to 1 to load your index by a key. A value of 0 means you are not loading your index by a key.</td>
</tr>
<tr>
<td>#DKEY (4 binary)</td>
<td>The length of your key. The first byte in your record must be the beginning of your key. The values are 1 to 999 or 0 for no key.</td>
</tr>
<tr>
<td>#DUPD (1)</td>
<td>Whether or not the data in your user index will be immediately updated. Each data change to your index is written to auxiliary storage. The values are 0 for no immediate update or 1 for immediate update. Generally, this is 0.</td>
</tr>
<tr>
<td>#DOPT (1)</td>
<td>The type of access in which to optimize your index. The values are 0 to optimize for random references or 1 to optimize for sequential references. Generally, this is 1.</td>
</tr>
<tr>
<td>#SPAUT (10)</td>
<td>The authority you give users to your user index. Generally, this is *ALL.</td>
</tr>
<tr>
<td>#SPTXT (50)</td>
<td>The text description of your user index.</td>
</tr>
</tbody>
</table>

**Note:** You may want to define data structures containing some of the information required for the parameters to avoid having to enter values. The user index name, record length, key length, and user index text are good examples.
Writing to a User Index

To write to a User Index

JD Edwards World provides an external program called User Index Server (X00IDX) to manipulate data for user index entries.

For example:  CALL  "X00IDX"  81

```
  PARM  #0XNAM
  PARM  #0XACT
  PARM  #0XRUL
  PARM  #0XKLN
  PARM  #0XKEY
  PARM  #0XRLN
  PARM  #0XREC
  PARM  #0XSTA
```

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#0XNAM (20)</td>
<td>The first 10 characters contain your user index name, and the second 10 characters contain the name of the library where your user index is located. Remember to place your user index in library QTEMP to automatically delete your index when you sign off.</td>
</tr>
<tr>
<td>#0XACT (1)</td>
<td>The action you want to perform on your user index. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>I  Inquire</td>
</tr>
<tr>
<td></td>
<td>A  Add</td>
</tr>
<tr>
<td></td>
<td>C  Change</td>
</tr>
<tr>
<td></td>
<td>D  Delete</td>
</tr>
<tr>
<td>#0XRUL (2)</td>
<td>The rule used to search your user index depending on the action to be performed. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>EQ   Equal to</td>
</tr>
<tr>
<td></td>
<td>GT   Greater than</td>
</tr>
<tr>
<td></td>
<td>LT   Less than</td>
</tr>
<tr>
<td></td>
<td>GE   Greater than or Equal to</td>
</tr>
<tr>
<td></td>
<td>LE   Less than or Equal to</td>
</tr>
</tbody>
</table>
Appearance of Records

The records added to your user index will appear in ascending order.

For example: You created a user index to keep track of your ice cream sales. Each record within your user index contains the total sales amount, item, item description, and cost center. The key for your user index consists of total sales amount and item (remember the key must be unique).

The following records are to be loaded into your user index:

<table>
<thead>
<tr>
<th>Total</th>
<th>Sales</th>
<th>Item</th>
<th>Description</th>
<th>Cost Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ 500.00</td>
<td>CHO</td>
<td>Chocolate</td>
<td>$ 500.00</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 250.00</td>
<td>STR</td>
<td>Strawberry</td>
<td>$ 250.00</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 750.00</td>
<td>C&amp;C</td>
<td>Cookies &amp; Cream</td>
<td>$ 750.00</td>
<td>Denver</td>
</tr>
<tr>
<td>$1200.00</td>
<td>VAN</td>
<td>Vanilla</td>
<td>$1200.00</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 400.00</td>
<td>ROC</td>
<td>Rocky Road</td>
<td>$ 400.00</td>
<td>Denver</td>
</tr>
</tbody>
</table>

Because the key to your user index is total sales amount and item, the records will be entered into your index in ascending order by total sales amount first, then item. So your user index will look like this:
**Ice Cream Sales Index**

<table>
<thead>
<tr>
<th>Total</th>
<th>Sales Item</th>
<th>Description</th>
<th>Cost Center</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ 250.00</td>
<td>STR</td>
<td>Strawberry</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 400.00</td>
<td>ROC</td>
<td>Rocky Road</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 500.00</td>
<td>CHO</td>
<td>Chocolate</td>
<td>Denver</td>
</tr>
<tr>
<td>$ 750.00</td>
<td>C&amp;C</td>
<td>Cookies &amp; Cream</td>
<td>Denver</td>
</tr>
<tr>
<td>$1200.00</td>
<td>VAN</td>
<td>Vanilla</td>
<td>Denver</td>
</tr>
</tbody>
</table>

**Retrieving Data from a User Index**

You can retrieve data in ascending or descending order.

**To retrieve data in Ascending Order**

1. Use the User Index Server (X00IDX).
2. Set the Action parm to inquire (I).
3. Set Rule to Equal to (EQ).
4. Set the Key to the first record (*FIRST).

For example:

```
CALL 'X00IDX'
```

```
FARM  #0XNAM 20
FARM  #0XACT 1
FARM  #0XRUL 2
FARM  #0XKLN 30
FARM  #0XKEY120
FARM  #0XRLN 30
FARM  #0XREC120
FARM  #0XSTA 1
```

5. To retrieve the next record, load the key with the current record's values and change your rule to "GT".
24BUser Indices

To retrieve data in descending order

1. Use the User Index Server (X00IDX).
2. Set the Action parm to inquire (I).
3. Set Rule to Equal to (EQ).
4. Set the Key to the first record (*LAST).

For example:

```
CALL 'X00IDX'

PARM #0XNAM 20
PARM 'I' #0XACT 1
PARM 'GT' #0XRUL 2
PARM #0XKLN 30
PARM #0XKEY120
PARM #0XRLN 30
PARM #0XREC120
PARM #0XSTA 1
```

5. To retrieve the next record, load the key with the current record’s values and change your rule to “LT”.

For example:

```
CALL 'X00IDX'

PARM #0XNAM 20
PARM 'I' #0XACT 1
PARM 'GT' #0XRUL 2
PARM #0XKLN 30
PARM #0XKEY120
PARM #0XRLN 30
PARM #0XREC120
PARM #0XSTA 1
```
For example:  CALL 'X00IDX'

- - -  - - -

FARM #0XNAM 20
FARM 'T' #0XACT 1
FARM 'LT' #0XRUL 2
FARM #0XKLN 30
FARM #0XKEY120
FARM #0XRLN 30
FARM #0XREC120
FARM #0XSTA 1
24BUser Indices

24B User Indices

83.00 I*
84.00 I*    copy member for composite common subroutine - COOCC
85.00 I*
87.00 I* /COPY JDMP1, IGENZ
88.00 I*-------------------------------------------------------------
89.00 I*
90.00 I*    data structures for user index.
91.00 I*-------------------------------------------------------------
92.00 I* * entry record
93.00 I*-------------------------------------------------------------
94.00 I*    BRIAN DC 1 1 $REXL
95.00 I* 2 2 $SIC0
96.00 I* 7 7 $SINX
97.00 I* 19 46 $SIZEL
98.00 I* 46 51 $SIZ01
100.00 I* 46 51 $SIZ01
101.00 I* 46 51 $SIZ01
102.00 I* 46 51 $SIZ01
103.00 I*   * entry length, name/library, text
104.00 I*
105.00 I*    DC 1 1 D4 54 8 1 400$REXL
106.00 I*  2 2 D4 54 8 1 400$REXL
107.00 I*  3 3 D4 54 8 1 400$REXL
108.00 I*  4 4 D4 54 8 1 400$REXL
109.00 I*-------------------------------------------------------------
110.00 I*    * partial key 1 a 2, full unique key saved.
111.00 I*
112.00 I*    DC 1 1 D4 54 8 1 400$REXL
113.00 I*  2 2 D4 54 8 1 400$REXL
114.00 I*  3 3 D4 54 8 1 400$REXL
115.00 I*-------------------------------------------------------------
116.00 I*    * data structure for file servers
117.00 I*-------------------------------------------------------------
118.00 I*-------------------------------------------------------------
119.00 I*-------------------------------------------------------------
120.00 I*    0*-------------------------------------------------------------------
121.00 I*    C*    MAINLINE PROGRAM
122.00 I*    C*-----------------------------------------------------------
123.00 I*    C*-------------------------------------------------------------------
124.00 I*    C*    process housekeeping.
125.00 I*      C*---------------------------------------------------------
126.00 I*    C*    20 20 SEND 9999
127.00 I*      C*---------------------------------------------------------
128.00 I*    C*    29 29 IF LN ON, AND PROG.
129.00 I*      C*---------------------------------------------------------
130.00 I*    C*    30 30 FORM CARRY'1' ENY
131.00 I*      C*---------------------------------------------------------
132.00 I*    C*    31 31 FORM CARRY'1' ENY
133.00 I*      C*---------------------------------------------------------
134.00 I*    C*    32 32 IF AUTOMATIC INQUIRY SET, PROCESS INQUIRY.
135.00 I*      C*---------------------------------------------------------
136.00 I*    C*    33 33 SAVTO CARRY'1' G003 24
137.00 I*      C*---------------------------------------------------------
138.00 I*    C*    34 34 END
139.00 I*      C*---------------------------------------------------------
140.00 I*    C*    35 35 BEGIN NORMAL PROGRAM PROCESSING.
141.00 I*      C*---------------------------------------------------------
142.00 I*    C*    36 36 FORM CARRY'0' ENY
143.00 I*      C*---------------------------------------------------------
144.00 I*    C*    37 37 IF SUBTLE PAGE DISPLAY NOT SET, SET SUBTLE PAGE DISPLAY.
145.00 I*      C*---------------------------------------------------------
146.00 I*    C*    38 38 SFRNC ZRC C
147.00 I*      C*---------------------------------------------------------
148.00 I*    C*    39 39 SFRNC ENY
149.00 I*      C*---------------------------------------------------------
150.00 I*    C*    40 40 END
151.00 I*      C*---------------------------------------------------------
152.00 I*    C*    41 41 IFR Y C
153.00 I*      C*---------------------------------------------------------
154.00 I*    C*    42 42 ENY
155.00 I*      C*---------------------------------------------------------
156.00 I*    C*    43 43 END
157.00 I*      C*---------------------------------------------------------
158.00 I*    C*    44 44 ENY
159.00 I*      C*---------------------------------------------------------
160.00 I*    C*    45 45 WRITE VIDEO SCREEN.
24BUser Indices

413.00 C* 3-scrollbar - cursor control return values
413.00 C*---------------------------------------------------------------
414.00 C* or format, find the field to update and move in the
415.00 C* returned value. If the format is a subfile, the record
416.00 C* to change is found in ERRH.
417.00 C*
418.00 C*
419.00 C* DOVVL SEEK
420.00 C*----------
421.00 C* **KVAL IPFQ &"SLAK"
422.00 C* NOTE &"SLAK**KVAL
423.00 C* END
424.00 C*
425.00 C*
426.00 C* return values for fields in format VINDESC
427.00 C*
428.00 C* **KRM IPFQ &"VINDESC"
429.00 C*
430.00 C* **FLUSH IPFQ &"ACTION"
431.00 C* NOVEP=AVL ACTION
432.00 C* GOTO MOVRSVL
433.00 C*----------
434.00 C*
435.00 C*
436.00 C* **FLUSH IPFQ &"VDCO"
437.00 C* VDCO
438.00 C* GOTO MOVRSVL
439.00 C*----------
440.00 C*
441.00 C* END
442.00 C*
443.00 C*
444.00 C* return values for fields in format VINDESC
445.00 C*
446.00 C* **KRM IPFQ &"VINDESC"
447.00 C* GOTO SERCTO
448.00 C*
449.00 C* GOTO MOVPEML
450.00 C* $1
451.00 C* *SER1 IPFQ &"VDCO"
452.00 C* GOTO MOVRSVL
453.00 C* *IN,1
454.00 C*
455.00 C*
456.00 C* **FLUSH IPFQ &"SPMU"
457.00 C* GOTO MOVRSVL
458.00 C*----------
459.00 C* END
460.00 C*
461.00 C* **FLUSH IPFQ &"SPLOI"
462.00 C* GOTO MOVRSVL
463.00 C*----------
464.00 C* END
465.00 C*
466.00 C* END
467.00 C* **FLUSH IPFQ &"SPPOI"
468.00 C* GOTO MOVRSVL
469.00 C*----------
470.00 C* END
471.00 C* END
472.00 C* **FLUSH IPFQ &"SPPOI"
473.00 C* GOTO MOVRSVL
474.00 C*----------
475.00 C* END
476.00 C* END
477.00 C* TOVVA TAG
478.00 C*----------
479.00 C*
480.00 C* GETCH
481.00 C* MOVRSVL CNTN $1
482.00 C* END
483.00 C* END
484.00 C* END
485.00 C*
486.00 C* return values for fields in format VINDESC
487.00 C*
488.00 C* **KRM IPFQ &"VINDESC"
489.00 C* END
490.00 C *
491.00 C*
492.00 C*---------------------------------------------------------------
493.00 C*
494.00 C* SE-Routines 2001 - clear fields

JD Edwards World, A9.1
24BUser Indices

495.00 C*  -----------------------------------------------
496.00 C*  
497.00 C*  PROCESSING:  1. reset all video screen and data file fields
498.00 C*  for next transaction.
499.00 C*  
500.00 C*  "                      
501.00 C*  clear action code only if requested.
502.00 C*  
503.00 C*  GSR 2001  BEGSR
504.00 C*  
505.00 C*  MOVE 'BLANK' $10L01
506.00 C*  MOVE 'BLANK' $10P01
507.00 C*  MOVE 'BLANK' $10P02
508.00 C*  MOVE 'BLANK' $10P02
509.00 C*  MOVE 'BLANK' $10L01
510.00 C*  MOVE 'BLANK' $10L01
511.00 C*  MOVE 'BLANK' $10P01
512.00 C*  MOVE 'BLANK' $10P02
513.00 C*  MOVE 'BLANK' $10P02
514.00 C*  MOVE 'BLANK' $10L01
515.00 C*  MOVE 'BLANK' $10P01
516.00 C*  MOVE 'BLANK' $10L02
517.00 C*  
518.00 C*  clear action code only if clear screen action.
519.00 C*  
520.00 C*  
521.00 C*  
522.00 C*  
523.00 C*  
524.00 C*  
525.00 C*  
526.00 C*  
527.00 C*  
528.00 C*  
529.00 C*  
530.00 C*  
531.00 C*  
532.00 C*  
533.00 C*  
534.00 C*  
535.00 C*  
536.00 C*  
537.00 C*  
538.00 C*  
539.00 C*  
540.00 C*  
541.00 C*  
542.00 C*  
543.00 C*  
544.00 C*  
545.00 C*  
546.00 C*  
547.00 C*  
548.00 C*  
549.00 C*  
550.00 C*  
551.00 C*  
552.00 C*  
553.00 C*  
554.00 C*  
555.00 C*  
556.00 C*  
557.00 C*  
558.00 C*  
559.00 C*  
560.00 C*  
561.00 C*  
562.00 C*  
563.00 C*  
564.00 C*  
565.00 C*  
566.00 C*  
567.00 C*  
568.00 C*  
569.00 C*  
570.00 C*  
571.00 C*  
572.00 C*  
573.00 C*  
574.00 C*  
575.00 C*  
576.00 C*  
577.00 C*  

Clear Data Structure
contains record format for User Index

<table>
<thead>
<tr>
<th>Address</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>575.00</td>
<td>C*</td>
</tr>
<tr>
<td>579.00</td>
<td>C*</td>
</tr>
<tr>
<td>581.00</td>
<td>C*</td>
</tr>
<tr>
<td>584.00</td>
<td>C*</td>
</tr>
<tr>
<td>566.00</td>
<td>C*</td>
</tr>
<tr>
<td>569.00</td>
<td>C*</td>
</tr>
<tr>
<td>581.00</td>
<td>C*</td>
</tr>
<tr>
<td>584.00</td>
<td>C*</td>
</tr>
<tr>
<td>586.00</td>
<td>C*</td>
</tr>
<tr>
<td>589.00</td>
<td>C*</td>
</tr>
<tr>
<td>592.00</td>
<td>C*</td>
</tr>
<tr>
<td>595.00</td>
<td>C*</td>
</tr>
</tbody>
</table>
24BUser Indices

C***********************************************************************
C
C  User Indices

C***********************************************************************

660.00 C
661.00 C*  copy common subroutine - right justify numeric fields
662.00 C*
663.00 C COPY JNUMFY, COO12
664.00 C***********************************************************************
665.00 C
666.00 C*
667.00 C* VARSUB004 - load video screen data
668.00 C*
669.00 C*
670.00 C* processing: 1. move data base information to video screen.
671.00 C* all video screen fields are alphas and
672.00 C* therefore numeric information must be
673.00 C* processed through subroutine ch04 to get
674.00 C* proper delimiters and provide editing for
675.00 C* display on screen.
676.00 C*
677.00 C* date fields must be converted from their
678.00 C* internal format of month, day and year or
679.00 C* Julian to the system format using program
680.00 C* ZD008.
681.00 C*
682.00 CGR 0004 BEGIN
683.00 C*
684.00 C* load data field dictionary parameters (one cycle only).
685.00 C*
686.00 C GR $98 CARGY ' $98
687.00 C*
688.00 C GR END
689.00 C*
690.00 C*
691.00 C* if subfile load completed, skip subroutine.
692.00 C*
693.00 C GR $ZEXEC IFED '1'
694.00 C*
695.00 C GR ZD008 IFB004
696.00 C GR END
697.00 C*
698.00 C*---------------------------------------------------------------------
699.00 C*
700.00 C* save company number for comparison later.
701.00 C*---------------------------------------------------------------------
702.00 C GR MOTH $200 $200 5
703.00 C*
704.00 C*
705.00 C* move to output - company description.
706.00 C*
707.00 C GR MOTH 'BLOCK' PDB
708.00 C GR MOTH 'SSUBF' PDB
709.00 C GR MOTH 'SUBF' PDB
710.00 C GR MOTH 'SUBEDIT' PDB
711.00 C GR MOTH 'SUBEDIT' PDB
712.00 C GR MOTH 'SUBEDIT' PDB
713.00 C*
714.00 C* initialize subfile page control and index.
715.00 C*
716.00 C GR Z-ADEO $PG 30
717.00 C GR Z-ADEO $RENO
718.00 C GR Z-ADEOPTY 1
719.00 C*---------------------------------------------------------------------
720.00 C*
721.00 C* read user index until end or subfile page filled.
722.00 C*
723.00 C GR STOP 16
724.00 C GR 'ZEXEC DIVEND 0'
725.00 C*
726.00 C* first time through, have already read first record, so skip
727.00 C* the index logic. if 'ZEXEC '1'
728.00 C*---------------------------------------------------------------------
729.00 C*
730.00 CGR $ZEXEC MOTH ' ' $7FANY7
731.00 CGR MOTH ' ' $7FANY7
732.00 CGR END

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 6-75
24BUse Indexs

733.00 C- successful times through, read next "greater" entry.
734.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
735.00 C- 2-ADDEQINVL PTRINVL 2-ADDEQINVR PTRINVR
736.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
737.00 C- CALL 'ADVIXS'
738.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
739.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
740.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
741.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
742.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
743.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
744.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
745.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
746.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
747.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
748.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
749.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
750.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
751.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
752.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
753.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
754.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
755.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
756.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
757.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
758.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
759.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
760.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
761.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
762.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
763.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
764.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
765.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
766.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
767.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
768.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
769.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
770.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
771.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
772.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
773.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
774.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
775.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
776.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
777.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
778.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
779.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
780.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
781.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
782.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
783.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
784.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
785.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
786.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
787.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
788.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
789.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
790.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
791.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
792.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
793.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
794.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
795.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
796.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
797.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
798.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
799.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
800.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
801.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
802.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
803.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
804.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
805.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
806.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
807.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
808.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
809.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
810.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
811.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
812.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
813.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
814.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
815.00 C- $%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

3-76

JD Edwards World, A9.1
24BUser Indices

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 6-77
24B User Indices

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 6-81
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1314</td>
<td>C</td>
<td>24BUser Indices</td>
</tr>
<tr>
<td>1315</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1316</td>
<td>C</td>
<td>Summarize User - load dictionary parameters.</td>
</tr>
<tr>
<td>1317</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1318</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1319</td>
<td>C</td>
<td>s000 $RESULT</td>
</tr>
<tr>
<td>1320</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1321</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1322</td>
<td>C</td>
<td>dictionary parameters for - description 01</td>
</tr>
<tr>
<td>1323</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1324</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1325</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1326</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1327</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1328</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1329</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1330</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1331</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1332</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1333</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1334</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1335</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1336</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1337</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1338</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1339</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1340</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1341</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1342</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1343</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1344</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1345</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1346</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1347</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1348</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1349</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1350</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1351</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1352</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1353</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1354</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1355</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1356</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1357</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1358</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1359</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1360</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1361</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1362</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1363</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1364</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1365</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1366</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1367</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1368</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1369</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1370</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1371</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1372</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1373</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1374</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1375</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1376</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1377</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1378</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1379</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1380</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1381</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1382</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>1383</td>
<td>C</td>
<td></td>
</tr>
</tbody>
</table>
CUNIT 1336 00
CUNIT 1337 00
CUNIT 1338 00
CUNIT 1339 00
CUNIT 1400 00
CUNIT 1401 00
CUNIT 1402 00
CUNIT 1403 00
CUNIT 1404 00
CUNIT 1405 00
CUNIT 1406 00
CUNIT 1407 00
CUNIT 1408 00
CUNIT 1409 00
CUNIT 1410 00
CUNIT 1411 00
CUNIT 1412 00
CUNIT 1413 00
CUNIT 1414 00
CUNIT 1415 00
CUNIT 1416 00
CUNIT 1417 00
CUNIT 1418 00
CUNIT 1419 00
CUNIT 1420 00
CUNIT 1421 00
CUNIT 1422 00
CUNIT 1423 00
CUNIT 1424 00
CUNIT 1425 00
CUNIT 1426 00
CUNIT 1427 00
CUNIT 1428 00
CUNIT 1429 00
CUNIT 1430 00
CUNIT 1431 00
CUNIT 1432 00
CUNIT 1433 00
CUNIT 1434 00
CUNIT 1435 00
CUNIT 1436 00
CUNIT 1437 00
CUNIT 1438 00
CUNIT 1439 00
CUNIT 1440 00
CUNIT 1441 00
CUNIT 1442 00
CUNIT 1443 00
CUNIT 1444 00
CUNIT 1445 00
CUNIT 1446 00
CUNIT 1447 00
CUNIT 1448 00
CUNIT 1449 00
CUNIT 1450 00
CUNIT 1451 00
CUNIT 1452 00
CUNIT 1453 00
CUNIT 1454 00
CUNIT 1455 00
CUNIT 1456 00
CUNIT 1457 00
CUNIT 1458 00
CUNIT 1459 00
CUNIT 1460 00
CUNIT 1461 00
CUNIT 1462 00
CUNIT 1463 00
CUNIT 1464 00
CUNIT 1465 00
CUNIT 1466 00
CUNIT 1467 00
CUNIT 1468 00
CUNIT 1469 00
CUNIT 1470 00
CUNIT 1471 00
CUNIT 1472 00
CUNIT 1473 00
CUNIT 1474 00

24BUser Indices

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 6-83
create or clear the demonstration user index

Check to see if user index already exists

If it doesn't exist, create it.

Check error status parameter to see if User Index exists

Load key length, record length, and key so clear User Index if it already exists

User Index

set subroutine execution flag.

-----------

processing:

1. Load video screen text.

2. Retrieve screen title data area, title and move to video screen.

3. Initialize key line.

4. Load roll bars.

5. Display parameters.


7. Initialize subtitle display.

8. Display...

... no parameters passed
24BUser Indices

1557.00 C*  Test for auto inquiry function.
1558.00 C*
1559.00 CUR  SANSO  IPAC *BLANK
1560.00 CUR  MOVES '1'  SANSO  1
1561.00 CUR  END
1562.00 C*-----------------------------------------------
1563.00 C* Load video screen text.
1564.00 C*
1565.10 CUR  MOVESCREEN  PAPER  10
1566.00 CUR  Z-ADDCC  $PUNIT$  30
1567.00 C*/COPY JISXEPY,COSEC
1568.00 C*
1569.00 C*-----------------------------------------------
1570.00 C* Load error messages array.
1571.00 C*
1572.00 CUR  MOVES '0001'  ERR,01  INV Action
1573.00 CUR  MOVES '0002'  ERR,02  INV Resp
1574.00 CUR  MOVES '0003'  ERR,03  INVplied
1575.00 CUR  MOVES '0004'  ERR,04  INV Data
1576.00 CUR  MOVES '0005'  ERR,05  INV MAXR MTR
1577.00 CUR  MOVES '0007'  ERR,07  IN USR
1578.00 CUR  MOVES '0025'  ERR,07  INV Values
1579.00 CUR  MOVES '0026'  ERR,08  INV NuC
1580.00 CUR  MOVES '0027'  ERR,09  INV Dsct Ttl
1581.00 CUR  MOVES '0028'  ERR,09  INV NCD
1582.00 C*-----------------------------------------------
1583.00 C* Load invalid action code array.
1584.00 C*
1585.10 CUR  MOVES ' '  $MC
1586.00 C*-----------------------------------------------
1587.00 C* Initialize subtitle display.
1588.00 C*
1589.00 CUR  Z-ADDC  11
1590.00 CUR  Z-ADDC  30
1591.10 CUR  DO  DOPE
1592.00 CUR  ADD  1  11
1593.00 CUR  MOVES'IN'  SHIN
1594.00 CUR  WRITESHIN
1595.00 CUR  END
1596.00 CUR  Z-ADDCC  $MC
1597.00 CUR  MOVES ' '  $MC
1598.00 C*-----------------------------------------------
1600.00 C* Load system data.
1601.00 C*
1602.00 CUR  TIME  $SWHLS$  120
1603.00 CUR  MOVES $SWHLS$  $TIME$  80
1604.00 C*-----------------------------------------------
1605.00 CUR  $MC9999$  ENDER
About File Servers

- Sometimes called I/O servers.
- Allow you to enhance the processing time of your program.
- Ease the maintenance of your programs by making your system more modular.

Eventually, every program should perform database functions using either a file server or a functional server.

Note: all logical files are accessed through servers by their based-on file. Embedded in one server, there may be many access paths available.

This section contains the following:

- What is a File Server?
- What are the Advantages of Using a File Server?
- What are the Disadvantages of Using a File Server?
- How Does a File Server Function?
- What Are Control Parameters?
- What Are Returned Parameters?
- Implementing a File Server
- Searching for Key Lists
- Tips when Using File Servers
- Commonly Used File Servers

What is a File Server?

A file server, or I/O server, is a server that performs RPG database operations.

This type of server has no effect on program logic, but it isolates the actual database from the application program. Once you implement a file server into a program, the file specification is no longer required.
Types of File Servers

There are three types of file servers you can use:

<table>
<thead>
<tr>
<th>File Server</th>
<th>Description</th>
</tr>
</thead>
</table>
| XS          | Input-Only/Caching Servers  
They can be used in place of a simple CHAIN operation for input only. You may request descriptions only, or the entire record. They provide caching logic to decrease physical I/O for duplicate requests. |
| XF          | Input/Output File Servers  
They allow you to replace all RPG database operation codes for a given file with program calls. They can perform such functions as READ, CHAIN, and SETLL to a file. |
| X           | Special Scrub & Edit Servers  
They can accept cost center, account numbers, numeric fields, or a data string in any valid data entry or file format, convert them to any format, validate the existence of the master record, and optionally pass the master record or the scrubbed data back to the calling program. |

What are the Advantages of Using a File Server?

The advantages of using a file server are:

- Minimized maintenance of your software
- Ability to change a physical file without having to make changes to application programs that use the file, or even having to recompile them
- Use of versions in future releases to allow programs from a previous release to run against a changed database
- Smoother transition from an old database to a new database. Instead of applying all new programs, you will only have to apply a new set of file servers
- Ability to implement one file server at a time without affecting the rest of your system

What are the Disadvantages of Using a File Server?

The disadvantages are:

- A file server is minutely slower because you are performing an external call to the server from your program
- File server programs tend to be large

File servers are designed to perform all database functions that can be performed directly.
How Does a File Server Function?

A file server performs all the interfaces between a program and file. After you load the control parameters, which contain information about the record you are retrieving, the file server converts the control parameters and returns a record to the program.

If you plan to use any of the file server programs and you are asking them to return the database record, you must use the record image / COPY member that the corresponding I/O server uses. The / COPY member has the following naming convention:

I(file name) (release level).

For example: The copy member for the F0101 record image should appear as:

I/ COPY JDECPY, I010171

Note: Some technical file servers (X9800E, X0005) have a / COPY member with the naming convention:

I(file name)(special character)
I/ COPY JDECPY, I0005U

A file server may be called with two parameters:

For example:

<table>
<thead>
<tr>
<th>PARM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PS@@1</td>
<td>Contains all of the control parameters. It is contained in copy module I00XFSRV, and it is common to all file servers.</td>
</tr>
<tr>
<td>I(file name)</td>
<td>Contains the record image for updates and writes specific for each I/O server. It is an exact duplicate of the record image. It is contained in the copy module I(file name) (release level).</td>
</tr>
</tbody>
</table>

What Are Control Parameters?

The parameter PS@@1 is a Data Structure which contains all the control parameters for the file server. All control parameters, except the format name, are cleared every time the server returns control to the calling program. You must set the parameter values every time the server is called unless you are satisfied with the default values.
### PARM (Length) Description

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@ACCS (1)</td>
<td>The type of access to the file. The valid values are K for Keyed access (default), R for relative record number access and S for sequential access (DREAM Writer).</td>
</tr>
<tr>
<td>@@OPER (10)</td>
<td>The operation to be performed to the file. The valid values are presented below: CHAIN Chain by key list or RRN CLOSE Close the access path DELET Delete current record or by key or RRN EXIST Test existence of record by key OPEN Open access path (optional) READ Read next record READE Read next equal key READP Read previous record REDPE Read previous equal key SETGT Set greater than key SETHV Set greater than with *HIVAL SETLL Set lower limit by key SETLV Set lower limit with *LOVAL UPDAT Update locked record UPDATC Update current record WRITE Write new record UNLCK Unlock current record</td>
</tr>
<tr>
<td>@@LOCK (1)</td>
<td>Whether you do or do not want to lock the record. The valid values are Y to lock the record (default) or N to not lock the record. <strong>Note:</strong> This parameter is only valid for chain and read operations, and is ignored for all other operations. You should set it to N when reading records not to be updated.</td>
</tr>
<tr>
<td>PARM (Length)</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>@@CHGR (1)</td>
<td>Servers allow records to be read without lock and then be updated (UPDATC). In this situation, the record will be re-read before it is updated and if it has changed since it was last read, action will have to be taken. This parameter determines what that action will be. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>N Do not update the record. A return code (RC) is returned and it comes up to the program to determine what action to take. (default)</td>
</tr>
<tr>
<td></td>
<td>O Overlay the changed record with the values you are currently updating. This will cause the changes made by the other user to be lost.</td>
</tr>
<tr>
<td></td>
<td>W Call the Changed Record Window (P0000U) that will prompt you for what action to take. Use this option with interactive programs only.</td>
</tr>
<tr>
<td>@@KLST (10)</td>
<td>The key list that will be used for access. The calling program does not specify a logical file so that the application program is isolated from any database changes. A value must be specified unless you are accessing the file by relative record number or sequentially (@@ACCS = R or S). Note: The server maintains status information for each access path, so multiple paths can be accessed through the server in one program. The @@KLST parameter determines which access path is affected by the current call to the server.</td>
</tr>
<tr>
<td>@@KNUM (5,0)</td>
<td>Specifies how many key fields in the list will be used for the current operation. This allows you to perform a read equal by a partial key. The valid values are 1 through the number of fields in the key, and blank for operations not requiring a key.</td>
</tr>
<tr>
<td>@@FMT (10)</td>
<td>Specifies the release level the program is expecting. This field does not get cleared upon returning from the server, so it can be set once in S999.</td>
</tr>
<tr>
<td>@@#RRN (9,0)</td>
<td>The relative record number for RRN access.</td>
</tr>
<tr>
<td>I (file name)</td>
<td>Record image for updates and writes. This parameter is optional for OPEN, CLOSE, DELETE, SETHV, SETLV, and UNLK operations.</td>
</tr>
</tbody>
</table>

Access paths are opened automatically when the first operation is performed. Therefore, it is not necessary to call the server with the OPEN operation.

A server normally remains active as long as the calling program is active. If you know you will need a server for only a limited period of time and do not want it taking up space in the PAG, you can call the server the @@OPER parameter blank, this causes the server to return and end.
What Are Returned Parameters?

When the file server returns the record to the program, there are several parameters associated with it.

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@@IOR(3)</td>
<td>The I/O return code. The possible values are: blank No errors NF Record not found NE Not equal for a READE operation EOF End of file EQ Equal for a SETLL operation BOF Beginning of file RL Record Locked, could not read RC Record changed YES Record found NO Record not found ERR Error, check error fields for explanation</td>
</tr>
<tr>
<td>@@ERR (10)</td>
<td>Short description of the cause of the problem (invalid, redock, error, required, deleted, chgrec).</td>
</tr>
<tr>
<td>@@ERRS (10)</td>
<td>The subject causing the error. The value could be a parameter (KLST), an operation (OPEN), or a file name (Fxoox). Used in combination with @@ERR gives a good idea of what happened. The application program will generally only use @@IOR. @@ERR and @@ERRS are most useful for debugging purposes.</td>
</tr>
<tr>
<td>@@#RRN (9,0)</td>
<td>Returns the relative record number of the record just read (both input and output).</td>
</tr>
<tr>
<td>I (filename)</td>
<td>Returns an exact duplicate of the record image (both input and output).</td>
</tr>
</tbody>
</table>

Implementing a File Server

To implement a file server

The following are generally the steps needed to set up a file server in a program. Some programs may differ.

1. Remove F-spec line for file being accessed through the server, and replace it with a comment mentioning access through the server.
2. Add clear statement in S999 (CLEAR PS@1). You can optionally set @@FMT to the JD Edwards base release value ("A73" or "A81") so it does not have to be set on every call.

3. Copy in I-spec copy module 100XFSRV.

4. Copy in I-spec copy module for the required server, following the naming convention: I(file name) (release level). For example, 1010171.

5. Code call to server for each database access. Naming convention for server is X(file name). For example, XF0101 for F0101 and any of its logicals.
   - Load control parameters
   - Load record image if a write or update
   - Call the server
   - Check the return code

6. Remove any open statements and key lists for this file from S999 in the calling program.

7. Remove any output specifications dealing with EXCPT unlock statements at the bottom of the program. The server will handle all of the unlock and lock operations.

---

**Note:** When reading sequentially (@@ACCS = S) through the physical file or through a DREAM Writer based-on file that is overridden to the physical, some operations are not available. Do not use: CHAIN, EXIST, READE, REDPE, UPDATC, SETLL, SETGT, SETHV, SETLV. Since UPDATC is not available and you are going to update a record, you need to read with lock.

If the file you are accessing though the server is the DREAM Writer based on file, the Open Query Options on the DREAM Writer Additional Parameters screen need to be changed. Change all of the “Open for xxxxx” parameters to “Y” on the DW Additional Parameters screen.

---

**Searching for Key Lists**

When converting programs to use the file servers, make note of what logical files are being accessed, and what mode (update or input) and what each of the defined key lists for those access paths represent. There are some servers with information in the F93201 which is accessed by the following process. Otherwise, using the Where Used function on a file entry in SVR may show which “X” (server) programs access a particular file.

**To search for Key Lists**

1. Look up the corresponding server key list name using P93KL (fast path, KL).
2. Search for the format name for files that are accessed in the program.
3. Replace each instance of file access code with a call to the server with the correct parameters.
Tips when Using File Servers

The following tips can help when using file servers:

- When converting a program to use the file servers, always set the `@@LOCK` parameter to “N” when reading records through an access path that the program uses to open for input only.

  The reason for this is that all access paths are open for update in the server. This can cause record lock problems when a program opens multiple paths into the same file. Correct use of the `@@LOCK` parameter solves these problems.

- Some programs may be doing a CHAIN or EXCPT to unlock a record. Instead of replacing it with a CHAIN through the server, take advantage of the UNLCK operation. Performing an UNLCK on a file that does not have a record locked does not produce an error.

- Some programs perform a SETLL to validate that a record exists. The new operation EXIST is provided to handle this function. It returns a YES or NO in return code (`@@IOR`).

- There is only one instance in which a particular file server is active in your job, so if one program calls another program that accesses the file through the same access path, they are actually sharing the same open data path. If it is possible that a call to another program could relocate a file pointer that could error out the program, it would be a good idea to save the keys and reset the pointer (CHAIN or SETLL) upon returning.
## File Server Examples

<table>
<thead>
<tr>
<th>Seq No.</th>
<th>C*</th>
<th>Code Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>C</td>
<td>C Index of Examples:</td>
<td></td>
</tr>
<tr>
<td>4.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>5.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>6.00</td>
<td>C</td>
<td>C* Scan For: To Find:</td>
<td></td>
</tr>
<tr>
<td>7.00</td>
<td>C</td>
<td>C*-------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>8.00</td>
<td>C</td>
<td>1.1 File Server Calls</td>
<td></td>
</tr>
<tr>
<td>9.00</td>
<td>C</td>
<td>1.1.1 Chain</td>
<td></td>
</tr>
<tr>
<td>10.00</td>
<td>C</td>
<td>1.1.2 Close</td>
<td></td>
</tr>
<tr>
<td>11.00</td>
<td>C</td>
<td>1.1.3 Delete</td>
<td></td>
</tr>
<tr>
<td>12.00</td>
<td>C</td>
<td>1.1.4 Deletion Text</td>
<td></td>
</tr>
<tr>
<td>13.00</td>
<td>C</td>
<td>1.1.5 Open</td>
<td></td>
</tr>
<tr>
<td>14.00</td>
<td>C</td>
<td>1.1.6 Read</td>
<td></td>
</tr>
<tr>
<td>15.00</td>
<td>C</td>
<td>1.1.7 Read Equal</td>
<td></td>
</tr>
<tr>
<td>16.00</td>
<td>C</td>
<td>1.1.8 Read Previous</td>
<td></td>
</tr>
<tr>
<td>17.00</td>
<td>C</td>
<td>1.1.9 Read Template</td>
<td></td>
</tr>
<tr>
<td>18.00</td>
<td>C</td>
<td>1.1.10 Get Greater Than</td>
<td></td>
</tr>
<tr>
<td>19.00</td>
<td>C</td>
<td>1.1.11 Get Previous Level</td>
<td></td>
</tr>
<tr>
<td>20.00</td>
<td>C</td>
<td>1.1.12 Update</td>
<td></td>
</tr>
<tr>
<td>21.00</td>
<td>C</td>
<td>1.1.13 Write</td>
<td></td>
</tr>
<tr>
<td>22.00</td>
<td>C</td>
<td>1.1.14 Unlock</td>
<td></td>
</tr>
<tr>
<td>23.00</td>
<td>C</td>
<td>2.1.1 Z90031</td>
<td></td>
</tr>
<tr>
<td>24.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>25.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>26.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>27.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>28.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>29.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>30.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>31.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>32.00</td>
<td>C</td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

---

Advanced Programming Concepts & Skills Guide (Revised - January 15, 2007) 6-95
<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
<th>Address</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>CALL 'FX0101'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>CALL 'FAMX'</td>
<td>FSS1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>CALL 'Delete'</td>
<td>10101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>Old Code:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>New Code:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>MOVEL 'A1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>MOVEL 'A2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>MOVEL 'A3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>MOVEL 'A4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>MOVEL 'A5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>MOVEL 'A6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>MOVEL 'A7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>MOVEL 'A8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>MOVEL 'A9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>MOVEL 'A0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>MOVEL 'A1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>101</td>
<td>MOVEL 'A2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>102</td>
<td>MOVEL 'A3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>103</td>
<td>MOVEL 'A4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>104</td>
<td>MOVEL 'A5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>105</td>
<td>MOVEL 'A6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>106</td>
<td>MOVEL 'A7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>MOVEL 'A8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>MOVEL 'A9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>109</td>
<td>MOVEL 'A0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>MOVEL 'B1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>111</td>
<td>MOVEL 'B2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>MOVEL 'B3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>MOVEL 'B4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>MOVEL 'B5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>MOVEL 'B6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>MOVEL 'B7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>MOVEL 'B8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>MOVEL 'B9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>MOVEL 'C1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>MOVEL 'C2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>MOVEL 'C3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>MOVEL 'C4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>MOVEL 'C5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>MOVEL 'C6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>MOVEL 'C7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>MOVEL 'C8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>MOVEL 'C9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>MOVEL 'D1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>MOVEL 'D2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>MOVEL 'D3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>MOVEL 'D4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>132</td>
<td>MOVEL 'D5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>133</td>
<td>MOVEL 'D6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>MOVEL 'D7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>135</td>
<td>MOVEL 'D8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>136</td>
<td>MOVEL 'D9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>137</td>
<td>MOVEL 'A1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>138</td>
<td>MOVEL 'A2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>MOVEL 'A3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>MOVEL 'A4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141</td>
<td>MOVEL 'A5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>142</td>
<td>MOVEL 'A6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>143</td>
<td>MOVEL 'A7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144</td>
<td>MOVEL 'A8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145</td>
<td>MOVEL 'A9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146</td>
<td>MOVEL 'A0'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>147</td>
<td>MOVEL 'B1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>MOVEL 'B2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>MOVEL 'B3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>MOVEL 'B4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>MOVEL 'B5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>152</td>
<td>MOVEL 'B6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>153</td>
<td>MOVEL 'B7'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>MOVEL 'B8'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>MOVEL 'B9'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>MOVEL 'C1'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>MOVEL 'C2'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>MOVEL 'C3'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>MOVEL 'C4'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>MOVEL 'C5'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>MOVEL 'C6'</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>162</td>
<td>MOVEL 'C7'</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

JD Edwards World, A9.1
<table>
<thead>
<tr>
<th>Line</th>
<th>CSE</th>
<th>OP</th>
<th>Operand 1</th>
<th>Operand 2</th>
<th>Next Line</th>
</tr>
</thead>
<tbody>
<tr>
<td>160.00</td>
<td>CALL 'RPL101'</td>
<td>161.00</td>
<td>162.00</td>
<td>163.00</td>
<td>164.00</td>
</tr>
<tr>
<td>161.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>162.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>163.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>164.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>165.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>166.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>167.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>168.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>169.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>170.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>171.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>172.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>173.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>174.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>175.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>176.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>177.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>178.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>179.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>180.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>181.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>182.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>183.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>184.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>185.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>186.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>187.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>188.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>189.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>190.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>191.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>192.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>193.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>194.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>195.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>196.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>197.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>198.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>199.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>200.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>201.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>202.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>203.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>204.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>205.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>206.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>207.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>208.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>209.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>210.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>211.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>212.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>213.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>214.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>215.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>216.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>217.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>218.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>219.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>220.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>221.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>222.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>223.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>224.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>225.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>226.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>227.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>228.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>229.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>230.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>231.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>232.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>233.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>234.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>235.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>236.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>237.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>238.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>239.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>240.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>241.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>242.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>243.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>244.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>245.00</td>
<td>C8</td>
<td>0</td>
<td>87</td>
<td>99</td>
<td>0</td>
</tr>
<tr>
<td>Line</td>
<td>CER</td>
<td>Command</td>
<td>Notes</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----</td>
<td>---------</td>
<td>-------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>246.0</td>
<td>CER</td>
<td>MOVEL'FILL'</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>247.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>248.0</td>
<td>CER</td>
<td>CALL 'XP0101'</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>249.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>250.0</td>
<td>CER</td>
<td>UPDATE</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>251.0</td>
<td>CER</td>
<td>E-AD01</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>252.0</td>
<td>CER</td>
<td>PARC 'FDP'</td>
<td>04</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>253.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>05</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>254.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>09</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>255.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>256.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>257.0</td>
<td>CER</td>
<td>&quot;LOCAL&quot;</td>
<td>200000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>258.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>259.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>260.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>261.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>262.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>263.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>264.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>265.0</td>
<td>CER</td>
<td>PARC</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>266.0</td>
<td>CER</td>
<td>PARC</td>
<td>1010</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>267.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>1090</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>268.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>1090</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>269.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>270.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>271.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>272.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>273.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>274.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>275.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>276.0</td>
<td>CER</td>
<td>PARC 'FDP'</td>
<td>04</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>277.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>05</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>278.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>09</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>279.0</td>
<td>CER</td>
<td>CALL 'XP0101'</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>280.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>281.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>282.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>283.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>284.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>285.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>286.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>287.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>288.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>289.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>290.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>291.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>292.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>293.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>294.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>295.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>296.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>297.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>298.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>299.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>300.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>301.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>302.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>303.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>304.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>305.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>306.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>307.0</td>
<td>CER</td>
<td>C* 1.1.11 Write:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>308.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>309.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>310.0</td>
<td>CER</td>
<td>WRITE 'OUTPUT'</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>311.0</td>
<td>CER</td>
<td>C* New Code:</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>312.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>313.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>314.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>315.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>316.0</td>
<td>CER</td>
<td>MOVEL 'AD01'</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>317.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>318.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>319.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>1010</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>320.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>1090</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>321.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>322.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>323.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>324.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>325.0</td>
<td>CER</td>
<td>C* Old Code:</td>
<td>000000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>326.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>327.0</td>
<td>CER</td>
<td>C*</td>
<td></td>
<td>10.11.92</td>
<td></td>
</tr>
<tr>
<td>328.0</td>
<td>CER</td>
<td>E-AD01</td>
<td>0000</td>
<td>10.11.92</td>
<td></td>
</tr>
</tbody>
</table>
### Commonly Used File Servers

The following is a list of commonly used file servers:

<table>
<thead>
<tr>
<th>File Server</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>X0005</td>
<td>User Defined Codes Server</td>
<td>Retrieve Only</td>
</tr>
<tr>
<td>X0006</td>
<td>Retrieve Cost Center Master</td>
<td>Retrieve &amp; Scrub</td>
</tr>
<tr>
<td>XF0006</td>
<td>Cost Center I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
<tr>
<td>X0010</td>
<td>Automatic Next Numbering</td>
<td>Retrieve &amp; Increment</td>
</tr>
<tr>
<td>X9203</td>
<td>DD Alpha Description</td>
<td>Retrieve Only</td>
</tr>
<tr>
<td>X9800E</td>
<td>Data Dictionary Info</td>
<td>Editing Info</td>
</tr>
<tr>
<td>XF0101</td>
<td>Address Book I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
<tr>
<td>XS0101LA</td>
<td>Address Book</td>
<td>Retrieve Only</td>
</tr>
<tr>
<td>X0901</td>
<td>Account Master</td>
<td>Retrieve &amp; Formats</td>
</tr>
<tr>
<td>XF0901</td>
<td>Account Master I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
<tr>
<td>X41LOCN</td>
<td>Location Format</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>File Server</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>X41LOT</td>
<td>Lot Number Assignment</td>
<td></td>
</tr>
<tr>
<td>X41DUP</td>
<td>Lot Master Duplicate</td>
<td>Edits</td>
</tr>
<tr>
<td>X4101</td>
<td>Item Master</td>
<td>Retrieve &amp; Edit</td>
</tr>
<tr>
<td>X4108</td>
<td>Lot Master Update</td>
<td>Creates &amp; Updates</td>
</tr>
<tr>
<td>X4111</td>
<td>Write to Item Ledger</td>
<td>Writes Only</td>
</tr>
<tr>
<td>XF4111</td>
<td>CARDEX I/O</td>
<td>Retrieve Only</td>
</tr>
<tr>
<td>XF42119</td>
<td>Sales History I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
<tr>
<td>XF42199</td>
<td>Sales Detail Ledger I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
<tr>
<td>XF43199</td>
<td>Purchasing History I/O</td>
<td>Add/ Change/ Delete</td>
</tr>
</tbody>
</table>
About Functional Servers

A functional server allows you to enhance the processing and maintenance of your application programs. Functional servers 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

This section contains the following:

- What Are Functional Servers?
- What Are the Advantages of Using a Functional Server?
- Setting Up Business Rules for an Entry Program
- How Does a Functional Server Function?

What Are Functional Servers?

A functional server is a program that performs all transaction validation and database updates.

The functional server removes from the application program the burden of performing edit and update operations. Rather, this functionality is placed within the server.

A functional server is a called program. The application program calling the server must tell the server what action to perform for every transaction. In turn, the functional server will record error flags and update flags and return error messages to the application program for use when determining the result of a call to a server.

Functional servers have the following naming convention:

XT (file name) (server version)

For example: The functional server for file F0411 would appear as XT0411Z1.
The following diagram depicts the flow of a typical program using a functional server:

```
[Diagram showing flow from Application Program to Functional Server to File Server to File]
```

**What Are the Advantages of Using a Functional Server?**

Advantages of using a functional server include:

- Minimized maintenance and versioning of your software.
- Ability to isolate data editing routines and file updates.
- Increased flexibility because multiple programs can use the same functional server.
- Smoother transition from an old database to a new database. Instead of modifying all programs, you will only have to apply a new set of functional servers.
- Ability to implement one functional server at a time without affecting your entire system.
What Are the Disadvantages of Using a Functional Server?

Disadvantages of functional servers include:

- A functional server is minutely slower because you are performing an external call to the server from your program.
- Functional server programs tend to be large.

Setting Up Business Rules for an Entry Program

To set up business rules for an 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.

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.

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 the Technical Foundation Guide.

How Does a Functional Server Function?

When a functional server is called, an entire transaction is processed.

- Generally, once a functional server is called, it receives the data that the user entered and loads it into a user space.
- It performs its functionality on the data.
- Finally, it returns the requested data back to the calling program using the user space. If any errors occur, the system loads them into a user index.

The system uses the following interfaces to communicate with the functional server:

- The call parameters
- The control fields within each user space line
- The error index
Functional Server Highlights

Functional servers are usefully for many things, highlights include:
- Provides all editing for a transaction
- Provides field default values
- Provides all database updates
- Performs inquiry for an entire transaction
- Runs interactively or in batch
- Supports a multitude of user interfaces

Basic Accounting Transactions

In the Financial System there are five basic transactions:
- Journal Entries
- A/P Voucher Entry
- A/P Checks
- A/R Invoice Entry
- A/R Cash Receipts

JD Edwards World uses one program for each part or transaction of the system.

Example: Voucher Processing Functional Server

The following graphic shows the programs that use the voucher processing functional server. JD Edwards World provides two demo versions of the functional server, ZJDE0001 and ZJDE0002.
Program Example - Traditional Architecture

Each program contains both the User Interface Logic and the Data Integrity Logic. You would access this one program to interface with the database.
User Interface Logic

Aspects of the user interface logic include:
- Screen format
- Field formatting
- Help functions
- Error message display
- Touch and feel

Data Integrity Logic

Aspects of the data integrity logic include:
- Field editing
- Multi-field editing
- Transaction editing
- Default logic
- Error message selection
- Tax processing
- Currency processing
- Database update
Example - Traditional Architecture - Alternative Method of Entry

If a user wanted the screen to look different, the User Interface Logic would have to change. The Data Integrity Logic would be duplicated.

Example - Traditional Architecture - Various Entry Methods

Several users each wanted their own User Integrity logic. The Data Integrity Logic remained the same and was duplicated numerous times.
Example - JD Edwards World Open Application Architecture - Various Entry Methods

The creation of a Functional Server allows you to maintain the Data Integrity Logic in one common program. The Functional Server becomes separated from each User Integrity Logic program. All of the User Integrity Logic programs access one Functional Server to interface with the database. This concept is called an Open Application Architecture.

Open Application Architecture

In the Open Application Architecture, the database is separated from each User Integrity Logic program by the Functional Server. Advantages of the Open Application Architecture include:

- Automatic consistency
- Reduced maintenance burden
- Stability of custom code
- Separation of development efforts
- Performance enhancements
Functional Server Interface

A functional server must handle two basic components:

- Data
- Error messages

Functional Server Transaction Data

Arithmetic:
1. Full transaction passed to server at one time.
2. A single transaction can have more than 1,000 lines.
3. Each line from 500 to 1,000 characters long.
   - A lot of space

Story Problem:
How can program A pass program B a one-thousand line transaction without using a 1-meg parameter?
**Functional Server Error Messages**

**Arithmetic:**
1. Each field can have an error.
2. Each line can have 150 or more fields.
3. Each transaction can have hundreds of lines.
   =A lot of space

**Story Problem:**
How can program A pass program B a one-thousand line transaction without using a 1-meg parameter?

**Answers**

#1. User Space
#2. User Index

**Functional Server Interface**

A Functional Server can interact with a User Space and a User Index by passing and receiving parameters.

**Functional Server Parameters**

Single data structure defined in / COPY module.
Parameters that are fixed and application specific.
- Fixed parameters
  - Action code (edit, update, inquire)
  - Number of lines in transactions
  - DREAM Writer version of Functional Server
- Application specific parameters
  - Contains header information for a transaction
  - Document number of transaction
  - Total amount of transaction
  - Batch number of transaction

**Functional Server User Space**

Single data structure defined in / COPY module.
- One big data area
- Maximum of 16 meg
- Beginning 100 bytes of user space reserved
- Data portion of user space contains formatted lines
  - User space lines defined by / COPY module
  - Each line contains three sections
    1) Control section
    2) Application specific section
    3) Record format section

Functional Server User Index

Single data structure defined in / COPY module.
- One big keyed data area
- Used to pass error messages back to application
- User index entry defined using a / COPY module
- Each user index entry contains two sections
  1) Key
     - Application ID
     - Line number (assigned by application program)
     - Data item in error
     - Error code
  2) Data - value of erroneous data

Functional Server /COPY Modules

Repository for all user space and user index formats.
- All User Space and User Index formats contained in / COPY modules
- All database record formats contained in / COPY modules
- / COPY module I00FS@@ contains generic data structures and constants
- Each Functional Server has its own I00FSxx / COPY module to define application specific data structures

Creating User Space and User Index

When you create user space and user index formats, use the following tools:
- OS/ 400 APIs
- X00991
  - Called once for each Functional Server the application program intends to use
  - Creates user space and user index for each Functional Server
  - Returns name and library where user space exists
Returns the length each user space line should be

**Accessing the User Space**

- Writing to the user space X98CHGUS.
- JD Edwards World version of QUSCHGUS API
- Updates a user space beginning at offset x for length
- Similar to CHGDTAARA command
- Reading from the user space QUSRVTUS.
- API
- Retrieves data from a user space beginning at offset x for length
- Similar to RTVDTAARA command
- Application responsibilities
  - Remember number of lines written to user space
  - Increment user space offset

**Accessing the User Index**

- User Index written to by Functional Server.
- Reading from the User Index.
  - C00RIX/COPY module reads the User Index
  - C00RIX returns formatted error message defined by /COPY module
  - First execution of C00RIX reads first entry in User Index
  - Subsequent executions of C00RIX do read nexts
    - Uses X00IDX under the covers
- Application responsibilities.
  - Remember the value of your Application ID (typically program name)
  - Set flag for initial read of User Index by C00RIX
  - Use the data item name and line number in error to set ON screen indicators

**Interactive Program Cycle Using a Functional Server**

- Mainline - no change
- S001 - no change
- S003
  - No change for add, change, or delete
  - Call Functional Server to perform an inquiry
- S004 - Retrieve records from User Space for display on screen
- S005
- Application program performs “scrubs” only
- Write data records to User Space
- Call Functional Server to perform edits
- Read each line from User Space to redisplay defaulted information
- Execute C00RIX to determine each data item in error so that screen error indicators may be set ON
- S010 - call Functional Server to perform an update

The Call Parameters for the Functional Server

The call parameters provide commands to the functional server which applies to all transaction lines in the input user space.

The first parameter is a Data Structure. The following fields are from the #PPARM Data Structure defined in the I00FS@@copy module.

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#PFUNC (1)</td>
<td>Specifies a function code. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>0 Edit and Update</td>
</tr>
<tr>
<td></td>
<td>1 Edit only</td>
</tr>
<tr>
<td></td>
<td>2 Update only</td>
</tr>
<tr>
<td></td>
<td>3 Inquire</td>
</tr>
<tr>
<td>#PVERS (10)</td>
<td>The DREAM Writer version number you are executing. This parameter uses the version number to retrieve processing options for the server. The default version number will be ZJDE0001. This allows global processing options to be set at the server level, instead of for each program.</td>
</tr>
<tr>
<td>(3 before A6)</td>
<td></td>
</tr>
<tr>
<td>#PSPCN (20)</td>
<td>The name of the user space which the program has used. The user space contains the modified database records. Characters 1-10 contain the space name, and characters 11-20 contain the library name.</td>
</tr>
<tr>
<td>#PSPCB (9,0)</td>
<td>The byte position within the user space where the application data begins. Characters in the space prior to this position contain header information used by the functional server.</td>
</tr>
<tr>
<td>#PNBRL (5,0)</td>
<td>The number of lines in the input user space which the application program has loaded. When inquiring, this contains the number of lines output to the user space.</td>
</tr>
<tr>
<td>PARM (Length)</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>#PWARN (1)</td>
<td>This parameter contains a code explaining how you want warnings to be handled. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>0 Normal warning processing</td>
</tr>
<tr>
<td></td>
<td>1 Treat warnings as errors</td>
</tr>
<tr>
<td></td>
<td>2 Ignore warnings</td>
</tr>
<tr>
<td>#PCYCL (1)</td>
<td>This parameter is only used if the #PWARN parameter specifies normal warning processing. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>0 No cycle, all cycle processing ignored</td>
</tr>
<tr>
<td></td>
<td>1 First cycle, all warning messages are sent to the program</td>
</tr>
<tr>
<td></td>
<td>2 Second cycle, only warning messages not previously sent are sent to the program</td>
</tr>
<tr>
<td>#PDFTC (1)</td>
<td>Specifies how you want field values to be defaulted. 0 will default field values for add lines only and 1 will default field values for change or add lines.</td>
</tr>
<tr>
<td>#PXATP (3)</td>
<td>The application specific transaction type.</td>
</tr>
<tr>
<td>#PLVL (1)</td>
<td>The transaction level. 0 implies that each detail record to be updated or added has been sent in the input user space. 1 applies only to changes or deletions because only one record is sent in the input user space and the server will change or delete all other records for that transaction.</td>
</tr>
<tr>
<td>#PPROG (10)</td>
<td>The name of the calling program. This is used by the server to update the program name field in the updated database records.</td>
</tr>
<tr>
<td>#PAPPL (10)</td>
<td>The application ID value used for writing entries to the error index. Generally, this may be the same value as the calling program.</td>
</tr>
<tr>
<td>#PFLDS (4,0)</td>
<td>The number of fields which have been loaded to the Field Names Array parameter.</td>
</tr>
<tr>
<td>#PFMT (10)</td>
<td>The record format identifier the application program has used. This is used for versioning, allowing the database to change without the need for recompiling the application program.</td>
</tr>
<tr>
<td>#PEDIT (1)</td>
<td>Indicates the overall result of edits performed against all transaction lines. 0 implies that the edits went OK, 1 means there were some warnings, 2 is errors occurred.</td>
</tr>
<tr>
<td>#PUPDT (5,0)</td>
<td>The number of database updates which occurred. This will allow the program to know whether any updates actually occurred.</td>
</tr>
<tr>
<td>#PERR (4)</td>
<td>Specifies any errors that occurred within the server. A non-blank value indicates a fatal error occurred.</td>
</tr>
<tr>
<td>PARM (Length)</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>#PFERR (4)</td>
<td>Contains the first error message found during editing.</td>
</tr>
<tr>
<td>#PFDTA (4)</td>
<td>Contains the data item of the first field which had an error during editing.</td>
</tr>
<tr>
<td>#P#MDE (1)</td>
<td>For currency translations, this contains the mode of entry. If this value is passed as blank, the server will output the default mode of entry.</td>
</tr>
<tr>
<td>#PCRCD (3)</td>
<td>For currency translations, this contains the currency code of entry. If this value is passed as blank, the server will output the default currency code.</td>
</tr>
<tr>
<td>#PCRR (15,7)</td>
<td>For currency translations, this contains the currency exchange rate of entry. If this value is passed as zero, the server will output the default currency rate.</td>
</tr>
<tr>
<td>#PIDXN (20)</td>
<td>The name of the user index which the functional server will use to return error messages to the program. Characters 1-10 contain the index name, and characters 11-20 contain the library name.</td>
</tr>
<tr>
<td>#SPCL (5,0)</td>
<td>The total length of each user space record. This includes both the user space control fields and the database record format.</td>
</tr>
<tr>
<td>#SPEC (250)</td>
<td>This is a data structure which is redefined by each server. Generally, this will contain the application key fields which a specific server uses.</td>
</tr>
</tbody>
</table>

The second parameter will be an array.

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variable Array</td>
<td>An array of field names which the program has used. Only fields in this array will be updated in the database. If the first element contains &quot;ALL&quot;, then all fields will be used. The number of field names parameter should contain the number of entries loaded into this array.</td>
</tr>
</tbody>
</table>
## Control Fields within the User Space

The input user space can contain multiple lines for each control field. The following fields are defined in the #SSPCR Data Structure in the I00FS@@ copy module.

<table>
<thead>
<tr>
<th>PARM (Length)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SPCAC (1)</td>
<td>The line action code. The valid values are:</td>
</tr>
<tr>
<td></td>
<td>A  Add the record</td>
</tr>
<tr>
<td></td>
<td>D  Delete the record</td>
</tr>
<tr>
<td></td>
<td>C  Change the record</td>
</tr>
<tr>
<td></td>
<td>U  Change the record if it already exists, otherwise add the record</td>
</tr>
<tr>
<td></td>
<td>V  Void the record</td>
</tr>
<tr>
<td>#SPCID (15,0)</td>
<td>Used by the program to uniquely identify each line in the user space. (optional)</td>
</tr>
<tr>
<td>#SPCER (1)</td>
<td>The line error code.</td>
</tr>
<tr>
<td></td>
<td>X  the line is OK</td>
</tr>
<tr>
<td></td>
<td>1  some warnings</td>
</tr>
<tr>
<td></td>
<td>2  errors.</td>
</tr>
<tr>
<td>#SPCUP (1)</td>
<td>The line update code.</td>
</tr>
<tr>
<td></td>
<td>0  the line was not updated</td>
</tr>
<tr>
<td></td>
<td>1  updated</td>
</tr>
<tr>
<td>#SPCRR (9,0)</td>
<td>Contains the database relative record number which corresponds to this user space record. For adds, this is only loaded following an update operation. For changes and deletes, this is updated following an edit operation.</td>
</tr>
<tr>
<td>#SPCMN (2,0)</td>
<td>Contains the database physical file member number which corresponds to this user space record. For adds, this is only loaded following an update operation. For changes and deletes, this is updated following an edit operation.</td>
</tr>
<tr>
<td>#SPCPG (12)</td>
<td>Allows the program to store up to 12 bytes of information with each user space record.</td>
</tr>
<tr>
<td>#SPCAP (200)</td>
<td>Any application specific information which must be passed to the server for each transaction line, but is not contained within the transaction record format.</td>
</tr>
</tbody>
</table>

**Application fields**

Externally described record format for the transaction record.
Error Message Index Line (C00RIX)

The output error message index contains warning and error messages issued for each line in the user space. The following fields are from the #IIDXR Data Structure defined in the I00FS@@ copy module. The structure of the message index line is as follows:

<table>
<thead>
<tr>
<th>Field (Length)</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>#IDXAP (10)</td>
<td>The application identifier from the input parameter. Allows a program to access only its error messages.</td>
</tr>
<tr>
<td>#IDXID (15,0)</td>
<td>The line identifier from the input user space.</td>
</tr>
<tr>
<td>#IDXFN (10)</td>
<td>The data item portion of the field name.</td>
</tr>
<tr>
<td>#DXER (4)</td>
<td>Contains the data dictionary error message code.</td>
</tr>
<tr>
<td>#DXWN (1)</td>
<td>Line warning code 1 = Warning, 2 = Error.</td>
</tr>
<tr>
<td>#DXMD (87)</td>
<td>Not used.</td>
</tr>
</tbody>
</table>

Example - Functional Server Program Sections

```
COPY /COPY_JEDCPY,0001FSRV

CLEMCO
MOVELEN000
CALL 'XS0410',1X, 81
RDEM,Pro,DERO60

*IN01
P02
RDEM,Pro,DERO60,01
S00,IN
ACO03
ACP03
ACS0
AC03
SNREF
SNEP
```
26BFunctio nal Servers

MOVU 'ACSM' @SPMT
MOVU 'REVO1' @EKLOT
MOVU 'CHAIN' @CPPER
MOVU 'Y' @LOCXR
Z-ADD1 @SNUM
CALL 'XF0101'

---- ---- ----
PANN PSL01
PANN 10101

*IS81 IFPQ 'o'
ADDO @KCT @ADDCT
MOVEL 'ACSM' @SMTT
MOVEL 'REVO1' @EKLOT
MOVEL 'UDEFT' @SMTT
CALL 'XF0101'

---- ---- ----
PANN PSL01
PANN 10101

Load AR Specific Parameters

MOVU @KLOT @ARCT
MOVU DMTDOC @ARCO
Z-ADDC @LDCD @ARDC
Z-ADDC @LDCU @ARIC
MOVE @LCT @ARCT
MOVE *BLANK @ARFPL

Load functional server para for edit/update.

MOVU LAZ32N @PN01
MOVU JA232AP @APPC
MOVU @HDTP @HDTPC
MOVU @SEPR @SEPRC
D-ADDR @VOLRP @VOLRB
MOVE @NOHR @NOHRB
MOVE @DONE @DOPC
MOVE @PROG @PROGC
MOVEL @INT @INTC
MOVEL @SPMN @SPMNC
Z-ADDR1 @VOLRP @VOLRB
Z-ADDR2 @VOLRP @VOLRB
MOVE @BLANK @PRMT
MOVE @HLA1 @PRMT

CALL 'XT0311Z1'

---- ---- ----
PANN @PARN
PANN @ART

Call to file server XF0101 to retrieve record

Call to file server XF0101 to update record

Call functional server XT0311Z1
Write records to user space for functional server.
Load G/L Functional Server Specific Parameters

MOVE '1.EREO' $LOCUC One-to-One Rel
MOVE '8.EREO' $LOCUC Document Type
MOVE '3.EREO' $LOCUC Document CD
Z-ADDGEO $LOCUC G/L Entry
Z-ADDGEO# $LOCUC G/L Entry
$SFICU

Z-ADDJEO $LOCUC Batch Number
Z-ADDPG $LOCUC Batch Number
Z-DTIP
MOVE 'Y' $LOCUC Batch Type
MOVE '2000' $LOCUC Company
MOVE '4000' $LOCUC Load's Model
MOVE '4000' $LOCUC Change's Model
MOVE '4000' $LOCUC Redeem's Model
MOVE '4000' $LOCUC A/R Spec Name
MOVE '4000' $LOCUC A/R Spec Part
MOVE '4000' $LOCUC A/R Spec Lengh
MOVE '4000' $LOCUC A/R Spec

Call functional server - XTO911Z1 - Edit and Update

Load functional server parms for edit and update

MOVE 'Z.S402' $F402C Action Code
MOVE 'Z.S402' $F402C Data version
MOVE 'Z.S402' $F402C Space name
Z-ADDGR $F402C Space offset
Z-ADDG $F402C Number of lines
Z-ADDI $F402C Waring Handler
Z-ADD $F402C Exchange Rate
MOVE 'Z.S402' $F402C Cycle, nes
MOVE 'Z.S402' $F402C Default on neg
MOVE 'Z.S402' $F402C Type
MOVE 'Z.S402' $F402C Detail level
MOVE 'Z.S402' $F402C Program name
Z-ADDG $F402C Number of field
Z-ADDI $F402C Number of field
MOVE 'Z.S402' $F402C Form name
MOVE 'Z.S402' $F402C Variable name
MOVE 'Z.S402' $F402C Currency code
MOVE 'Z.S402' $F402C Exchange Rate
MOVE 'Z.S402' $F402C Index name
Z-ADDG $F402C Space length
MOVE 'Z.S402' $F402C Application par

CALL 'XTO911Z1' 'S1'

10

User space description

MOVE 'USER' $USERCD

Current user space offset

S-ADDG4$USERCD $USERCD

Read record from user space

CALL 'READ' $USERCD 01
Available Functional Servers

<table>
<thead>
<tr>
<th>Functional Server</th>
<th>Description</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>XT0101Z1</td>
<td>Address Book</td>
<td></td>
</tr>
<tr>
<td>XT0311Z1</td>
<td>Accounts Receivable</td>
<td></td>
</tr>
<tr>
<td>XT0311Z1E</td>
<td>Accounts Receivable</td>
<td>User Exit</td>
</tr>
<tr>
<td>XT0411Z1</td>
<td>Accounts Payable</td>
<td></td>
</tr>
<tr>
<td>XT0411Z1E</td>
<td>Accounts Payable</td>
<td>User Exit</td>
</tr>
<tr>
<td>XT06116Z1</td>
<td>Payroll Time Entry</td>
<td></td>
</tr>
<tr>
<td>XT0901Z1</td>
<td>Account Master</td>
<td></td>
</tr>
<tr>
<td>XT0911Z1</td>
<td>Journal Entry</td>
<td></td>
</tr>
<tr>
<td>XT0911Z1E</td>
<td>Journal Entry</td>
<td>User Exit</td>
</tr>
<tr>
<td>XT4102Z1</td>
<td>Item Balance</td>
<td></td>
</tr>
</tbody>
</table>
About Source Debugger

There are two types of programs that can be executed under the JD Edwards World Source Debugger - interactive and batch. The only difference when running the Source Debugger on an interactive program compared to a batch program is the initial execution statements. Once the Source Debugger has begun, all of the features are the same for both interactive and batch programs.

The JD Edwards World Source Debugger is a tool designed to help you determine where an error exists in your program. You can apply the Source Debugger to any program, whether it is in production or development. Since the Source Debugger displays source code, you must have the source code installed on your machine.

The source code you see while running the Source Debugger is displayed in SEU Browse mode, so you cannot change a line within the program. However, you may display or change the value of any field, variable, or indicator within the program. In addition, you can add or remove a breakpoint anywhere in the program.

This section contains the following:

- Using Debugger with an Interactive Program
- Using Debugger with a Batch Program

Before You Begin

- The JD Edwards debug tool is based on IBM debug objects. You must have authority to the IBM commands STRDBG, ADDPGM, ADDBKPS at a minimum. IBM has a STRISDB command which interacts with the program source. As of V5R1, the STRDBG command allows use of F10 to step through the program statements once the initial breakpoint is set.

  **Note:** In order for the program to be debugged to be visible, it must have been compiled on the current machine.

Using Debugger with an Interactive Program

The program can exist in your production environment, your development environment, or both. To use Debugger complete the following tasks:

- Determine the program environment
- Initiate the JD Edwards World Source Debugger
- Execute the program being debugged
To determine the program environment

1. From the Computer Assisted Design menu (G92), select Software Versions Repository.

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Name</td>
<td>Type your program name.</td>
</tr>
</tbody>
</table>

Note: If the program exists in several environments (production and development), you must determine against which program environment to run the Source Debugger. While in the debug environment, use WRKOBJ to see which library the program is attached to.

To initiate the JD Edwards World Source Debugger

1. Type the JD Edwards World debug command (JDEDBG) and press F4.

Field Description

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Name</td>
<td>Type your program name.</td>
</tr>
</tbody>
</table>
Source File
Type the file name that contains the source code of your program. If *OBJECT is entered, the source file name and library will be taken from the program object. If *SPLF is entered, the compile listing will be used for the source statements. This will include the statements included from COPY members which allows them to be debugged as well.

Library
Type the name of the library that contains the source file. Generally, this is JDFSRC for your production environment or DEVSRC for your development environment.

2. Enter the correct values in the proper fields and press Enter to start the Source Debugger.

- Any time the program being debugged is executed, the source code will display in debug mode, until you end the Source Debugger. (ENDDBG in all cases.)

To execute the program being debugged

If it is an interactive program, you can either call the program from a command line or select the menu option that will execute the program.

- Call program name and include the required parameters
- Selection/Menu

After you have executed the program, the first thing you will see is the program source code.

Note: The source code is displayed in browse mode, so you cannot edit or change any code.
Using Debugger with a Batch Program

To use Debugger with a batch program you should complete the following tasks:

- Sign on to the program environment
- Initiate the JD Edwards World Source Debugger
- Execute the program
- Set the break point
- Continue execution

To initiate the JD Edwards World Source Debugger

This step is similar to debugging an interactive program. The difference is that you must enter the debug command twice.

The first time you initiate JD Edwards World Source Debugger (JDEDBG - F4), the Program Name will be the CL Program.

1. Enter the correct values in the proper fields on the Debug Program form and press Enter.

2. Enter the JD Edwards World Source Debugger command (JDEDBG - F4) again, but this time change the Program Name to the RPG Program Name.

   The reason for this is you cannot run the Source Debugger on a program that is submitted and executed in a subsystem. You must “trick” the Source Debugger into thinking that your batch program is actually an interactive program.
To execute the program

Since you are executing a batch program interactively, you must call the CL Program from a command line.

1. **CALL CL program** (The parameters are usually “program name” and “version”).

2. The CL Program source code displays.

```plaintext
$3701 Scan:  JDE Visual Debug  JDBSRC  JDSRC
     Current Breakpoint:  J7001
0001.00 /***************PROGRAM 098401***************
0002.00 / Program .... J98401
0004.00 / Description .... Inventory by Cost Center
0005.00 / Program Revision Log
0006.00 / Date  Programmer  Description
0010.00 / 11/10/99  BB0501300  SAR # 09365896
0012.00 / 11/10/99  BB0501300  SAR # 09365896
0014.00 /***************Define program file(s) and variable(s).***************
0016.00 /
0017.00 /  F3=File Command Line  F4=ADDSEP  F5=ADDEXP w/prompt  F7=DISPVAR
            F8=MODVAR  F13=Display Indicators  F16/15Scan Fwd/Rev  F14=More
```

**Note:** The source code is displayed in browse mode, so you cannot edit or change any code.

To set the break point

Set a break point on the line testing the job type in order to change a variable in the CL. The variable &JOBTYPE normally edits against a batch program being executed by calling it from a command line.

1. Find the line of code that contains the variable &JOBTYPE.
2. Press F5 anywhere on the line containing &JOBTYPE to set the breakpoint.
   - The line is highlighted, indicating that a breakpoint has been set on that line.

To continue execution

1. Allow your program to continue executing. Press F3 to continue to a breakpoint.
   - The line on which you set the breakpoint will display in reverse image. This indicates that the program has reached this point in the CL program and is ready to execute this line.
   - You must change the value of the &JOBTYPE variable to something other than 1 so that the compare to '1' fails.

2. To change the value of &JOBTYPE, press F8 to access the Change Program Variable form.

<table>
<thead>
<tr>
<th>Change Program Variable (CHUPGMVAR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type choices, press Enter.</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Program variables:</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Program variable . . . . . . .</td>
</tr>
<tr>
<td>' &amp;JOBTYPE '</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>- Earing pointer variable</td>
</tr>
<tr>
<td>-</td>
</tr>
<tr>
<td>+ for more values</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>New value . . . . . . . . .</td>
</tr>
<tr>
<td>' 2 '</td>
</tr>
<tr>
<td>Program . . . . . . . . . .</td>
</tr>
<tr>
<td>7928401</td>
</tr>
<tr>
<td>Name, *OPTION</td>
</tr>
</tbody>
</table>

3. Complete the Change Program Variable form and press enter.
   - The value of &JOBTYPE is now changed to your specified value.

4. Press F3 to allow the CL program to continue processing.
   - The RPG program source is displayed next.
Features of the JD Edwards World Source Debugger

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F2</td>
<td>To display a JD Edwards World command line window, press F2.</td>
</tr>
<tr>
<td>F3</td>
<td>Once the program hits a breakpoint or when you first enter the source, F3 will allow the program to continue processing.</td>
</tr>
<tr>
<td>F5</td>
<td>Position the cursor on an executable line and press F5 to add a breakpoint. You cannot add breakpoints to a comment line, only to executable lines. Once the breakpoint is set, the line will be highlighted. If the program executes a line with a breakpoint set on it, the line will appear in reverse image and the program will pause before executing the line.</td>
</tr>
</tbody>
</table>

Example F5

```
03701 Shield
Current Breakpoint: J0001
0319.00 C*
0320.00 C* $AUTO CB81 '1' $003 24
0321.00 C*
0322.00 C* RUN
0323.00 C*
0324.00 C* Begin normal program processing.
0325.00 C* -----------------------------
0326.00 C* 
0327.00 C* *INLR DNEQ 'O'
0328.00 C*  
0329.00 C* If subroutine page display not set, set subroutine page display.
0330.00 C* 
0331.00 C* $FPR0  IPLE 6
0332.00 C* LARG1 $FPR0
0333.00 C* RND
0334.00 C* 
0335.00 C* Write video screen.
0336.00 C* *DMPR0 Determines which format is written
F2-JDE Command Line PC-ADDBK PK-ADDBK w/prepent Ft-ADSPMONAR
F8-ADSPMONAR F10-display indicators F11/15-Icon Pwd/Btn F24-NextPage
```

Example F6

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F6</td>
<td>Position the cursor on an executable line and press F6 to add a breakpoint with a prompt. You cannot add breakpoints to a comment line, only to executable lines. Once the breakpoint is set, the line will highlight. If the program executes a line with a breakpoint set on it, the line will reverse image and the program will pause before executing the line.</td>
</tr>
</tbody>
</table>

Example F6
Use the prompt, after pressing F10, to assign a skip value or breakpoint conditions.

### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F7</strong></td>
<td>Position the cursor on an executable line and press F7 to display the values of all of the variables on that line. Breakpoints within copy modules will stop at the correct source sequence number.</td>
</tr>
</tbody>
</table>

#### Example F7

Display Program Variables

<table>
<thead>
<tr>
<th>Program</th>
<th>P01051</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delimiter level</td>
<td>1</td>
</tr>
<tr>
<td>Start position</td>
<td>1</td>
</tr>
<tr>
<td>Format</td>
<td>*DCL</td>
</tr>
<tr>
<td>Variable</td>
<td>*IN99</td>
</tr>
<tr>
<td>Type</td>
<td>CHARACTER</td>
</tr>
<tr>
<td>Length</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>'1'...'+1...'...'+2...'...'+3...'...'+4...'...'+5'</td>
</tr>
</tbody>
</table>

| Variable         | *IN99  |
| Type             | CHARACTER |
| Length           | 1      |
|                   | '1'...'+1...'...'+2...'...'+3...'...'+4...'...'+5' |

Press Enter to continue.

P0: Exit  F12: Cancel

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>F8</strong></td>
<td>To change the value of a variable, press F8 and type the correct values in the prompt screen.</td>
</tr>
</tbody>
</table>

#### Example F8

| Variable         | *IN99  |
| Type             | CHARACTER |
| Length           | 1      |
|                   | '1'...'+1...'...'+2...'...'+3...'...'+4...'...'+5' |
Function | Description
--- | ---
F10 | Move Line to Top of Page.
F12 | From anywhere on the screen, press F12 to remove the current breakpoint. The line is no longer highlighted, indicating the line is no longer set as a breakpoint. The program will immediately continue processing.
F13 | To display the current values of all indicators, press F13.

Example F13

Display Program Variables

<table>
<thead>
<tr>
<th>Program</th>
<th>Recursion level</th>
<th>Start position</th>
<th>Format</th>
<th>Length</th>
<th>Variable</th>
<th>Type</th>
<th>Lower/upper bounds</th>
<th>Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>P01051</td>
<td>1</td>
<td>1</td>
<td>*CHAR</td>
<td>1</td>
<td>*INH</td>
<td></td>
<td>[3,99]</td>
<td>1</td>
</tr>
</tbody>
</table>

Press Enter to continue.
**Function** | **Description**
--- | ---
F15 | Type in a value on the Scan Line at the top of the screen and press F15 to scan backward from the point you are at to the end of the source code. If a match is found, the line containing the matching value will be displayed. To continue scanning backward, press F15 again.
F16 | Type in a value on the Scan Line at the top of the screen and press F16 to scan forward from the point you are at to the beginning of the source code. If a match is found, the line containing the matching value will be displayed. To continue scanning forward, press F16 again.
F21 | To display a command line, press F21.

**ENDDBG  End Debug**

To stop the JD Edwards World Source Debugger, enter ENDDBG from a command line. You can not enter ENDDBG while displaying the source code of a program in debug. This command will end debug mode for all programs in the Debugger at that point.

**Note:** You can remove a single program from debug mode by using the RMVPGM (remove program) command.
Software Scan and Replace

About Software Scan and Replace

The Software Scan and Replace feature lets you scan source members to accomplish the following:

- Scan for a particular item and replace it with a new item
- Produce a list of all members that meet the search criteria
- Scan for a particular item and insert a source file after each occurrence

Because you can potentially replace source code across all systems, this job is submitted to batch and held in the job queue until you release it.

This section contains the following:

- To Work with Software Scan and Replace
- Report
- Guidelines

To Work with Software Scan and Replace

1. From the Computer Assisted Programming menu (G93), choose Developer’s Workbench. From the Developer’s Workbench menu (G9362), choose Software Scan and Replace.

   98810 | Software Scan & Replace
   | System code . . . . 55  | (Blank = all)
   | Function code . . . ERP  | (Blank = all)
   | Specific object . . . . (Generic = *)  | (Blank = all)
   | File ID . . . . KB5tRSC  | (Blank = all)
   | Source library . EP5tRSC  | (Defaults to source lib in member master)
   | Scan argument: I00SRC  | (If search argument contains imbedded blanks enclose argument with .)
   | Replacement argument: I00SRC  | (If argument contains imbedded blanks enclose argument with .)

   Column replacement: Beginning column - ___  Ending column - ___

   Replacement Overflow Code . . . .

   Insert Source From: File . . . . Libr . . . . Member . . . .

   The previous screen illustrates how you replace the copy module I00SC with the copy module I00RSC for all RPG members coded to install system code 55.

2. Complete the form and press Enter.
The job submits to batch and a message displays. The job is held on the job queue.

3. When you are ready to process the job, go to the Work with Submitted Jobs form (hidden selection 33) and release the job.

Report

When the job completes, it produces a report that indicates those objects where the scan and replace occurred.

Guidelines

If you leave the Replacement argument field blank, the utility produces a listing of all source members that meet the search criteria.

Because this job could be used to update all code across systems and could severely impact processing, it is automatically held on the job queue.

Use this job to replace a copy module across systems or determine a listing of members that meet certain criteria. Use with caution.
Performance Issues

General Performance Issues

Following are some performance issues you should consider when executing JD Edwards World software, changing current JD Edwards World programs, or writing new programs:

- Purge your files on a regular basis to avoid excess, unnecessary records existing in files. The REUSEDLT *YES parameter on a Physical file can be used on files where records are deleted to minimize file expansion and contraction. G9645/14 will present a list of P98999 versions which can be used to RGZPFM the named files. The RGZPFM command, in general, removes deleted records and rewrites the remaining records in a file.
- Minimize the number of open files in a program. If a file may not be used, define it as a User Controlled Open file.
- Use User Spaces and User Indexes wherever possible.
- Use File Servers and Functional Servers wherever possible.
- Minimize the number of subroutine calls within your program.
- Weigh the advantages of inter-program calls. Although this method is very modular in design, you should consider the effect on performance.
- Substitute the comparison of a literal with the comparison of a variable. For example: Use *ON and *OFF to set an indicator on and off rather than a 1 and 0.
- Consider flexibility versus performance when using User Defined Codes, Vocabulary Overrides, and loading Data Dictionary values extensively.

DREAM Writer

One definition of “Performance Problem” is that some Dream Writers take a long time. The usual symptom is that the user does a Work with Submitted Jobs and sees the job spending a long time “indexing”. Programs which have a very large based-on file are the prime suspects. P09800 is not a suspect!

The problem in these cases often involves the IBM Query File Optimizer. Simply stated, the optimizer tries to find an existing access path (logical file) that it can use to select records. If it can’t find one or finds one that will return more than 20% of the file, it builds a whole new access path which takes a long time for large files.
Possible solutions:

Solution 1

The first assumption is that the applicable IBM PTFs have been applied.

1. On the DW Additional Parameters screen, there is a File Output Type field. An F1 on this field shows that values 2 and 4 use a logical file. If an existing logical should be used, enter a 2 in the File Output Type and the file name in the Override Logical File field. This setup often makes the DW run faster since the IBM Query File Optimizer is bypassed by using an LF rather than the OPNQRYF.

2. To have DW create the Logical File for you, enter a 4 in the File Output Type and blanks in the Override Logical file field. This option will use the DW Data Selection and Sequencing parameters to create a Logical File over the based-on physical, and will name it by replacing the P in the program name with an F and adding a 00n suffix.

3. IMPORTANT: You must then run the DW so the LF will be created. Then, before going back into the DW with a 2 to change, (which will cause the LF to be deleted) you must find the file using a DSPDBR on the based-on file. Do RNMOBJ so DW will not find this new file. Then, go back to the Versions list and enter a 2 to change the DW version. DW will automatically have changed the File Output Type to a 2 and have entered the name of the logical file it created. Enter the file name from the above RNMOBJ command.

4. When the File Output Type is a 2, the options for Data Selection Values and Data Sequencing Values will not show up on the change window. This is because the values in the LF will be used. If the values need to be changed, go to Step 3C.

5. On the DW version list, if you use option 7, you will see either the OPNQRYF statement or the DDS for the logical file.

Solution 2

The based-on file can be changed to a logical file attached to the Physical file, but then the File Output Type must be set to 1. This reintroduces the OPNQRYF command into the process using the Query File Optimizer. The Data Selection and Sequencing options are now available.
7 Group Jobs
Overview to Group Jobs

About Group Jobs

Group Jobs is an IBM concept which allows the user to switch between sessions on the Series i. This involves communication with the Series i. Most users now use IBM Client Access which can have several sessions active. Switching between these sessions does not require Series i communication. The following is a description of JD Edwards's implementation of IBM Group Jobs.

The Group Jobs form allows you to perform a number of tasks from a single form, saving you both time and effort. You can perform the following functions from this form:

- Run up to 16 jobs under a single sign on
- Execute (or run) CL and fast path commands from a single command line
- Execute (or run) JD Edwards World Hidden Selections

In addition to the added convenience, the Group Jobs function keeps the files for each of the jobs selected opened, whether they are currently active or not.

This section covers the following tasks:

- Access the JD Edwards World Group Job Form
- Create New Group Jobs
- Activate Suspended Group Jobs
- Terminate Job Groups
- Change to Non-Group Mode
- Sign Off with Suspended Group Jobs
Access the JD Edwards World Group Job window

About the JD Edwards World Group Job window

This section contains the following:

- Accessing the JD Edwards World Group Job window
- Creating New Group Jobs
- Activating Suspended Group Jobs
- Terminating Group Jobs
- Changing to Non-Group Mode
- Signing Off with Suspended Group Jobs
- Work with Non-JD Edwards World Group Jobs
- Advanced Functions of the JD Edwards World Group Job window

Before You Begin

For a user to access the JD Edwards World Group Job window at any time, the Esc (ATTN) key program should be set to call the JD Edwards World Group Job window program (P98GRP).

To set the ATTN key program

1. From the Security Officer Menu (G94), choose User Information.

```plaintext
<table>
<thead>
<tr>
<th>User ID</th>
<th>User Information</th>
<th>Action Code</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TEACH</td>
<td>I</td>
</tr>
<tr>
<td>Library List</td>
<td>CPMP JGPG COMMON PROCDATA JEDSEC GDPL</td>
<td></td>
</tr>
</tbody>
</table>

User Security:
- A J E D P E Allow Command Entry (Y/N): Y
- Initial Menu to Execute: A
- Initial Program to Execute: Allow Fast Path (Y/N): Y
- Menu Level: [Space]
- User Type: TRACER
- User Class/Group: [Space]
- Batch Job Queue: CRATCH
- Job Scheduling Priority: [Space]
- Logging (level/severity/messages): [Space]
- Output Queue: P4E
- Optional Printer File Library: [Space]
- Current Library: [Space]
- Employee Address Number (PPAT): [Space]
- Display/Lang Pref | P9-Library Inquiry | P21-Print Lib List | P24-More Keys
```
2. Enter the JD Edwards World Group Job window program ID (P98GRP) in the Set Attention Program field.

**Accessing the JD Edwards World Group Job window**

After the Attention Key program has been set up in the JD Edwards World environment, you can access the Group Job window.

**To access the JD Edwards World Group Job window**

1. Sign off and sign back on to reset the Attention key program within the JD Edwards World Menu Driver.
2. Press the Esc (ATTN) key and the following is displayed.
3. Whenever the Group Job window is displayed, an entry can be made in the Description field <Enter> to describe the session associated with that job.

```
G0 J.D. Edwards & Company
General Business Systems

... GENERAL BUSINESS SYSTEMS
2. Address Book
3. General Accounting
4. Accounts Payable
5. Accounts Receivable
6. Financial Reporting
7. Modeling & Allocation
8. Fixed Asset
9. Payroll
10. Human Resources
11. Electronic Mail

Selection or command
***
```

**Creating New Group Jobs**

**To create new group jobs**

1. Press F5 on the Group Jobs window for New Job.
2. When the JD Edwards World Menu Driver is displayed, (J98INIT is the initial program) press the Esc (ATTN) key and the following is displayed.
Access the JD Edwards World Group Job window

The new group job GROUP02 is now in process. The group job GROUP01 was suspended when the function key F5 was pressed.

**Note:** If you are set up to access JD Edwards World software by J98INITA, your library list selection list will appear when F5 is used. Select an environment and then you will be able to display the JD Edwards World Group Job window.

Activating Suspended Group Jobs

To activate suspended group jobs

Press the Esc (ATTN) key to display the JD Edwards World Group Job window and enter option 4 next the job you want to activate.

- All suspended group jobs are displayed in the window.
- Any suspended group job can be activated, as illustrated below.
Terminating Group Jobs

Any group job, active or suspended, may be terminated from the JD Edwards World Group Job window.

To terminate group jobs

Enter option 9 next to the group job you want to terminate.

---

Changing to Non-Group Mode

To change to non-group mode

Enter option 9 beside all active and suspended group jobs.

---

Signing Off with Suspended Group Jobs

You can use two different methods to sign off with suspended group jobs.
To sign off with suspended group jobs

Select one of the following methods:

1. Press F18 within the JD Edwards World Group Job window.
2. Enter SIGNOFF, 90, or “...” on the JD Edwards World Group Job Cmd/HS command line.

**Note:** Because group jobs are created under one sign on, all group jobs are terminated when the signoff command is executed.

Work with Non-JD Edwards World Group Jobs

To work with non-JD Edwards World group jobs

To create group jobs that call a program outside the JD Edwards World software, the JD Edwards World Group Job window allows an external program to be executed. In addition, the Esc (ATTN) key can be pressed within the external program and still allow access to the JD Edwards World group jobs.

1. To call an external program, press F11 within the JD Edwards World Group Job window.

The following illustrates what will be displayed when F11 is pressed.

```
Change Library List (CHGLIBL)
Type choices, press Enter.
Libraries for current job . . . . . > QTEMP Name, *SAME, *NONE
   > TCAG200B
   > QP70R
   > TCA3020TA
   > A1090900
   > TRUHAMA
   > TCA0204R
   > SAPP
   > VFAAYAULIB
   > VFDAYAULIB
   > VCTAYAULIB
   > VCADLIB
   > VR6TS925

   + for more values
   *SAME, Name, *CRTDPT

Current library . . . . . . . . *SAME, Name, *CRTDPT
Bottom
F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

2. Complete the Change Library List screen.

You can add libraries related to the external program. Libraries currently in the library list can be removed if desired. However, the following libraries must be left in the library list to retain the link to the JD Edwards World group jobs:

- `QTEMP`
- `Library containing F9220 (JD Edwards World Vocabulary Overrides)`
- `Library containing F0082 (JD Edwards World Menu Master)`
- `Library containing F0092 (JD Edwards World User Information)`
Access the JD Edwards World Group Job window

- Library containing JD Edwards World Objects (For example, RPG, CL, DSPF)

After the CHGLIBL command has been executed, the CALL command prompt is displayed.

3. Enter the external program.

The following illustrates the CALL command prompt.

When the CALL command is executed, the external program will be executed.

- To work with a JD Edwards World group job, the Esc (ATTN) key can be pressed to display the JD Edwards World Group Job window.
- Any suspended group job can be activated from the JD Edwards World Group Job window.

Advanced Functions of the JD Edwards World Group Job window

**JD Edwards World Hidden Selections**

Hidden Selections are commands and features of the JD Edwards World products that are not available through a menu selection.

- Most JD Edwards World Hidden Selections (31+) can be executed from the command line at the bottom of the JD Edwards World Group Job window.
- The JD Edwards World Hidden Selection window (HS) can be used to display and execute hidden selections.
- JD Edwards World Hidden Selection (Menu) Security is used when users execute hidden selections.
- No JD Edwards World Menus or JD Edwards World Hidden Selections related to menus are allowed, (for example 27 or 29).
Entering Commands

You can enter any command on the command line at the bottom of the JD Edwards World Group Job window.

- Press F4 to prompt for a command.
- Place a “?” in front of a command to prompt.
- Press F9 to retrieve previous commands.
- Any parameters you enter while in prompt mode are not retrieved.
  - The last 10 previous commands are saved.
  - Only successfully executed commands are saved.
  - When you exit by pressing F3, previous commands are lost.
- JD Edwards World Fast Path Commands from User Defined Code 00/FP can be executed. F13 to display all Fast Path Commands.
  - To retain all commands entered and retrieve parameters entered in prompt mode, access the IBM Command Entry screen from the JD Edwards World Group Job window (For example, JD Edwards World Hidden Selection 36) and enter commands.
  - Commands can only be executed if there is a value of “Y” or “ ” in the Allow Command Entry (Y/ N) field defined in the JD Edwards World User Information option found on A94.

Securing the CHGJOB Command (HS33)

HS 33 uses the IBM command WRKSBMJOB. On this command screen, a user can access the CHGJOB command to move jobs to a different queue or change priorities.

Caution: Securing the CHGJOB command is problematic since JD Edwards World uses this command during the sign on process.

As of JD Edwards World release A81CU5 or A73CU15, the following workaround is available.

1. Objects J00WSJ, P00WSJ, V00WSJ and X00WSJ should be in the JD Edwards World object library.
2. Change the ZHIDDEN menu selection with SELECTION33 on it so the Option Key has J00WSJ rather than J00SBMJOB.
3. Sign out of the environment and back in. HS33 will present the information on a JD Edwards World menu which enables Function key and Option key security.

Note: Use Knowledge Document WST-00-0023 for IBM releases below V4R5.
Access the JD Edwards World Group Job window

JD Edwards World Group Job window Summary

The program allows you to:

- Create up to 16 jobs per sign on
- Execute commands, JD Edwards World hidden selections, JD Edwards World Fast Path Command, and JD Edwards World Fast Path Menu Execution

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>Exit the JD Edwards World Group Job window</td>
</tr>
<tr>
<td>F4</td>
<td>Prompt a command</td>
</tr>
<tr>
<td>F5</td>
<td>Create a new JD Edwards World group job</td>
</tr>
<tr>
<td>F6</td>
<td>Submit job to batch</td>
</tr>
<tr>
<td>F8</td>
<td>JD Edwards World Menu Word Search</td>
</tr>
<tr>
<td>F9</td>
<td>Retrieve previous command</td>
</tr>
<tr>
<td>F11</td>
<td>Create a new Non-JD Edwards World group job</td>
</tr>
<tr>
<td>F13</td>
<td>Display all fast path commands</td>
</tr>
<tr>
<td>F18</td>
<td>SIGNOFF all group jobs</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Selection Exits</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Activate a suspended group job</td>
</tr>
<tr>
<td>9</td>
<td>End a group job</td>
</tr>
</tbody>
</table>

The JD Edwards World Group Job window is not accessible when using:

- SysReq (Source Machine Only)
- A program that has reset the ATTN Key program (For example, OFFICE/400)
Work with the Attention MENU window

About the Attention MENU Window

The JD Edwards World Attention Menu window program is a generic program that allows you to access up to 15 predefined programs via the Esc (ATTN) Key. The 15 predefined programs are associated with options on a JD Edwards World Menu.

Note: Each user can be assigned a different JD Edwards World Menu

Before You Begin

To access the JD Edwards World Attention Menu window at any time, the Set Attention Program field on the JD Edwards User Information video should be set to a JD Edwards World Menu, for example *G92.

The following illustrates how the Esc (ATTN) key program is set in the JD Edwards World software. The User Information screen can be accessed from the Security Officers Menu (G94).

<table>
<thead>
<tr>
<th>0052</th>
<th>User Information</th>
<th>Action Code. . . . . . . 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>User ID. . . . . . . . . .</td>
<td>TEACH</td>
<td></td>
</tr>
<tr>
<td>Library list . . . . . . .</td>
<td>OTMP_EDP01 COMMON PRODDATA JDPSC_C0PL</td>
<td></td>
</tr>
<tr>
<td>User Security:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Key . . . . . . . . .</td>
<td>A</td>
<td>Y</td>
</tr>
<tr>
<td>Initial Menu to Execute . .</td>
<td>A</td>
<td>Y</td>
</tr>
<tr>
<td>Initial Program to Execute</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Menu Level. . . . . . . . .</td>
<td>TEACHER</td>
<td></td>
</tr>
<tr>
<td>User Type. . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>User Class/Group . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Batch Job Queue . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Job Scheduling Priority . .</td>
<td>S</td>
<td>Y</td>
</tr>
<tr>
<td>Logging(level):severity/messages</td>
<td>4</td>
<td>ENLIST</td>
</tr>
<tr>
<td>Output Queue . . . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optional Printer File Library . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Current Library . . . . . .</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Employee Address Number (PFAT)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Attention Program . . . .</td>
<td>4999</td>
<td></td>
</tr>
<tr>
<td>F6-Display/Eng Pref F6-Library Inquiry F21-Print Lib List F24-More Keys</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: An * (asterisk) must precede the menu name.

Accessing the JD Edwards World Attention Menu window

After the Esc (ATTN) key program has been set up for the JD Edwards World software you can access the JD Edwards World attention menu window.
To access the JD Edwards World attention menu window

1. Sign off and sign back on to reset the Esc (ATTN) key program within the JD Edwards World Menu Driver.
2. Press the Esc (ATTN) key and the menu options for the menu will be displayed as follows.

**Note:** Original Job refers to the current job that has been converted to a group job. The remaining jobs refer to the first 15 interactive programs on the menu which the user is authorized to.

**Summary of JD Edwards World Attention Menu window Functions**

The program allows you to:

- Access 15 predefined programs via the Esc (ATTN) Key

<table>
<thead>
<tr>
<th>Function Key</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F3</td>
<td>Exit the JD Edwards World Attention Menu window</td>
</tr>
<tr>
<td>F4</td>
<td>Prompt a command</td>
</tr>
<tr>
<td>F6</td>
<td>Submit a job to batch</td>
</tr>
<tr>
<td>F8</td>
<td>JD Edwards World Menu Word Search</td>
</tr>
<tr>
<td>F9</td>
<td>Retrieve previous command</td>
</tr>
<tr>
<td>F13</td>
<td>Display all fast path commands</td>
</tr>
<tr>
<td>F18</td>
<td>SIGNOFF all group jobs</td>
</tr>
<tr>
<td>Selection Exits</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>4</td>
<td>Activate a menu selection</td>
</tr>
<tr>
<td>9</td>
<td>End a group job (Ignored on a Menu window)</td>
</tr>
</tbody>
</table>

The JD Edwards World Attention Menu window is not accessible while using
- SysReq (Source Machine Only)
- a program that has reset the Esc (ATTN) Key program (for example OFFICE/400)
Work with IBM Pass-Through

About Working with IBM Pass-Through

Note: This section is predicated on Series i network setup enabling use of the IBM command STRPASTHR. The steps for that implementation will not be covered in this manual.

To create group jobs on remote locations and still retain a link to the group jobs created on the source machine, use IBM Pass-Through. Perform the following tasks:

- Setting Up Access to Remote Locations
- Using IBM Pass-Through with Group Jobs

Setting Up Access to Remote Locations

To set up access to remote locations

To set up access to remote locations, go to the DREAM Writer versions list for Form ID P98GRP5.

<table>
<thead>
<tr>
<th>Version</th>
<th>Description</th>
<th>User</th>
<th>Chg Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDDD0001</td>
<td>Denver A</td>
<td>DEMO</td>
<td>08/23/93</td>
</tr>
<tr>
<td>XDDD0002</td>
<td>Denver C</td>
<td>DEMO</td>
<td>08/23/93</td>
</tr>
<tr>
<td>XDDD0003</td>
<td>Denver D</td>
<td>DEMO</td>
<td>08/23/93</td>
</tr>
<tr>
<td>XDDD0004</td>
<td>Denver E</td>
<td>DEMO</td>
<td>08/23/93</td>
</tr>
<tr>
<td>XDDD0005</td>
<td>Denver I</td>
<td>DEMO</td>
<td>08/23/93</td>
</tr>
<tr>
<td>XDDD0006</td>
<td>Atlanta</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0007</td>
<td>Chicago</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0008</td>
<td>New York</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0009</td>
<td>Dallas</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0010</td>
<td>Houston</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0011</td>
<td>San Francisco</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
<tr>
<td>XDDD0012</td>
<td>Washington DC</td>
<td>DEMO</td>
<td>11/13/93</td>
</tr>
</tbody>
</table>

The processing options for each version provide setup parameters for the STRPASTHR command allowing access to a remote location. The following illustrates the processing options.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination Virtual Control Unit</td>
<td>This is the control unit that the user will connect to at the remote location.</td>
</tr>
<tr>
<td></td>
<td>The first available device on the control unit will be selected.</td>
</tr>
<tr>
<td>Destination Location</td>
<td>This is the APPN network name for the remote location.</td>
</tr>
<tr>
<td>(Used in AS/400 Environment)</td>
<td></td>
</tr>
<tr>
<td>APPC Device(s)</td>
<td>These are the APPC devices that identify the route to the remote location.</td>
</tr>
<tr>
<td>(Used in S/38 Environment)</td>
<td>• Only one intermediate node is supported.</td>
</tr>
</tbody>
</table>

### Using IBM Pass-Through with Group Jobs

#### To use IBM Pass-Through with Group Jobs

1. Use the JD Edwards World menu B98P to start an IBM Pass-Through session to a remote machine.
2. Use the JD Edwards World Menu Revisions (1/G901) to attach your user defined DREAM Writer Form ID P98GRP5 versions to a menu (B98P).
When an option is selected on the menu, the IBM Start Pass-Through command will be executed to the remote machine, and still retain a link to the source machine group jobs.

The mechanism used to attach remote locations to the JD Edwards World Group Job window on the source machine is a parameter on the STRPASTHR (Start Pass-Through) command. The following illustrates the link to the source machine.

The SRQ10PGM (SysReq 10) parameter allows a program to be called on the source machine from the remote location. By entering the JD Edwards World Group Job window program (P98GRP) in this parameter, the JD Edwards World Group Job window can be displayed on the remote location by pressing SysReq 10, NOT the ATTN Key. This allows access to all suspended group jobs on the source machine and other remote locations.
8 Universal File Converter
Overview to Universal File Converter

About Universal File Converter

JD Edwards World Universal File Converter maps data from one data file resident on the Series i to another file on the same Series i using crossover rules. A DREAM Writer version can be set up so that the conversion can be repeated many times for a standing process or just once for a conversion process.

JD Edwards World Universal File Converter uses Crossover rules to map data:
- From one file to another file
- From one file to multiple files
- From multiple files to a single file

Caution: Do not attempt to use UFC on a file that contains double byte data. The converter program may corrupt the integrity of the bracketing shift in and shift out characters that are automatically inserted by double byte terminals.

The UFC does a character at a time move of the data from the input file data location into the output file data location. For this reason it is not suited to large, repeated data transfers. It is suited to one time data file conversions or small scale frequent data import situations.
The Cross Over Rules file defines the association between two files and includes data field information.
Step 1
You specify From files and To files through DREAM Writer processing options. You can specify up to four To files. If you require multiple From files, specify a join logical as the From file in the DREAM Writer “based on” file. The system returns file field information and pre-loads the Cross-Over Rules file with field name, length, size, type and reference (data dictionary name). The system pre-loads information in the Cross-Over Rules file for all fields that have the same reference (data dictionary field name) as the From file.

Step 2
You must manually associate the fields that were not automatically loaded in the Cross-Over Rules file. If you need special calculations for a field, you can specify special processing key words in the Conversion Rule field. You can also add the calculations into an external program that can be called from the converter program. The external program needs several parameters that are sent and passed back to the converter program. These parameters are: data, error, From field name, To field name, and number of To file records. You must specify the external program in the Conversion Rule field in the Cross-Over Rules file.

Step 3
In this step you specify the form ID and the version you selected in the first step. The From and To files should be the same (or exact equivalent) as the files specified in Step 1. The converter program accesses the cross-over instructions for the “From/To” combination and loads the information to arrays. The system then processes the arrays for each field that has an association. Finally, the system transfers the value in the From file to the To file.
Special Processing

Special processing procedures are available to help you in the conversion of one field to another.

To execute any of the special processing procedures listed below, you must type the appropriate keyword into the From or To Conversion Rule field. This is explained in Detail Cross Over Rules, later in this guide. There are special keywords for the following:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dates</td>
<td>The converter uses a keyword to decide what date translation is necessary.</td>
</tr>
<tr>
<td>Numeric Fields</td>
<td>The converter translates non-packed numeric data to packed data or vice versa, depending on your need. It also maintains decimal alignment, performing rounding or zero padding if required. Alphanumeric representations of numeric fields can be translated to numeric fields. Numeric fields can be translated into alphanumeric fields.</td>
</tr>
<tr>
<td>Business Unit</td>
<td>The converter processes the field through the Business Unit scrub routine. This routine right adjusts and fills the field with blanks.</td>
</tr>
<tr>
<td>Data Dictionary Default</td>
<td>The converter uses the reference field in the To file to access the data dictionary and retrieve the default value for the field.</td>
</tr>
<tr>
<td>Initialization</td>
<td>Fields in the To file are initialized to blanks for alphanumerics and zeros for numerics if no fields are defined to map to them.</td>
</tr>
<tr>
<td>Next Number</td>
<td>You can specify to have a next number value assigned to a field.</td>
</tr>
<tr>
<td>Check Data Dictionary</td>
<td>You can specify to have the value of the field validated against the data dictionary values, ranges, and user defined codes.</td>
</tr>
<tr>
<td>User Defined Code Lookup</td>
<td>Use the fields in the From file to look up a user defined code (UDC) and return the associated value in the Description 1 field as the To field value.</td>
</tr>
<tr>
<td>Default Constant</td>
<td>Specify constant value, up to six characters, for the To field value.</td>
</tr>
</tbody>
</table>
Database Considerations

The system creates records in the Cross-Over Rules file for each version of cross-over rules you specify. This file contains information about the fields in the From file and the To file and how the two fields are associated.

If the field lengths or characteristics of the files that the cross-over rules have been built upon change, you must redefine the cross-over rules. Otherwise, the rules are based on the erroneous field descriptions.

The system handles extra calculations through called programs specified in the Cross-Over Rules file for each field.

User Responsibilities

You are responsible for developing and maintaining the cross-over instruction rules. If the From file or To file definition of the cross-over instructions changes, you must revise the Cross-Over Rules.

Perform the following tasks:

- Set Up Universal File Converter
- Work with Crossover Rules
- Work with File Conversion
- Print a Report
- Create Conversion Versions
- Work with the Data Dictionary Glossary by File
Set Up Universal File Converter

About Universal File Converter (UFC)

The Universal File Converter is used to import data from one AS/400 physical file to another AS/400 physical file. If you have more than one file to convert, you can set up a separate version for each type of conversion required. The Universal File Conversion Setup program loads information to the Crossover Rules file (F0031) about the fields in the files you are converting.

A major benefit of using UFC is that once the process has been tested, it can be made into a production process using a Dream Writer version.

The system uses the information in the Crossover Rules file to transfer the data from a field in one file to a field in another file, or to a field in multiple files.

This program also has processing options that let you convert data from both JD Edwards World and non-JD Edwards World files.

This section contains the following:

- Understanding the Universal File Converter Setup
- Setting Up Universal File Converter

Before You Begin

- Before you run the setup procedure make sure the To files exist.

Caution: Do not attempt to use the Universal File Converter on a file that contains double byte data. The converter program may corrupt the integrity of the bracketing “shift in” and “shift out” characters that are automatically inserted by double byte terminals.

Understanding the Universal File Converter Setup

The setup program is the first part of a three-part conversion process. Specify a From file and a To file through the DREAM Writer processing options. You can specify up to four To files. If you require multiple From files, specify a join logical as the From file. This join logical is over all the files you select for the From file. Use the name of the join logical in the first processing option.

The program retrieves field information for all fields in the From file and loads this information to the Crossover Rules file, F0031.

The program then retrieves field information for the To files. If the Reference (data dictionary) field in the To file matches the From file Reference field, the program
makes an association between the two fields. The system writes information for the
To file to the record in the Crossover Rules file associated with the From file field.

**FILLER conversions are automatically generated for From file fields with no
corresponding To file fields and for To file fields with no corresponding From file
fields. **FILLER fields will be ignored during the conversion process. You can
override a **FILLER entry with the appropriate field name, position, and
characteristics if the field exists in the file but has a different field name.

If there are any other associations you need, do them manually using the Crossover
Rules selection on the menu.

Considerations

UFC is written to map data character-by-character from any physical file resident on
the iSeries (AS/400) to any other resident physical file. If the input file resides on a
PC, it must be transferred to the iSeries via a data transfer utility.

If you are using UFC to populate Z1 files, Z1 batch processing is recommended for
uploading data to the appropriate production files. Areas where these processes are
available include:

<table>
<thead>
<tr>
<th>Area</th>
<th>Menu</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address Book</td>
<td>G01313</td>
</tr>
<tr>
<td>General Accounting</td>
<td>G09311</td>
</tr>
<tr>
<td>Accounts Payable</td>
<td>G04311</td>
</tr>
<tr>
<td>Accounts Receivable</td>
<td>G03311</td>
</tr>
<tr>
<td>Time Card Entry</td>
<td>G07121</td>
</tr>
<tr>
<td>Budgeting</td>
<td>G1421</td>
</tr>
<tr>
<td>EDI</td>
<td>G47 menus</td>
</tr>
<tr>
<td>Batch Sales Order Entry</td>
<td>G4212</td>
</tr>
</tbody>
</table>

To protect your production data, create a test library and put a copy of the “from
file” and “to file” with attached logical files in it. There may be a logical file attached
to the to file with a unique key requirement. By having this file attached during the
UFC process, all the key field requirements will be met. Now gather the following
two pieces of information. Type command DSPFP on the from file in the test library.
Page down to the last panel. Write down the Format name and the number of
records in the file. The number of records should be 100 or less for the initial testing.
During testing, the number of records can be used to make sure all records were
read. When the conversion has been tested successfully a full-file test can be made.
The record format name will be used in the last step.
Setting Up Universal File Converter

To create a new version

1. From the Universal File Converter menu (G9841), choose Versions Setup.

   The Versions Setup form displays. The examples shown are for illustrative purposes only.

   This program loads information into the Crossover Rules File (F0031) about the fields in the files you are converting. The system uses the information in the Crossover Rules File to transfer the data from a field in one file to a field in another file or to a field in multiple files. Neither file has to be a JD Edwards world file. However, the from file cannot be a multi-member type file. To verify the number of members in a file do the DSPFD command on the from file. Look for the Number of members field. The value should be a one.

2. Make a copy of the XJDE0001 version.

   Use your version name and title to tie it to the data you are converting, for example ABCONV and “Address book conversion.”

   Caution: Do NOT make any changes on the Additional Parameters screen. The based on file should be *NONE and the Format name should be INONE.

3. Display the processing options.
4. Specify a From file and a To file. You can specify up to four To files. If you require multiple From files, specify a join logical as the based on file for your version. The join logical will encompass all the files you wish to use for the From file.

**Note:** If either file is set to JD Edwards World file = Y, numeric field names will be looked up for decimal point information. If either file is not created with Data Dictionary data item name, make sure to set the JD Edwards World file = N. Page down to enter the test library name.

5. Return to the Versions List.

6. Execute the version.

After entering the file names and the libraries in the processing options, you do need to submit the DREAM Writer version. P00120 will find the “from” and “to” files and create a record in F0031 for each field in these files. If there is no print file, the F0031 will have version records added for each field in the setup in field name sequence.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter the name of the file</td>
<td>The name of the From file to convert the data from.</td>
</tr>
<tr>
<td>JD Edwards World File?</td>
<td>Y if the From file is a JD Edwards World file, or N if it is not. If Y, the first two characters of the field name will be stripped and the remaining field name will be looked up in the JD Edwards Data Dictionary.</td>
</tr>
<tr>
<td>Enter the name of the file</td>
<td>The names of the To files in the spaces provided.</td>
</tr>
<tr>
<td>OR files to convert the data to</td>
<td></td>
</tr>
<tr>
<td>JD Edwards World File?</td>
<td>Y if the To file is a JD Edwards World file, or N if it is not.</td>
</tr>
</tbody>
</table>
To set up crossover rules

This is where the mapping details are entered. There are many conversion rules in UFC to cover most situations, try them first before using custom Xxxxx programs. You need only map the fields that you are concerned with.

1. From menu G9841, take Selection 3 to go to the Crossover Rules.

2. Inquire on the version name you created and run in Step 2. P00120 is pre-loaded in the Form ID field.

   **Note:** The program in Step 2 above attempts to match field names. Any field with no match will have **FILLER for a Field Name. If **FILLER is on either the “From” or “To” side, the data will not be mapped. If the “From” side has fewer **FILLER fields, press F8 to clear them. If the “To” side has fewer, press F9.

The remaining fields will be in alphabetical order, not by position within the record. UFC is usually used to bring data into a JD Edwards World file, so that assumption will be used in this exercise.

3. Pick a field on the “To” side that you want to map into.

   Start with the field name on the “From” side and key in a meaningful field name. Use F14 to enter additional documentation.

4. Set the “From” field “type” to A for Alpha.

   This is the most inclusive data type and will be used for numeric fields as well.

   You will need to know where the data to be mapped starts in the input record. UFC moves data character-by-character, so if the input data has leading spaces, increment the “Begin Pos” field to skip over them. Similarly, on the “To” side, manipulate the “Begin Pos” as necessary so the data will be mapped correctly.
The field names are not important to the process. The type, position and length values are important.

5. On the “To” side, the type can be “A”, “S”, or “P” for Alpha, Signed or Packed. “A” fields will have the “Byte” field set to the length of the data and the “Dig” field should be 0. “S” fields should have the “Bytes” and “Dig” fields set to the same value. “P” fields should have the “Dig” set to ((“Bytes” x 2) – 1). It takes 8 bytes to hold 15 packed digits. In most cases, the “Dec” field will be 00.

6. Press F4 to open the fold area. Press F1 on the “Conv Rule” field. The most used functions are the date formats, *DFT and *NN. Most file dates are stored in Julian format in JD Edwards World World. The incoming data should be 6 digits long and will probably be in a *MDY or *DMY format. On the “To” side, the field will also be 6 digits long but will have the *JUL format. The *DFT rule has room for 6 contiguous digits of data. *NN uses the first four digits for system code and the next 2 for the bucket number.

7. When all the incoming fields with data have been mapped, type C in the action code and press enter. All fields in the “To” file will be initialized to the proper empty format if not mapped.

See Also

- Work with Cross Over Rules

To create a new version of P00111

1. On menu G9843, take Selection 4 to go to the Versions List for P00111.

2. Make a copy of the XJDE0001 version and give it the same name and title as in Step 2A for continuity purposes.

3. On the Additional Parameters screen:
   - Enter the input file name in the Based on file field.
   - Press Enter.
   - Press F12 to return to the Add’l Parameters screen.
   - Ensure the format name is the one from the DSPFD command in the preparation step.
   - Press Enter.

4. Display the Processing Options.

5. Set the Processing Options as follows:
   - Processing Option 1 needs the version of P00120 from Step 2 so it can find the right crossover rule records in the F0031.
   - Enter the appropriate file and library names for options 2, 3, and 4.
   - Option 5 is usually set to 1 to clear the file so just the records from the latest conversion will be there.
   - Option 6 is usually blank.
Caution: Do not change Data Selection and Data Sequencing. They are used for the OPNQRYF statement. Since we want to use the full file, they don’t apply.

6. Return to the Versions List.
7. Submit your version.

To review your results

1. Go to Work with Spooled Files and look for R00111.
   At the bottom of this spooled file will be how many records were read and written.
2. Type command DSPPFM on the “to” file to see if the mapping is correct.
3. In order to see all the data, press F10, then F11 to see the hexadecimal representation of the data in an over/ under presentation.

Trouble Shooting

- Sometimes P00111 will have problems with the input file.
  - On the Additional Parameters screen, if the file output type is 1, change it to 2 and vice versa.
- Make sure the P00111 based on file and format names are correct.
  - See To create a new version of P00111.
- There may be just one record in the “to file” and a message in the Joblog about a duplicate record.
  - Type command DSPFD on the file and its logicals to see which one has the “unique” key requirement. You will need to map data to each of the key fields. If there is no matching data, maybe a *NN conversion rule will have to be attached to one of the key fields as a tie breaker.

Key Words

Valid conversion rules keywords include:

<table>
<thead>
<tr>
<th>Function</th>
<th>Key Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Conversion</td>
<td>*MDY, *DMY, *YMD, *JUL, *SYSVAL</td>
</tr>
<tr>
<td>Initialization</td>
<td>*BLANKS, *ZEROES</td>
</tr>
<tr>
<td>Business Unit</td>
<td>*RAB right adjust blank fill</td>
</tr>
<tr>
<td>Default from Dictionary</td>
<td>*DEF User Defined Code Lookup, *UDCxxxxxx where xxs is the System Code and rr is the Code Type</td>
</tr>
<tr>
<td>Default Constant</td>
<td>*DFTcccccc where ccccc is the constant</td>
</tr>
<tr>
<td>Function</td>
<td>Key Words</td>
</tr>
<tr>
<td>-----------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Next Number</td>
<td>*N N:ss:sxx where ssss is the System Code and xx is the Number</td>
</tr>
</tbody>
</table>
Work with Crossover Rules

Working with the Crossover Rules Screen

The Crossover Rules screen lets you add, change, and delete crossover rules used in the Universal File Converter process. Use this form to set up or maintain associations between fields in the From file and the To file. You need only map the fields that you are concerned with.

Note: There are many conversion rules in UFC to cover most situations, try them first before using custom Xxxxx programs.

Lines with **FILLER in either the from file or the to files are ignored enabling you to view From file fields with no corresponding To file fields or view To file fields with no corresponding From file fields.

This section contains the following:
- Displaying Field Descriptions
- Adding Fields
- Deleting Records
- Keywords
- About the Conversion Rule Program
- Available Functions and Options

To work with the Crossover Rules form

1. From the Universal File converter menu, choose Crossover Rules.
36B Work with Crossover Rules

The form above displays illustrative data only. This is where the mapping details are entered. The From file fields appear on the left. The To file fields display on the right.

2. Complete the Crossover Rules form.

- F8 and F9 are toggles. Press them to suppress or activate the display of the **FILLER fields in the From and To files.
- Use F6 to pull up full information about both fields on the Crossover Rules screen. See To add a field.
- F14 is cursor-sensitive. If you are on a From file field, press F14 to enter text for that field. When the cursor is on a To file field name, press F14 and the Generic Text Form opens for that To file field name. You can also enter text for the From file and To file by placing the cursor on the appropriate field. The field name is highlighted on V0031 if generic text exists. For additional information refer to the Advanced Functions Reference Guide.
- Press F4 to display detail information in the detail area.

### Field Explanation

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form Id</td>
<td>This will always be P00120, the Version Setup program.</td>
</tr>
<tr>
<td>From File Name</td>
<td>The file that data is being transferred “from” in the file conversion process.</td>
</tr>
<tr>
<td>Version</td>
<td>This is the version name you created and ran in the Setup step.</td>
</tr>
<tr>
<td>To File Name</td>
<td>The file that data is being transferred “to” in the file conversion process.</td>
</tr>
<tr>
<td>Skip to...</td>
<td>Enter either the FROM field name or the TO field name to skip to in the subfile.</td>
</tr>
<tr>
<td>From File ....</td>
<td>Heading for the fields that data is being transferred “from” in the file conversion process.</td>
</tr>
<tr>
<td>To File ....</td>
<td>Heading for the fields that data is being transferred “to” in the file conversion process.</td>
</tr>
</tbody>
</table>
### Field Explanation

| Field From File Field name | The name of the field in the FROM file for source data that is used in the conversion process. |
| Field From File Field Data Type | The type of data of the field in the “from” file. The data item types are defined in User Defined Codes, system code ‘98’, record type ‘DT’. (F1) Usually A for fields in a PC (ASCII) file. |
| Field From File Field Begin Pos | The beginning position of the field in the “from” file. |
| Field From File Field Size In Bytes | The number of bytes for the field in the “from” file. |
| Field From File Field Number of Digits | The actual number of digits in the “from” file field. In a non-packed field this is the same as the number of bytes. (Numeric fields) |
| Field From File Field Decimal Positions | The number of decimal positions in the “from” file field. (Numeric fields) |
| Field From File Field Description | The description of the “from” file field. If the file is declared a JD Edwards World file, this will be the first part of the Data Dictionary description. |
| REF (Reference) | Use reference field for those conversion rule entries that refer to the Data Dictionary. |
The left side of the form contains information about the From file.

The right side of the form contains information about the To file fields. If the setup program made associations with the To file fields, they display in the right columns when you inquire on a Form ID. Otherwise, these columns contain **FILLER information.

## What You Should Know About

You should be aware of the following rules when you work with crossover rules.

| **To review a specific set of crossover rules** | Enter the DREAM Writer version you used to create the rules. |
| **To update information on Crossover Rules form** | Enter the To file field, type, beginning position, number of bytes, and number of digits and decimals, if applicable. Required information is name, type, beginning position and number of bytes. |
| **“Skip to” capabilities** | Two “skip to” capabilities are available on this form. You can skip to a field in either the From file or the To file. |
| ***DFT** | When using *DFT if the literal is over 6 positions then part it out as if dealing with two fields. |
| **Packed numeric** | Packed numeric values use ½ their length plus one in a file. A numeric field defined as a length of 15 only uses 8 bytes. Negative, or signed fields, must have the ‘sign’ character in the FROM file. In the Crossover Rules, when using a User Defined Code Lookup keyword (*UDCssssrr) and the system code is only 2 numbers, such as 55, enter it as _55 (with leading blanks), not 0055 or 5500. |
| **Negative, or signed fields** | Negative, or signed fields, must have the ‘sign’ character in the FROM file. |
| **Using lookup keyword *UDCssssrr** | In the Crossover Rules, when using a User Defined Code Lookup keyword (*UDCssssrr) and the system code is only 2 numbers, such as 55, enter it as _55 (with leading blanks), not 0055 or 5500. |

## Displaying Field Descriptions

### To display field descriptions

1. Press F13 in the Field Name column for the From or To file. The File Field Descriptions form displays, as shown below.
When you use option 4 to select a field from the form, the program returns the name, type, number of bytes, number of decimals, number of digits, description, reference, and key position to the appropriate fields on the form. For Crossovers on the File Field Descriptions window, refer to the Computer Assisted Design Reference Guide.


2. Enter 4 in the option field. The program returns the field description to the associated field as shown in this example.

### Adding Fields

To add a field:

1. Press F6 to open the Add Crossover Instructions form.
This window presents most of the fields on the Crossover Rules screen.

2. With the cursor in the Field Name field on the Crossover Rules screen, press F13 to open the File Field Descriptions form.

3. Select a field using the 4 option.

4. Press Enter.

   The program returns the field information to the Field Name when you exit the form.

The required fields for adding a field are:

- From field name, type, number of bytes, and beginning position
- To field name, type, number of bytes, and beginning position

The add function is available to associate a single field in the From file with multiple fields in the To file and to break apart a From field into multiple fields.

A field can exist in the To file and have nothing associated with it in the From file. In this case, the To file field is initialized as described in the section Special Processing in the Introduction of this guide.

Deleting Records

To delete a record

Choose option 9 to delete records from the Crossover Rules file.

- This cancels the From-To relationship so that no conversion takes place.
- If you blank out the To file field name, the program does not delete the record from the Crossover Rules file, but only clears the To file field information. The converter program looks only at records that have both a From and To file field name.

**Note:** You do not need to delete lines with blank (**FILLER**) To file field names, they are automatically skipped in the conversion process.
Keywords

Keywords in the Conversion Rule field (in the fold area) trigger special processing for a field before the data is transferred. Following are the keywords that are available and a brief explanation of what processing they trigger.

**Caution:** With the exception of the date keywords listed below, specify conversion rules for either From field or To field, never for both.

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dates - *MDY, *DMY, *YMD, *JUL, *SYSVAL</td>
<td>These keywords activate a date conversion between the From file field and the To file field. You must type keywords into both the From file Conversion Rule field and the To file Conversion Rule field. Each keyword on the From field specifies how the field is stored in the From file. The keyword on the To field conveys the output format on the To field. <strong>Note:</strong> This does not work on packed fields.</td>
</tr>
<tr>
<td>Business Unit - *RAB.</td>
<td>This keyword activates the business unit scrub of right adjust and blank fill to the From file field before moving it to the To field.</td>
</tr>
<tr>
<td>Initialize - *ZEROES, *BLANKS</td>
<td>These keywords move either zeroes or blanks to the From file field before it is transferred. With the initialization rules, these keywords are not required unless you want to initialize an alphanumeric field to zeroes.</td>
</tr>
<tr>
<td>Data Dictionary Default - *DEF</td>
<td>This keyword retrieves the Data Dictionary default for the To file field, using the Reference field in the Data Dictionary, and loads it to the From file field before it is transferred.</td>
</tr>
<tr>
<td>User Defined Code - *UDCsssr</td>
<td>This keyword retrieves the definition of the user defined code used in a specific system and loads it to the To field. When typing your request, ssss is the system and rr is the user defined code.</td>
</tr>
<tr>
<td>Default Constant - *DFTcccccc</td>
<td>This keyword loads a default constant to the To field. When typing your request, ccccccc is the default constant.</td>
</tr>
<tr>
<td>Next Number - *NNssxx</td>
<td>This keyword computes a next number and loads it to the To field. When typing your request, ssss is the system and xx is the bucket number.</td>
</tr>
<tr>
<td>Check Data Dictionary - *CHK.</td>
<td>This keyword lets you edit a field against Data Dictionary values and ranges. The results of the edit print on the File Conversion report whenever any errors are detected.</td>
</tr>
</tbody>
</table>
About the Conversion Rule Program

Besides specifying the use of keywords in the conversion rule, you can specify an external program that runs before the data is transferred to the To file field. You must name the external program beginning with an X. For example, use an “X” program to determine a range of valid values in a From file field, excluding records based on a given field. Other examples include writing multiple To file records based on a single From file record, or manipulating the data before it is transferred.

The external program requires five parameters:

<table>
<thead>
<tr>
<th>PARM</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>First parameter</td>
<td>Must be 50 bytes and contains the value of the field being processed. Use it to send the value to the converter program and receive the value when the “X” program is done with it.</td>
</tr>
<tr>
<td>Second parameter</td>
<td>One-byte error flag. If the error flag returns blank, the data in parameter 1 from the “X” program is placed in the To file.</td>
</tr>
<tr>
<td></td>
<td>• If the error flag returns with 2, the data in parameter 1 is not transferred to the To file. Use this error if you are writing multiple To file records and different From file fields are used for a single field in the To file.</td>
</tr>
<tr>
<td></td>
<td>• If the error flag returns with 3, a record will not be written to the To file. Use this error if you do not want to write a record when the value of a certain field in parameter 1 is blank, zero, or not valid for your purposes.</td>
</tr>
<tr>
<td>Third parameter</td>
<td>Four-byte alphanumeric field for the number of the To file records. The field always has numeric characters and is zero-filled. This lets your “X” program know which record the converter program will write when you are writing multiple To file records.</td>
</tr>
<tr>
<td>Fourth parameter</td>
<td>Ten-byte field for the From file field name. This lets your “X” program know which field you are processing if multiple fields in the From file are updating a single To file field.</td>
</tr>
<tr>
<td>Fifth parameter</td>
<td>Ten-byte field for the To file field name. This lets your “X” program know which field you are processing if multiple fields in the From file are updating a single To file field.</td>
</tr>
</tbody>
</table>
Available Functions and Options

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| F6       | Add Instructions  
To add fields to be converted, press F6 to access the Add Cross Over Instructions Form. The required fields for adding a field are Field Name, Field Data Type, Field Beginning Position, and Number of Bytes. |
| F8       | Suppress From **FILLER Fields  
Will not display those lines with **FILLER values in the From field. |
| F9       | Suppress To **FILLER Fields  
Will not display those lines with **FILLER values in the To field. |
| F13      | File Field Description  
Place cursor on any Field Name field and press F13 to display the File Field Description form. |
### Function Description

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| F14      | User Defined Text  
Highlight to indicate that there is generic text associated with this field.  
- Press F14 in the top area of the form to enter text about the conversion.  
- Press F14 in the From Field area (left side of the form) to enter text describing the From Field.  
- Press F14 in the To Field area (right side of the form) to obtain text describing the To Field.  
The field will highlight to indicate that there is generic text associated with this field. |

### Option 9 - Delete Records

To delete records so that no conversion takes place, enter Option 9. If you blank out the To File Field Name, the program does not delete the record from the Cross Over Rules file (F0031), but only clears the To File Field information. The converter program will only look at records that have both a from and to file field name.
Work with File Conversion

Working with File Conversion

The File Conversion program accesses the Crossover Rules file (F0031) and transfers data fields from one file to another, from one file to multiple files, or from multiple files to one file.

To run file conversion

1. From the Universal File Converter menu (G9841), choose File Conversion.

   ![Universal File Converter Menu]

   J.D. Edwards & Company
   Universal File Converter

   ... DATA FILE CONVERSION
   2. Version Setup
   3. Crossover Rules
   4. File Conversion
   5. Report

   Selection or command
   ****

   Note: When creating an execution form, be sure the Based on File and the Format Name fields on the Additional Parameters screen contain your Based-on filename and the correct Format name for that file. The Data selection and sequence records should be left as-is since the converter reads the entire from file.

2. Add your own version from a Demo version and go to the processing options of your new version.
### Option Description

**Enter the name of the Form ID and version containing the Initial Setup step.**

The Form ID will be P00120. Enter your version from the conversion specifications.

**Enter the name and library of "from" file, if different than the Form ID and version specified.**

Type the name of the From file and library, if it is different from the From file and library in the Form ID and version specified above. This file must have the same organization as the file used in the crossover rules.
### Work with File Conversion

**Option**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter a “1” to clear the file data is being transferred to. Enter “1” to clear the To file. The To file will be filled only with converted records. If this field is left blank, the converted data records are added to the To file. Enter the number of To file records you want to create for each From file record. Enter the number of To file records to be created for each From file record. Enter the correct values on Processing Options and submit your version to complete the conversion process.</td>
</tr>
</tbody>
</table>

#### What You Should Know About

**Multiple "From" files**

If you are using multiple From files, remember to create a join logical over all the From files you want to use.

**New versions**

When adding a new version, check to see that the format name for the based-on file is correct for the file. The default is Ixxxx and may not be appropriate.

**"From" file name and the "To" file formats**

The From file name and the To file formats should be the same as used to set up the conversion rules in Step 1.

**Error conditions**

A printed report lists error conditions detected by *CHK keyword and lists the total number of records read and number of records converted. The report lists the description of the errors. Depending on the error condition, you may need to correct the values in the incoming data and rerun the conversion.
### Trouble Shooting

<table>
<thead>
<tr>
<th>Problem</th>
<th>Explanation / Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incorrect Value in the Format Name field.</td>
<td>The most common UFC error is an incorrect value in the Format Name field on the Additional Parameters screen for P00111. The Format Name is for the Based On File (the from file). To verify the format name do a DSPFD on the from file. The file format is listed at the bottom of the display. Always exit the version and re-inquire on the Additional Parameters to verify that the Format Name change was accepted.</td>
</tr>
</tbody>
</table>
| Not writing to the output file               | Problems writing to output file (in P00111) may be because of duplicate keys.  
  - Check if the to file has a logical attached that has a unique key and verify the user is not populating that field with blanks.  
  - The File Output Type field on the Additional Parameters screen for P00111 DREAM Writer should be a 2 (logical). Use a file output type of a 1 (OPNQRYF) for files that contain DDS (JD Edwards World files). If one type does not work, try the other.  
  - Data Definition Specifications (DDS) means the file is formatted (the lengths of the fields in the record are defined). Files used in UFC do not have to be formatted (contain DDS) but they must be orderly (all fields in each record must be in the same place.) Use DSPPFM to see the data in each record. Use F10, F11 to see the hexadecimal value and the ASCII value.  
  - If all setup options look correct check for multiple F0031 files – you may be executing out of the wrong Crossover Rules file. The data in this file is created when running the first DREAM Writer, P00120 (2/ G9841). |
| Job log “version for P00111 cannot be found” | If you receive a job log after running P00111 that states the version for P00111 cannot be found, check that only one set of DREAM Writer files exist in your library list. Otherwise unpredictable results may occur. |
Print a Report

Printing a Report

You can print a report that displays the Cross Over Rules and any associated generic text.

To print a report

1. From the Universal File Converter Menu (G9841), choose Report.

```
<table>
<thead>
<tr>
<th>0</th>
<th>Version</th>
<th>Description</th>
<th>User</th>
<th>Chg Date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>XJDB0001</td>
<td>File Converter Report</td>
<td>DEMO</td>
<td>10/25/91</td>
</tr>
</tbody>
</table>
```

Opt: 1=Run 2=Chg 3=Add 4=Rpt Dist 5=Cover 6=Rpt Ovr 8=Repair 9=Dlt

The form can list different versions of the File Converter Report. The example shown is for illustrative purposes only.

2. On the Data Selection form, specify your Form ID and version.

```
| 0 | Processing Options | Revisions | Form ID |...| D001EJ1 |
|---|--------------------|-----------|---------|...|---------|
|   | File Converter Report |           |         |...|         |
|   | This job has various options described below. Enter the desired values and press ENTER to continue. |           |         |...|         |
|   | 1) Enter a '1' to print Data Dictionary | 1         |         |...|         |
|   | Glossary for each item. Leave blank to not print the Data Dict. Glossary. |         |         |...|         |
|   | (Prints for "TO" fields only) |           |         |...|         |
|   | 2) Enter a '1' to print File Specific | 1         |         |...|         |
|   | Glossary for each data item. Leave blank to not print. |         |         |...|         |
|   | (Prints for "TO" fields only) |           |         |...|         |
|   | 3) Enter a '1' to print the Generic | 1         |         |...|         |
|   | Text instructions for each data item. Leave blank to not print the Generic |         |         |...|         |
|   | Text. | (Prints for both "FROM" and "TO" fields) |         |...|         |

PS=Printer Overrides
```
3. Select one of the following print options:

<table>
<thead>
<tr>
<th>Field</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enter a “1” to print Data Dictionary Glossary for each item.</td>
<td>Prints Data Dictionary Glossary for each To field.</td>
</tr>
<tr>
<td>Enter “1” to print File Specific Glossary for each item.</td>
<td>Prints file specific glossary from Generic Text file (F00163) for each To file.</td>
</tr>
<tr>
<td>Enter “1” to print generic text instructions for each item</td>
<td>Prints any generic text associated with either To fields or From fields.</td>
</tr>
</tbody>
</table>
Create Conversion Versions

Creating Conversion Versions

Universal File Converter enables you to create conversion forms to use for planning purposes when you convert your non-JD Edwards World files into JD Edwards World files.

- Start by creating a form that specifies the major file in the “Convert to” file. The name of the file you convert from is intentionally left blank. This lets you create a blank set of conversion rules which you can print using the Report selection.

- JD Edwards World supplies a special Data Dictionary glossary relating to specific fields in specific files in your JD Edwards World Data Dictionary text. You can also create new field descriptions that better correspond to your system by pressing F14 for generic text in the crossover rules revisions.

- If you decide to use the blank version (described above) for actual file conversion, type the From file specifications corresponding to the appropriate To field using the Crossover Rules. Be sure to override the From file before you execute the conversion program.

Creating Conversion Forms

To create a conversion form

1. From Universal File Converter (G9841), choose either Versions Setup or Report.

2. Complete the Processing Options Revisions form.

<table>
<thead>
<tr>
<th>Processing Options Revisions</th>
<th>Form ID: ...</th>
<th>D00120 __</th>
<th>Version: ...</th>
<th>XJDB0801</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generate Cross Over Instruction - Sample</td>
<td>Display Level:</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   This job has various options described below. Enter the desired values and press ENTER to continue.

   **FILE SPECIFICATION:**
   1. Enter the name of the file to convert the data from.
   - JDE File?
   2. Enter the name of the file OR files to convert the data to.
   - File 1 JDE File? F32B01V
     - File 2 JDE File?
     - File 3 JDE File?
     - File 4 JDE File?

   F5=Printer Overrides +
- If you selected Versions Setup, be sure to leave the first processing option blank under File Conversion.

- In the second option, type the name of the files you want to convert, and then Y if they are JD Edwards World files or N if they are not.

<table>
<thead>
<tr>
<th>98312</th>
<th>Processing Options Revisions</th>
<th>Form ID.</th>
<th>Z003151</th>
<th>Version.</th>
<th>ZG003901</th>
</tr>
</thead>
<tbody>
<tr>
<td>File Converter Report</td>
<td>Display Level. 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This job has various options described below. Enter the desired values and press ENTER to continue.

1) Enter a "I" to print Data Dictionary Glossary for each item. Leave blank to not print the Data Dict. Glossary. (Prints for "TO" fields only) 1

2) Enter a "I" to print File Specific Glossary for each data item. Leave blank to not print. (Prints for "TO" fields only) 1

3) Enter a "I" to print the Generic Text. Instructions for each data item. Leave blank to not print the Generic Text. (Prints for both "FROM" and "TO" fields) + Bottom

PS=Printer Overrides

If you select Report, type 1 next to all three options as shown above.
Work with the Data Dictionary Glossary by File

About Working with the Data Dictionary Glossary by File

When using the Universal File Converter, small details often differ for each file. Keeping these details clear, especially when the conversion form might be used by another department, is a potential problem. To remedy this, JD Edwards World has made it possible to attach Data Dictionary glossary text to each data item that explains the details particular to that specific file.

To work with the Data Dictionary Glossary by file perform the following tasks:

- Accessing the Data Dictionary Glossary by File
- Adding a File Specific Glossary Item
- Printing the Data Dictionary Glossary Information

Accessing the Data Dictionary Glossary by File

To access the Data Dictionary Glossary by file

1. From Universal File Converter (G9841), type DD on the command line and press Enter.

   The Data Dictionary Repository screen appears.

2. Press F10 to display the glossary definition of the data item you selected.
3. Use the Data Item Glossary Revisions form to change the glossary text for a Data Dictionary item or to add a File-Specific glossary item, as needed.

Adding a File Specific Glossary Item

To add a File Specific Glossary item

From the Data Item Glossary Revisions form
1. Type A in the Action Code field.
2. Type the file name in the Scrn/Rpt field.
3. Type the new text and press Enter.

Printing the Data Dictionary Glossary Information

To print the Data Dictionary Glossary information
1. From Universal File Converter (G9841), choose Report.
2. Complete the Processing Options Revisions form.
3. Type 1 next to all three options to print the Data Dictionary glossary.
4. Use Option 2 to print the File-Specific glossary text.
9 Appendices
Appendix A – Common & Production Library Files

This appendix lists the files that are automatically created in the common and production libraries during the installation process.

Common Library Files Automatically Created by JD Edwards World Build Programs

The following chart lists files automatically generated from the Data Dictionary as a result of a build program that JD Edwards World offers from a menu. It is recommended that these files be maintained in your common library along with the Data Dictionary.

<table>
<thead>
<tr>
<th>File Name</th>
<th>File Description</th>
<th>System Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>F98FRF@</td>
<td>Field Reference - “@” Data Items</td>
<td>98</td>
</tr>
<tr>
<td>F98FRF$</td>
<td>Field Reference - “$” Data Items</td>
<td>98</td>
</tr>
<tr>
<td>F98FRFA thru</td>
<td>Field Reference - “A” Data Items through</td>
<td>98</td>
</tr>
<tr>
<td>F98FRFZ</td>
<td>Field Reference - “Z” Data Items</td>
<td>98</td>
</tr>
</tbody>
</table>

Physical and Logical Files Created in a Common Library

On the SVR screen, there are two fields which govern the location and content of data files in user libraries during an install.

- The first field is Common File. If a file has this field set to “Y”, the file will be created in the user’s Common library, if one is specified.
- The second field is Copy Data (Y/ N). Most files to be located in the Common library will have Copy Data Y but there are some exceptions such as F0016. Most non-Common (User data) files will be Copy Data N but there are exceptions such as F0010 and F0009. No data will be copied for a logical file.
- Inquire on a file in SVR to see if it should be located in Common or not.
- Inquire on a file in SVR to see if data from the pristine environment should be copied into it.
Appendix B – Upgrading Customized Source Code

JD Edwards World provides access to several complementary products. If you have customized JD Edwards World source code, the following products will help you upgrade your source code.

This section contains the following:

- **S/Compare**
- **Harmonizer**
- **About Harmonizer Plus**

**S/Compare**

Overall, S/Compare is a valuable aid used to:

- Identify differences between any two programs
- Simplify the task of documenting program changes
- Simplify the task of consolidating your custom changes into new releases of programs
- Identify differences between the names of the programs in two different files to quickly locate added or deleted programs in the new release

The S/Compare utility is specifically designed to compare two versions of source code. It will locate inserted, deleted, changed, or moved records in a source program. Processing options are provided to include or exclude comment lines, blank lines, and formatting differences. S/Compare's output clearly identifies differences between two source members on a composite list of both programs. An option allows the records that are the same in the programs to be omitted from the listing to produce a report of only the differences between the files. This option also allows a given number of matching records before and after a mismatch to be listed to help in identifying the section of source code.
Features of S/Compare

Some of the features and capabilities of S/Compare are:

- Flags are used in the composite listing to clearly mark statements or blocks of statements that have been inserted, deleted, or moved.
- Records that are moved from one location in the original file to another in the new program are indicated by source and target locations.
- Printing large blocks of identical code can be eliminated by a processing option. Only the differences will be printed and you can control the number of matching lines that are listed before and after each block of mismatched code.
- Differences between your program and the new program can be listed in an edit program.
- There is a processing option that can eliminate mismatches being printed because of spacing between words.

Harmonizer

Harmonizer adds to the capabilities of S/Compare by allowing the comparison of 3 to 16 program versions. Like S/Compare, the comparison results are written in a format that clearly depicts the differences between source members. In addition, Harmonizer has the capability of merging program versions to generate a composite source member. You can control what is written to the composite source member when potential conflicts are found.

Features and Capabilities of Harmonizer

Some of the features and capabilities of Harmonizer are:

- The comparison of 3 to 16 versions of a program.
- Two report formats are available. The MULTI-Compare report compares 3 to 16 programs. The TRI-Compare report is specifically designed for 3 programs.
- Statements from the original file that have been replaced, inserted, or deleted are noted on the comparison reports.
- All of the features of S/Compare are supported by Harmonizer when 3 programs are being compared, except the creation of an edit program which has been replaced by the creation a composite output program.
- The composite program may be compiled immediately or it may be edited. The ScmpEdit utility can be used to remove specified code in the composite program.
- The HARMONIZER command can be used to execute S/Compare and Harmonizer making the utilities easier to use.
Harmonizer Added to S/Compare

- You can incorporate your program changes into new releases easier. Harmonizer can compare the JD Edwards World original program, the JD Edwards World new release, and your customized program to produce a composite source file and a composite report. The composite report notifies you of discrepancies in the replacement, insertion, or deletion of code.
- The Source File Synopsis report produces a comparison of the program names in the JD Edwards World original source file, the JD Edwards World new source file, and your source file to determine any additions or deletions of programs.
- You can merge the development work of several programmers working on the same program.

About Harmonizer Plus

Harmonizer Plus adds to the capabilities of S/ Compare and Harmonizer by helping you manage the ENTIRE process of building a new software release.

About the Project Manager Function

The Project Manager function will display an up-to-the-minute status of every program in your upgrade project. It shows:

- Which merged objects need a programmer review due to conflicts between local changes and vendor changes
- Modified objects that are already created and ones that need to be created
- Objects that are ready for production
- Unmodified objects that must be recreated because they are dependent on modified objects
- Objects that must be present before the object you are working with can be created

Additional Functions

Harmonizer Plus provides a workbench for programmers to perform a variety of functions. Given the proper authority, a programmer can:

- Directly access SEU for editing programs
- Mass compile entire groups of programs
- Selectively compile individual programs
- Selectively create all objects dependent on a modified object
- Add or delete programs from the new production version

Harmonizer Plus identifies unchanged modules that must be recompiled due to changes in prerequisite objects. For example, if you have modified DDS, Harmonizer Plus can identify programs that reference the related files. It can then
recompile those programs. All you need to do is test and move the new libraries into production.
Appendix C – CL Models

J98MODEL1 - Interactive Video

<table>
<thead>
<tr>
<th>Action Code...</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member ID...</td>
<td>J98MODEL1</td>
</tr>
<tr>
<td>Description...</td>
<td>Model CL Program – Interactive Video</td>
</tr>
<tr>
<td>Function Code...</td>
<td>CLP CL Programs</td>
</tr>
<tr>
<td>Function Use...</td>
<td>ISR Model Source Member</td>
</tr>
<tr>
<td>System Code...</td>
<td>28 Technical Tools</td>
</tr>
<tr>
<td>Reporting System...</td>
<td>28 Technical Tools</td>
</tr>
<tr>
<td>Base Member Name...</td>
<td>J98MODEL1</td>
</tr>
<tr>
<td>Maint/RESTDC...</td>
<td>No Option...</td>
</tr>
<tr>
<td>SAR...</td>
<td>9801.242 A/D</td>
</tr>
<tr>
<td>Library/Obj...</td>
<td>library1 library2 file1 file2</td>
</tr>
<tr>
<td>Version...</td>
<td>9801.242 A/D</td>
</tr>
<tr>
<td>User...</td>
<td>Mod</td>
</tr>
<tr>
<td>Date...</td>
<td>07/07/95</td>
</tr>
</tbody>
</table>

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

J98MODEL2 - Batch DREAM Writer without Printer File

<table>
<thead>
<tr>
<th>Action Code...</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Member ID...</td>
<td>J98MODEL2</td>
</tr>
<tr>
<td>Description...</td>
<td>Model CL Program – Batch DREAM Writer without Printer File</td>
</tr>
<tr>
<td>Function Code...</td>
<td>CLP CL Programs</td>
</tr>
<tr>
<td>Function Use...</td>
<td>ISR Model Source Member</td>
</tr>
<tr>
<td>System Code...</td>
<td>28 Technical Tools</td>
</tr>
<tr>
<td>Reporting System...</td>
<td>28 Technical Tools</td>
</tr>
<tr>
<td>Base Member Name...</td>
<td>J98MODEL2</td>
</tr>
<tr>
<td>Maint/RESTDC...</td>
<td>No Option...</td>
</tr>
<tr>
<td>SAR...</td>
<td>9801.242 A/D</td>
</tr>
<tr>
<td>Library/Obj...</td>
<td>library1 library2 file1 file2</td>
</tr>
<tr>
<td>Version...</td>
<td>9801.242 A/D</td>
</tr>
<tr>
<td>User...</td>
<td>Mod</td>
</tr>
<tr>
<td>Date...</td>
<td>07/07/95</td>
</tr>
</tbody>
</table>

Opt: 1-Browse 2-Edit 3=Copy 5=SAR 8=Print 9=Dlt 10=Design 14=Crt
J98MODEL3 - Interactive Video Prompt

9801  Software Versions Repository
Action Code... J
Member ID... J98MODEL3
Description... Model CL Program - Interactive Video Prompt
Function Code... CLP - CL Programs
Function Use... 198 Model Source Member
System Code... 98 Technical Tools
Reporting System... 98 Technical Tools
Base Member Name J98MODEL3
Maint/RSTDSP... Date Option... 0 Generation Sav...
Copy Data (Y/N)... N Optional File... N Common File... N

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

J98MODEL4 - Interactive/Batch with Processing Options

9801  Software Versions Repository
Action Code... J
Member ID... J98MODEL4
Description... Model CL Program - Interactive/Batch with Processing Options
Function Code... CLP - CL Programs
Function Use... 198 Model Source Member
System Code... 98 Technical Tools
Reporting System... 98 Technical Tools
Base Member Name J98MODEL4
Maint/RSTDSP... Date Option... 0 Generation Sav...
Copy Data (Y/N)... N Optional File... N Common File... N

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

J98MODEL5 - Batch Report Writer - No DDS File

9801  Software Versions Repository
Action Code... J
Member ID... J98MODEL5
Description... Model CL Program - Batch Report Writer - No DDS File
Function Code... CLP - CL Programs
Function Use... 198 Model Source Member
System Code... 98 Technical Tools
Reporting System... 98 Technical Tools
Base Member Name J98MODEL5
Maint/RSTDSP... Date Option... 0 Generation Sav...
Copy Data (Y/N)... N Optional File... N Common File... N

Opt: 1-Browse 2-Edit 3-Copy 5-SAR 8-Print 9-Dlt 10-Design 14-Crt
**J98MODEL6 - Batch Report Writer OPNQRYF**

```
9801 Software Versions Repository
Action Code... I
Member ID... J98MODEL6
Description... Model CL Program - Batch Report Writer OPNQRYF
Function Code... CLP CL Programs
Function Use... 198 Model Source Member
System Code... 58 Technical Tools
Reporting System 58 Technical Tools
Base Member Name J98MODEL6 File Prefix... __
Maint/BSTDSP... Omit Option... O Generation Seq.
Copy Data (Y/N)... N Optional File... N Common File... N

<table>
<thead>
<tr>
<th>Source</th>
<th>Object</th>
<th>SAR</th>
<th>Version</th>
<th>S D</th>
<th>User</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Library</td>
<td>File</td>
<td>Number</td>
<td>ID</td>
<td>O P</td>
<td>ID</td>
</tr>
<tr>
<td>JDPCET3</td>
<td>JDPCET3</td>
<td>JDRCDC</td>
<td>867523</td>
<td>AJ1</td>
<td>1</td>
<td>BACK</td>
</tr>
</tbody>
</table>

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

**J98MODEL7 - Batch Report Writer OPNQRYF w/OQF Reset**

```
9801 Software Versions Repository
Action Code... I
Member ID... J98MODEL7
Description... Model CL Program - Batch Report Writer OPNQRYF w/OQF Reset
Function Code... CLP CL Programs
Function Use... 198 Model Source Member
System Code... 58 Technical Tools
Reporting System 58 Technical Tools
Base Member Name J98MODEL7 File Prefix... __
Maint/BSTDSP... Omit Option... O Generation Seq.
Copy Data (Y/N)... N Optional File... N Common File... N

<table>
<thead>
<tr>
<th>Source</th>
<th>Object</th>
<th>SAR</th>
<th>Version</th>
<th>S D</th>
<th>User</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Library</td>
<td>File</td>
<td>Number</td>
<td>ID</td>
<td>O P</td>
<td>ID</td>
</tr>
<tr>
<td>JDPCET3</td>
<td>JDPCET3</td>
<td>JDRCDC</td>
<td>867523</td>
<td>AJ1</td>
<td>1</td>
<td>BACK</td>
</tr>
</tbody>
</table>

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

**J98MODEL8 - Control File Driven Batch Process**

```
9801 Software Versions Repository
Action Code... I
Member ID... J98MODEL8
Description... Model CL Program - Control File Driven Batch Process
Function Code... CLP CL Programs
Function Use... 198 Model Source Member
System Code... 58 Technical Tools
Reporting System 58 Technical Tools
Base Member Name J98MODEL8 File Prefix... __
Maint/BSTDSP... Omit Option... O Generation Seq.
Copy Data (Y/N)... N Optional File... N Common File... N

<table>
<thead>
<tr>
<th>Source</th>
<th>Object</th>
<th>SAR</th>
<th>Version</th>
<th>S D</th>
<th>User</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Library</td>
<td>Library</td>
<td>File</td>
<td>Number</td>
<td>ID</td>
<td>O P</td>
<td>ID</td>
</tr>
<tr>
<td>JDPCET3</td>
<td>JDPCET3</td>
<td>JDRCDC</td>
<td>867523</td>
<td>AJ1</td>
<td>1</td>
<td>BACK</td>
</tr>
</tbody>
</table>

Opt: 1-Browse 2=Edit 3=Copy 5=SAR 8=Print 9=Dlt 10=Design 14-Crt
```
Appendix D – Universal File Converter

About Universal File Converter

This is a simple, single file to single file walk-through of the JD Edwards World World Universal File Converter (UFC). A major benefit of using UFC is that once the process has been tested, it can be made into a production process using a Dream Writer version.

The UFC is written to map data character-by-character from any physical file resident on the System I (iSeries / AS400) to any other resident physical file. If the input file is from a PC spreadsheet, that file will have to be ported (FTP or other process) to the iSeries.

Caution: To protect your production data, create a test library and put a copy of the “from file” and the “to file” with its attached logicals into it. There may be a logical file attached to the “to file” with a “unique” key requirement. By having this file attached during the UFC process, all the key field requirements will be fulfilled.

Gathering Information

There are 2 pieces of information to gather:

- The format name
- The number of records in the file

The record format name will be used in the last step. The number of records should be 100 or less for the initial testing. During testing, the number of records can be used to make sure all records were read. When the conversion has been tested successfully a full-file test can be made.

To gather information

1. Do a DSPFD on the “from file” in your test library.
2. Page down to the last panel.
3. Write down the Format name and the number of records in the file.

To identify from and to files

1. You will need to be signed on to JD Edwards World World. You must have an F0031 file in this environment. Go to menu G9841.
2. Selection 2 will take you to a versions list for P00120.
   - Make a copy of the XJDE0001 version. Use your version name and title to tie it to the data you are converting, for example ACONV and “Address book conversion.”
   - Do NOT make any changes on the Additional Parameters screen. The based on file should be *NONE and the Format name should be INONE.
   - On the processing Options screen, enter the “from” and “to” file names. If either file is set to JD Edwards World file = “Y”, numeric field names will be looked up for decimal point information. Page down to enter the test library name.
     - Processing options three and four pertain to the library for test data.
   - Upon returning to the Versions List, run your version. P00120 will find the “from” and “to” files and create a record in the F0031 for each field in these files. If P00120 has no print files after it runs, it was successful.

To map data

Selection 3 takes you into the crossover rules. This is where the mapping details are entered.
   - P00120 is pre-loaded. Inquire on the version name you created and ran in Step 2.
   - The program in Step 2 above attempts to match field names. Any field with no match will have **FILLER for a Field Name. If **FILLER is on either the “From” or “To” side, the data will not be mapped. If the “From” side has fewer **FILLER fields, do an F8 to remove them. If the “To” side has fewer, do an F9.
   - The remaining fields will be in alphabetical order, not by position within the record. UFC is usually used to bring data into a JD Edwards World file, so that assumption will be used in this exercise.
   - Pick a field on the “To” side that you want to map into. Start with the field name on the “From” side and key in a meaningful field name. Use F14 to enter additional documentation.
     - The system highlights Fields that have additional documentation.
   - Set the “From” field “type” to A for Alpha. This is the most inclusive data type and will be used for numeric fields as well.
   - You will need to know where the data to be mapped starts in the input record. UFC moves data character-by-character, so if the input data has leading spaces, increment the “Begin Pos” field to skip over them. Similarly, on the “To” side, manipulate the “Begin Pos” as necessary so the data will be mapped correctly. The field names are not important to the process. The type, position and length values are important.
     - On the “To” side, the type can be “A”, “S”, or “P” for Alpha, Signed or Packed. “A” fields will have the “Byte” field set to the length of the data and the “Dig” field should be 0. “S” fields should have the “Bytes” and “Dig” fields set to the same value. “P” fields should have the “Dig” set to ((“Bytes” x 2) - 1). It takes 8 bytes to hold 15 packed digits. In most cases, the “Dec” field will be 00.
Open the fold with an F4. Do F1 on the “Conv Rule” field. The most used functions are the date formats, *DFT and *NN. Most file dates are stored in Julian format in JD Edwards World. The incoming data should be 6 digits long and will probably be in a *MDY or *DMy format. On the “To” side, the field will also be 6 digits long but will have the *JUL format. The *DFT rule has room for 6 contiguous digits of data. *NN uses the first four digits for system code and the next 2 for the bucket number.

When all the incoming fields with data have been mapped, put “C” in the action code and press enter. All fields in the “To” file will be initialized to the proper empty format if not mapped.

At the top of the screen, you can inquire with the name of an additional file that you’re mapping data to. Go through the above steps to map the data.

F3 back to the menu.

To convert data

1. Selection 4 takes you to the versions list for P00111.
   - Copy version XJDE0001 and give it the same name and title as in Step 2a for continuity purposes.
   - On the Additional Parameters screen:
     - Enter the input file name in the Based on file field.
     - Press Enter.
     - Press F12 to return to the Additional Parameters screen and make sure the format name is the one from the DSPFD command in the preparation step.
     - Press Enter.
   - Processing option 1 needs the version of P00120 from step 2 so it can find the right crossover rule records in the F0031.
   - For options 2, 3, and 4, enter the appropriate file and library names.
   - Option 5 is usually set to 1 to clear the file so just the records from the latest conversion will be there.
   - Option 6 is usually blank.
   - Leave the data selection and sequencing screens alone. They are used for the OPNQRYF statement. Since we want to use the full file, they don’t apply.

2. From the versions list, run your version.
   - There should be a R00111 in the spool files. At the bottom it will tell you how many records were read and written.
   - Do a DSSPPFM on the “to” file to see if the mapping is correct. In order to see all the data, you may want to do F10 followed by F11 to see the hexadecimal representation of the data in an over/under presentation. There will be an “F” or “D” sign in the bottom row at the right hand end of a packed field. Read that data by reading backwards from the sign in an up and down manner.
Trouble Shooting

Questions to ask yourself

Did you:

1. Set up and RUN step 2 on the menu 9841(P00120).
   - You should not change anything on the DW additional parameters.
   - You must enter the file and library names in the Proc Opt.

2. Map only the fields they want in the FROM and TO files on the crossover rules screen. **FILLER fields will be ignored and the fields will be created empty, either blanks or zeros.

3. Enter the correct format name for the based-on file(FROM) in the Additional parameters screen of the P00111 DW.
   - Check this with DSPFD of based-on file.

4. Try changing the file output type for P00111 from 1 to 2 or from 2 back to 1.
   
   Note: Sometimes LF works, sometimes OPNQYF.

5. Check the output file with a DSPPFM to see if there are any records created.

Factors to consider

1. What JD Edward World release level are you on?

2. What Step is having problems?
   - If Step 1: Check the Additional Parameters for the Based On File to be *NONE and format INONE.
   - If Step 2: Check if you ran the Dream Writer in Step 1 to ensure you are setting the action code to C when changing cross over rules.
   - If Step 3: Are you getting any records in the To file?
     - Yes: Check the cross over rules.
     - No: Use DSPFD on the from file to identify the format ID and verify the correct format is in the additional parameters of the conversion Dream Writer (P00111).

   Note: If you are on release A7.1 or above and using *DFT in the cross over rules, you needs a paper fix.

   - After making any changes to steps 2 or 3, rerun the conversion.
Things to remember

1. The all time leader in UFC errors is an incorrect format name for the Based on (input) file on the Additional Parameters screen for P00111. Do a DSPFD on the From file and go to the bottom of the display to determine the format. Option 3 from G9841.

2. Do not change anything in the additional parameters screen on the P00120 form. After entering the file names and libraries in the Processing options, you need to run the Dream Writer version. Option 1 from G9841.

3. Problems writing to output file (in P00111) may be because of duplicate keys. Check if the To file has a logical attached that has a unique key and verify the user is not populating that field with blanks.

4. File output type in P00111 Dream Writer additional parameters should be a 2 for logical: for some JD Edwards World file to JD Edwards World file conversions (DDS files), a 1 for OPNQRYF is necessary.

5. There are many conversion rules in UFC to cover most situations, try them first before using custom Xxxxx programs.

6. Use F6 to pull up full information about both fields on the Cross Over Rules screen.

7. Use reference field in the fold area of cross over for those conversion rule entries that refer to the Data Dictionary.

8. You need only map the fields that you are concerned with. If some fields are not being initialized correctly to *ZEROES or *BLANKS, you might have to find a one byte field of the right persuasion and move it into that field (longer length will be accommodated) or use *DFT.

9. The UFC does a character at a time move of the data from the input file data location into the output file data location. For this reason it is not suited to large, repeated data transfers. It is suited to one time data file conversions or small scale frequent data import situations.

10. When using *DFT if the literal is over 6 positions then part it out as if dealing with two fields.

11. When having problems with going from zoned or signed to a packed field, call it alpha to packed forcing UFC to go through the C0012 to edit the data.

12. When having problems with going from packed to alpha, try packed to signed.

13. When having problems with alpha to signed, try alpha to alpha.

14. *If all looks correct, check for multiple F0031 files, you may be executing out of the wrong cross over rules.

15. If you get a duplicate key message in the joblog, it is because there is a unique key on the physical or logical. You may not be populating one of the fields in the key and this may create the problem.

16. Set most of the FROM fields to type A for alpha.

   • Numeric TO fields can be P for packed or S for signed.

17. Sometimes P00111 will have problems with the input file.
- On the Add’l parameters screen, if the file output type is 1, change it to 2 and vice versa.

18. Make sure the P00111 based-on file and format names are correct. See Step 4(b).

19. There may be just one record in the “to file” and a message in the Joblog about a duplicate record.

- Do a DSPFD on the file and its logicals to see which one has the “unique” key requirement. You will need to map data to each of the key fields. If there is no matching data, maybe a *NN conversion rule will have to be attached to one of the key fields as a tie breaker.
Appendix E – Dates

About Date Programs

JD Edwards World uses four key RPG programs to work with dates in the system.

- X0027 is used to calculate a new date from a reference date.
- X0028 is the primary date formatting program which converts dates from Julian to Gregorian or Gregorian to Julian.
- X0035 determines the difference in days between two dates or can calculate the second date given one date and the difference in days.
- X98DAY determines the day of the week and the day and month names for a given date.

Date Formats

Julian Format

The JD Edwards World Julian (*JUL) format is CYYDDD where:

- C is the value added to 19 to create the century (0=>19, 1=>20).
- YY is the year within the century.
- DDD is the day in the year which can range from 001 to 731(next year).

For Example:

- 098185 is July 4, 1998
- 100001 is Jan 1, 2000
- 099666 is Oct 27, 2000

Gregorian Formats

The JD Edwards World Gregorian formats are *MDY, *DMY, and *YMD where:

- M represents the 2 character month
- D represents the 2 character day
- Y represents the 2 or 4 character year

Note: Edited means punctuation such as: 12/31/98, 22.11.45, or 2000-01-01.
X0027

X0027 is an RPG program the system uses to calculate a new date from a reference date including *TODAY. The interval can be + or -, the unit can be *DAY, *WEEK, *MONTH, or *YEAR.

The information here is intended to cover the usual usage of this program. The source code has samples and more documentation. Specifically, *FORMULA is used to pass a string like “*TODAY + 1”.

Using X0027

The reference date must be scrubbed (usually copy module C0012) before being passed to X0027. X0027 calls x0028 which uses the Data Dictionary item #CYR to establish the last year of the floating 100-year business window. See the X0028.

<table>
<thead>
<tr>
<th>#CYR DD default</th>
<th>Digits entered = (*MDY)</th>
<th>Julian date = CYYDDD</th>
<th>IBM QRY = (no leading 0’s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>01/ 01/ 10</td>
<td>110001 (2010)</td>
<td>110,001</td>
</tr>
<tr>
<td>10</td>
<td>12/ 31/ 11</td>
<td>011365 (1911)</td>
<td>11,365</td>
</tr>
<tr>
<td>30</td>
<td>07/ 29/ 55</td>
<td>055241 (1955)</td>
<td>55,241</td>
</tr>
<tr>
<td>60</td>
<td>07/ 29/ 55</td>
<td>155241 (2055)</td>
<td>155,241</td>
</tr>
<tr>
<td>98</td>
<td>09/ 14/ 98</td>
<td>198257 (2098)</td>
<td>198,257</td>
</tr>
<tr>
<td>97</td>
<td>09/ 14/ 98</td>
<td>098257 (1998)</td>
<td>98,257</td>
</tr>
</tbody>
</table>

Parameters

The parameters to pass to X0027 are ALL type Alpha. If the calling program is written in CL, be sure the variables are *CHAR.

The first parameter will accept a 4 digit year, for example 01011999(MMDDCCYY).

You must pass the first 8 parameters. If you want the day of the week and that day name, pass 10 parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SIDAT</td>
<td>$SIDAT is a 25 character field which can be any format as long as it agrees with the From format. To fetch today’s date use *TODAY (+/- optional) and use *FORMULA in $FFMT.</td>
</tr>
<tr>
<td>$CENT</td>
<td>$CENT is a 2 character field which should be sent blank.</td>
</tr>
<tr>
<td>$FFMT</td>
<td>$FFMT is an 8 character field describing the FROM format. The usual values will be *JUL, *MDY, *DMY, *YMD, *SYVAL or *FORMULA. See the program code for other allowed values.</td>
</tr>
</tbody>
</table>
Appendix E – Dates

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$OFFSET</td>
<td>$OFFSET is the 6 character number (+/-) of the units you have chosen in parameter 5.</td>
</tr>
<tr>
<td>$INTVL</td>
<td>$INTVL is the 8 character calendar unit used to calculate the number of days. The usual values are *DAY, *WEEK, *MONTH or *YEAR. See the program code for other allowed values.</td>
</tr>
<tr>
<td>$SODAT</td>
<td>$SODAT is a 6 character field which will contain the output date. This date will need to be passed to X0028 for any editing. It should be sent blank.</td>
</tr>
<tr>
<td>$TFMT</td>
<td>$TFMT is an 8 character field describing the TO format. The usual values will be *JUL, *MDY, *DMY, *YMD, or *SYSVAL. See the program code for other allowed values.</td>
</tr>
<tr>
<td>RTNCDE</td>
<td>RTNCDE is a single character error code. A ‘1’ will be placed in this field if X0027 receives an invalid parameter. Your program should check this field right after the call to X0027.</td>
</tr>
<tr>
<td>$DAY# (Optional)</td>
<td>$DAY# is a one character field for the day name, 1 = Sunday, 2 = Monday, and so on.</td>
</tr>
<tr>
<td>$DAY$ (Optional)</td>
<td>$DAY$ is a ten character field which, if it is sent, will have the day name loaded.</td>
</tr>
</tbody>
</table>

**X0028**

X0028 is the primary JD Edwards World date formatting program. Briefly, it is an RPG program which converts dates from Julian to Gregorian or Gregorian to Julian. It will do some editing, but that is not its purpose. This program will NOT convert edited dates to unedited.

**Using X0028**

The edited date must be scrubbed (usually copy module C0012) before being passed to X0028. The Data Dictionary item that X0028 uses is #CYR. It establishes the last year of the floating 100-year business window.

<table>
<thead>
<tr>
<th>#CYR DD default</th>
<th>Digits entered = (*MDY)</th>
<th>Julian date = CYYDDD</th>
<th>IBM QRY = (no leading 0’s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>01/ 01/ 10</td>
<td>110001 (2010)</td>
<td>110,001</td>
</tr>
<tr>
<td>10</td>
<td>12/ 31/ 11</td>
<td>011365 (1911)</td>
<td>11,365</td>
</tr>
<tr>
<td>30</td>
<td>07/ 29/ 55</td>
<td>055241 (1955)</td>
<td>55,241</td>
</tr>
<tr>
<td>60</td>
<td>07/ 29/ 55</td>
<td>155241 (2055)</td>
<td>155,241</td>
</tr>
</tbody>
</table>
Appendix E – Dates

<table>
<thead>
<tr>
<th>#CYR</th>
<th>Digits entered = (*MDY)</th>
<th>Julian date = CYYDDD</th>
<th>IBM QRY = (no leading 0’s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>09/14/98</td>
<td>198257 (2098)!</td>
<td>198,257</td>
</tr>
<tr>
<td>97</td>
<td>09/14/98</td>
<td>098257 (1998)</td>
<td>98,257</td>
</tr>
</tbody>
</table>

If the 2 digits for the year are less than or equal to #CYR, a 1 will be placed in the Julian century field so that when it is added to 19, a century of 20 will result.

**Caution:** Setting #CYR to 60 so that your business will go from 1961 to 2060 means entering a date of birth for 07/29/55 will be converted to 2055. Setting #CYR to 98 means you have a range from 1999 to 2098. Think about it!

In the prior paragraph, the stored Julian values were shown. Once the data has been stored in the file, no comparison or decision is necessary. The value of the century digit is added to 19 and the rest of the data will be used to compute the day and month. Remember, when displaying a file with QRY, the leading zero will not be displayed. DSPPFM will show you all the digits.

**Parameters**

The parameters to pass to X0028 are ALL type Alpha. If the calling program is written in CL, be sure the variables are *CHAR.

If you do not need 4 position years, pass the first 6 parameters. If you need a 4 position year, pass all 10 or 11 parameters.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$SIDAT</td>
<td>$SIDAT is a 6 character field which can be any format as long as it agrees with the From format field and has only digits from 0-9 or is all zeros. X0028 moves the converted unedited date back into this field.</td>
</tr>
<tr>
<td>#EDAT</td>
<td>#EDAT is the 8 character edited field. It has two extra positions to allow for the #SEP characters inserted by X0028. It will have the format specified by #TFMT. X0028 blanks this field upon receipt.</td>
</tr>
<tr>
<td>#FFMT</td>
<td>#FFMT is a 7 character field describing the FROM format. It can be *SYSVAL, *JUL, *MDY, *DMY, or *YMD.</td>
</tr>
<tr>
<td>#TFMT</td>
<td>#TFMT a 7 character field describing the TO format. It can be *SYSVAL, *JUL, *MDY, *DMY, or *YMD.</td>
</tr>
<tr>
<td>#SEP</td>
<td>#SEP is a 7 character field for the separator character to be used in editing the date. It can be *SYSVAL, *NONE, blank or a single editing character like ‘-’, ‘/’ or ‘.’.</td>
</tr>
</tbody>
</table>
### Parameter Description

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$ERTST</td>
<td>$ERTST is a single character error code. A ‘1’ will be placed in this field if X0028 receives an invalid date or invalid data. Your program should check this field right after the call to X0028.</td>
</tr>
<tr>
<td>$CTRY (Optional)</td>
<td>$CTRY is a two character field which will contain the century, 19 or 20. It should be sent with blanks.</td>
</tr>
<tr>
<td>#FJPN (Optional)</td>
<td>#FJPN is a one character field used for Japanese Era dates.</td>
</tr>
<tr>
<td>#TJPN (Optional)</td>
<td>#TJPN is a one character field used for Japanese Era dates.</td>
</tr>
<tr>
<td>#EDAT2</td>
<td>#EDAT2 is a 10 character field used for edited dates with 4 digits for the year. You MUST use this parameter if you want a 4 digit year date. It has two extra positions to allow for the #SEP characters inserted by X0028. It will have the format specified by #TFMT.</td>
</tr>
<tr>
<td>#SIDT2</td>
<td>#SIDT2 is an optional 8 character field intended for use with 4 digit year dates. It can be any format as long as it agrees with the From format field and has only digits from 0-9. If this parameter is used and the from format is Julian, the Julian date needs to be loaded here with 2 leading zeros and #SIDAT should be loaded with zeros. X0028 moves the converted unedited date back into this field as well as #SIDAT.</td>
</tr>
</tbody>
</table>

### X0035

X0035 is an RPG program which determines the difference in days between two dates or can calculate the second date given one date and the difference in days. It uses X0028 for date conversion.

#### Using X0035

The Data Dictionary item that X0028 uses is #CYR. It establishes the last year of the floating 100-year business window. See X0028.

#### Parameters

The parameters to pass to X0035 are ALL type Alpha. If the calling program is written in CL, be sure the variables are *CHAR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SIDA1</td>
<td>#SIDA1 is a 6 character date field which can be any format as long as it agrees with the #SIFM1 format and has only digits from 0-9. It can be sent as blanks if a reverse date is to be calculated from #SIDA2.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>#SIDA2</td>
<td>#SIDA2 is a 6 character date field which can be any format as long as it agrees with the #SIFM2 format and has only digits from 0-9. It can be sent as blanks if a forward date is to be calculated from #SIDA1.</td>
</tr>
<tr>
<td>#SIFM1</td>
<td>#SIFM1 is a 7 character field describing the #SIDA1 format. It can be *SYSVAL, *JUL, *MDY, *DMY, or *YMD.</td>
</tr>
<tr>
<td>#SIFM2</td>
<td>#SIFM2 is a 7 character field describing the #SIDA2 format. It can be *SYSVAL, *JUL, *MDY, *DMY, or *YMD.</td>
</tr>
<tr>
<td>#SITYE</td>
<td>#SITYE is a one-character field which is used when calculating the difference in days between two dates. It is usually sent blank. If the difference between two dates is to be calculated, a blank means one date is subtracted from the other. If this field is a 1, a 1 is added to the difference so that the beginning and ending date are included in the count. If this field is a 2, a 1 is subtracted from the difference which means neither date will be included in the count.</td>
</tr>
<tr>
<td>$#ERTST</td>
<td>$#ERTST is a one character field which should be sent blank. It will contain a 1 if two blank dates are sent, an invalid format is sent, or X0028 finds a problem with either date.</td>
</tr>
<tr>
<td>$#X</td>
<td>$#X is a 9 character field which contains the number of day difference to be calculated between #SIDA1 and #SIDA2. If #SIDA1 is blank, the reverse date will be placed in #SIDA1. If #SIDA2 is blank, the future date will be placed in #SIDA2.</td>
</tr>
<tr>
<td>#SICT1 (Optional)</td>
<td>#SICT1 is a 2 character century field for #SIDA1.</td>
</tr>
<tr>
<td>#SICT2 (Optional)</td>
<td>#SICT2 is a 2 character century field for #SIDA2.</td>
</tr>
</tbody>
</table>

**X98DAY**

X98DAY is an RPG program which determines the day of the week and the day and month names for a given date. It uses X0028 for date conversion.

**Using X98DAY**

The Data Dictionary item that X0028 uses is #CYR. It establishes the last year of the floating 100-year business window. See the User’s Guide for X0028.

*Note:* X98DAY only works inside this 100-year window.
Parameters

The parameters to pass to X0035 are ALL type Alpha. If the calling program is written in CL, be sure the variables are *CHAR.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>#SIDAT</td>
<td>#SIDAT is a 6 character field which can be any format as long as it agrees with the From format field and has only digits from 0-9.</td>
</tr>
<tr>
<td>#SIFMT</td>
<td>#SIFMT is a 7 character field describing the FROM format. It can be *SYSVAL, *JUL, *MDY, *DMY, or *YMD.</td>
</tr>
<tr>
<td>#SIDAY</td>
<td>#SIDAY is a 29 character field which will contain the converted day name, month name, day and year. If an ‘A’ is sent in this field, the month and day abbreviations will be used for a maximum size of 17 characters.</td>
</tr>
<tr>
<td>#SN DAY</td>
<td>#SN DAY is a one character field which will contain the number for the day of the week, 1 = Sunday, 7 = Saturday.</td>
</tr>
<tr>
<td>#SOFMT</td>
<td>#SOFMT is a 16 character field which is used to send an override output format. The edit codes are MM, DD, YY, YYYY, AM and AD where AM and AD are the abbreviated month and day names. The entries can be separated by comma, ;, :,, /, or a blank.</td>
</tr>
</tbody>
</table>
Index
<table>
<thead>
<tr>
<th>Index</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>graphic, 6-104</td>
<td></td>
</tr>
<tr>
<td>Build Transfer Library form (P98312), 3-106</td>
<td></td>
</tr>
<tr>
<td>Building blocks</td>
<td></td>
</tr>
<tr>
<td>JD Edwards World software, 1-1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>Calculation specifications, 5-6</td>
<td></td>
</tr>
<tr>
<td>Call Program form (CALL), 7-8</td>
<td></td>
</tr>
<tr>
<td>CASE profiles, 3-71</td>
<td></td>
</tr>
<tr>
<td>accessing, 3-71</td>
<td></td>
</tr>
<tr>
<td>summary, 3-75</td>
<td></td>
</tr>
<tr>
<td>CASE Profiles form (P98009), 3-72, 3-78</td>
<td></td>
</tr>
<tr>
<td>Change Library List form (CHGLIBL), 7-7</td>
<td></td>
</tr>
<tr>
<td>Changing compile option defaults for reports, 4-97</td>
<td></td>
</tr>
<tr>
<td>Changing record types, 3-37</td>
<td></td>
</tr>
<tr>
<td>Changing subfile boundaries, 4-77</td>
<td></td>
</tr>
<tr>
<td>to make larger, 4-78</td>
<td></td>
</tr>
<tr>
<td>to make smaller, 4-77</td>
<td></td>
</tr>
<tr>
<td>Changing to non-group mode, 7-6</td>
<td></td>
</tr>
<tr>
<td>Compile option defaults for reports changing, 4-97</td>
<td></td>
</tr>
<tr>
<td>Compiling a report, 4-96</td>
<td></td>
</tr>
<tr>
<td>Compiling your video, 4-60</td>
<td></td>
</tr>
<tr>
<td>Conducting a search, 4-10</td>
<td></td>
</tr>
<tr>
<td>Continue execution source debugger, 6-128</td>
<td></td>
</tr>
<tr>
<td>Control fields user space, 6-116</td>
<td></td>
</tr>
<tr>
<td>Control Files Copy form (P924127), 3-103</td>
<td></td>
</tr>
<tr>
<td>Control parameters, 6-89</td>
<td></td>
</tr>
<tr>
<td>Control specifications, 5-4</td>
<td></td>
</tr>
<tr>
<td>Conventions naming, 3-52</td>
<td></td>
</tr>
<tr>
<td>Conversion forms creating, 8-31</td>
<td></td>
</tr>
<tr>
<td>Conversion rule program, 8-22</td>
<td></td>
</tr>
<tr>
<td>fifth parameter, 8-22</td>
<td></td>
</tr>
<tr>
<td>first parameter, 8-22</td>
<td></td>
</tr>
<tr>
<td>fourth parameter, 8-22</td>
<td></td>
</tr>
<tr>
<td>second parameter, 8-22</td>
<td></td>
</tr>
<tr>
<td>third parameter, 8-22</td>
<td></td>
</tr>
<tr>
<td>Copy DD, VO, DW, UDC, SVR, Menus form (P99630), 3-22, 3-111</td>
<td></td>
</tr>
<tr>
<td>Copy File form (CPYF), 3-14, 3-20</td>
<td></td>
</tr>
<tr>
<td>Copy modules, 3-63</td>
<td></td>
</tr>
<tr>
<td>COPY modules functional servers, 6-111</td>
<td></td>
</tr>
<tr>
<td>Copying a file, 3-19</td>
<td></td>
</tr>
<tr>
<td>Copying a library, 3-18</td>
<td></td>
</tr>
<tr>
<td>Copying a record, 3-20</td>
<td></td>
</tr>
<tr>
<td>Copying data to your development environment, 3-18</td>
<td></td>
</tr>
<tr>
<td>Copying JD Edwards World record types, 3-21</td>
<td></td>
</tr>
<tr>
<td>Create Library form (CRTLIB), 3-14</td>
<td></td>
</tr>
<tr>
<td>Create New Screen form (P92510), 4-51</td>
<td></td>
</tr>
<tr>
<td>Create Source Physical File form (CRTSRCPF), 3-16</td>
<td></td>
</tr>
<tr>
<td>Creating a development environment, 3-5</td>
<td></td>
</tr>
<tr>
<td>Creating a development source library, 3-14</td>
<td></td>
</tr>
<tr>
<td>Creating conversion forms, 8-31</td>
<td></td>
</tr>
<tr>
<td>Creating development object libraries, 3-13</td>
<td></td>
</tr>
<tr>
<td>Creating JDESRC with JD Edwards World program generator, 3-14</td>
<td></td>
</tr>
<tr>
<td>Creating JDESRC without the program generator, 3-16</td>
<td></td>
</tr>
<tr>
<td>Creating libraries, 3-11</td>
<td></td>
</tr>
<tr>
<td>Creating new job groups, 7-4</td>
<td></td>
</tr>
<tr>
<td>Creating record type codes, 3-39</td>
<td></td>
</tr>
<tr>
<td>Creating the transfer library, 3-105</td>
<td></td>
</tr>
<tr>
<td>Creating user space and user index, 6-111</td>
<td></td>
</tr>
<tr>
<td>Cross Reference form (P980014), 3-66</td>
<td></td>
</tr>
<tr>
<td>Crossover rules working with, 8-15</td>
<td></td>
</tr>
<tr>
<td>Crossover Rules form (O031), 8-15, 8-19</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>D defined, 4-68</td>
<td></td>
</tr>
<tr>
<td>Data dictionary security, 4-21</td>
<td></td>
</tr>
<tr>
<td>user defined help instructions, 4-26</td>
<td></td>
</tr>
<tr>
<td>working with, 4-16</td>
<td></td>
</tr>
<tr>
<td>Data Dictionary form (P9201), 4-16, 8-33</td>
<td></td>
</tr>
<tr>
<td>Data dictionary glossary groups, 4-22</td>
<td></td>
</tr>
<tr>
<td>working with, 4-22</td>
<td></td>
</tr>
<tr>
<td>working with by file, 8-33</td>
<td></td>
</tr>
<tr>
<td>Data dictionary repository, 4-13</td>
<td></td>
</tr>
<tr>
<td>Data dictionary structure data field display text, 4-14</td>
<td></td>
</tr>
<tr>
<td>data field specifications, 4-14</td>
<td></td>
</tr>
<tr>
<td>data item aliases, 4-15</td>
<td></td>
</tr>
<tr>
<td>data item alpha description, 4-14</td>
<td></td>
</tr>
<tr>
<td>data item master, 4-14</td>
<td></td>
</tr>
<tr>
<td>error message program ID, 4-15</td>
<td></td>
</tr>
<tr>
<td>glossary text file, 4-15</td>
<td></td>
</tr>
<tr>
<td>key index file, 4-15</td>
<td></td>
</tr>
<tr>
<td>understanding, 4-13</td>
<td></td>
</tr>
<tr>
<td>Data Field Alias form (P9204), 4-22</td>
<td></td>
</tr>
<tr>
<td>Data field descriptions working with, 4-27</td>
<td></td>
</tr>
<tr>
<td>Data Field Descriptions form (P9202), 4-27</td>
<td></td>
</tr>
<tr>
<td>Data file design aid, 4-33</td>
<td></td>
</tr>
<tr>
<td>entering, 4-35</td>
<td></td>
</tr>
<tr>
<td>Data File Design Aid form (P92102), 4-35, 4-38</td>
<td></td>
</tr>
<tr>
<td>Data file design aid summary, 4-41</td>
<td></td>
</tr>
<tr>
<td>Data integrity logic, 6-106</td>
<td></td>
</tr>
<tr>
<td>Data item alias revisions, 4-21</td>
<td></td>
</tr>
<tr>
<td>Data Item Glossary Revisions form (P92001), 4-24, 4-26, 8-34</td>
<td></td>
</tr>
<tr>
<td>Data item name locating, 4-15</td>
<td></td>
</tr>
</tbody>
</table>
Data Item Size defined, 4-68
Data Item Type defined, 4-68
Data modeling accessing, 4-3
  selection exits, 4-6
  working with, 4-3
Data Modeling form (P98042), 4-3, 4-5
Database considerations, 8-5
Database fields
  selecting, 4-66
Debugger
  using with interactive program, 6-123
  with batch program, 6-126
Define
  project, 3-106
  promotion path, 3-106
  promotion path for control tables, 3-91
  promotion path for source code members, 3-89
Defining a project, 3-92
Defining a promotion path, 3-87
Defining access for a user profile using J98INIT, 3-17
Defining access for a user profile using J98INITA, 3-17
Defining record type titles, 3-41
Deleting records, 8-20
Description
  defined, 4-51, 4-68
Design Aid Exit/Save form (P92590), 4-59
Designing the report, 4-89
Detail Spec. Over Titles form (P48002), 3-37
Detail Spec. Types form (P00051), 3-36
Detailed explanation of a line, 4-5
Determining
  program environment, 6-124
Developer's Workbench form (G9362), 3-80
Development cycle, 1-1
Development environment
  copying data to, 3-18
Development environments, 3-5
  rules for creating, 3-5
Direction
  defined, 4-5
Displaying field descriptions, 8-18
  documentation, 6-6
Domestic Field Begin Pos defined, 8-17
Domestic Field Data Type defined, 8-17
Domestic Field Decimal Positions defined, 8-17
Domestic Field Description defined, 8-17
Domestic Field Name defined, 8-16
Domestic Field Size In Bytes defined, 8-17
Domestic File Name defined, 8-16
Domestic Number of Digits defined, 8-17

E
Edit and Promote form (P92412), 3-97
Editing commands, 4-45
End debug (ENDDBG), 6-132
ENDDBG (end debug), 6-132
Entry program
  setting up business rules, 6-103
Error handling, 6-3
Error message
  index line (COORIX), 6-117
Examples
  production and development, 3-8
Executing
  program being debugged, 6-125
  source debugger, 6-127
Existing field
  updating, 4-48
Exit
  function key, 4-48
Extension specifications, 5-5

F
Fast path create
  accessing for a new form, 4-51
Fast Path Create
  defined, 4-51
Fast Path File
  defined, 4-65, 4-94
Features
  system integration, 2-1
Field definition
  revising, 4-80
Field Definition report (P928400), 4-91
Field definition window
  report design aid vs. screen design aid, 4-84
Field descriptions
  displaying, 8-18
Field in RDA
  updating, 4-90
Field Name
  defined, 4-68
Field name standards, 4-46
Field reference file, 4-29
  rebuild, 4-30
Field Selection List form (P92524), 4-67, 4-69
Fields
  #Fields Selected, 4-65
  Action Code, 4-51
<table>
<thead>
<tr>
<th>add hidden, 4-75</th>
</tr>
</thead>
<tbody>
<tr>
<td>adding, 8-19</td>
</tr>
<tr>
<td>adding without using pick list, 4-54</td>
</tr>
<tr>
<td>Based On File, 4-40</td>
</tr>
<tr>
<td>cover page</td>
</tr>
<tr>
<td>graphic, 4-85</td>
</tr>
<tr>
<td>D, 4-68</td>
</tr>
<tr>
<td>Data Item Size, 4-68</td>
</tr>
<tr>
<td>Data Item Type, 4-68</td>
</tr>
<tr>
<td>Description, 4-51, 4-68</td>
</tr>
<tr>
<td>Direction, 4-5</td>
</tr>
<tr>
<td>Domestic Field Begin Pos, 8-17</td>
</tr>
<tr>
<td>Domestic Field Data Type, 8-17</td>
</tr>
<tr>
<td>Domestic Field Decimal Positions, 8-17</td>
</tr>
<tr>
<td>Domestic Field Description, 8-17</td>
</tr>
<tr>
<td>Domestic Field Name, 8-16</td>
</tr>
<tr>
<td>Domestic Field Size In Bytes, 8-17</td>
</tr>
<tr>
<td>Domestic File Name, 8-16</td>
</tr>
<tr>
<td>Domestic Number of Digits, 8-17</td>
</tr>
<tr>
<td>Fast Path Create, 4-51</td>
</tr>
<tr>
<td>Fast Path File, 4-65, 4-94</td>
</tr>
<tr>
<td>Field Name, 4-68</td>
</tr>
<tr>
<td>File Description, 4-34, 4-40</td>
</tr>
<tr>
<td>Fld Pfx, 4-65, 4-94</td>
</tr>
<tr>
<td>Foreign Field Name, 8-16</td>
</tr>
<tr>
<td>Foreign File Name, 8-16</td>
</tr>
<tr>
<td>Format Name, 4-65, 4-94</td>
</tr>
<tr>
<td>HDG, 4-68</td>
</tr>
<tr>
<td>Logical Files, 4-41</td>
</tr>
<tr>
<td>Opt, 4-64, 4-94</td>
</tr>
<tr>
<td>Ordering of Fields, 4-41</td>
</tr>
<tr>
<td>OVERLAY, 4-52</td>
</tr>
<tr>
<td>PUTOVR, 4-52</td>
</tr>
<tr>
<td>Quantifier, 4-5</td>
</tr>
<tr>
<td>Related Record, 4-65, 4-94</td>
</tr>
<tr>
<td>report header</td>
</tr>
<tr>
<td>graphic, 4-85</td>
</tr>
<tr>
<td>Selection Exits, 4-52</td>
</tr>
<tr>
<td>Seq No, 4-67</td>
</tr>
<tr>
<td>Start/ End Lines, 4-65, 4-94</td>
</tr>
<tr>
<td>Subfile, 4-51</td>
</tr>
<tr>
<td>Subfile Clear, 4-52</td>
</tr>
<tr>
<td>Subfile Fold, 4-52</td>
</tr>
<tr>
<td>Subfile portion of screen, 4-6</td>
</tr>
<tr>
<td>Type, 4-6, 4-65, 4-94</td>
</tr>
<tr>
<td>Unique Keys, 4-40</td>
</tr>
<tr>
<td>Use, 4-68</td>
</tr>
<tr>
<td>Version, 8-16</td>
</tr>
<tr>
<td>Wide Screen, 4-51</td>
</tr>
<tr>
<td>Window, 4-51</td>
</tr>
<tr>
<td>work, 6-7</td>
</tr>
<tr>
<td>File conversion</td>
</tr>
<tr>
<td>working with, 8-25</td>
</tr>
<tr>
<td>File Description</td>
</tr>
<tr>
<td>defined, 4-34, 4-40</td>
</tr>
<tr>
<td>File description specifications, 5-4</td>
</tr>
<tr>
<td>File prefix, 4-34</td>
</tr>
<tr>
<td>File Selection List form (P92522), 4-69</td>
</tr>
</tbody>
</table>

File servers, 6-87
advantages, 6-88
commonly used, 6-99
disadvantages, 6-88
how does it function, 6-89
implementing, 6-92
tips when using, 6-94
types, 6-88
what is a, 6-87

Fld Pfx defined, 4-65, 4-94

Flow of typical program
graphic, 6-102

Foreign Field Name defined, 8-16

Foreign File Name defined, 8-16

Format Keyword Maintenance form (P92537), 4-72

Format Name defined, 4-65, 4-94

Forms
Add Cross Over Instructions, 8-20
Build Transfer Library, 3-106
Call Program, 7-8
CASE Profiles, 3-72, 3-78
Change Library List, 7-7
Control Files Copy, 3-103
Copy DD, VO, DW, UDC, SVR, Menus, 3-22, 3-111
Copy File, 3-20
Copy File (CPYF), 3-14
Create Library (CRTLIB), 3-14
Create New Screen, 4-51
Create Source Physical File (CRTSRCPF), 3-16
Cross reference, 3-66
Crossover Rules, 8-15, 8-19
Data Dictionary, 4-16, 8-33
Data Field Alias, 4-22
Data Field Descriptions, 4-27
Data file design Aid, 4-35
Data File Unit, 4-38
Data Item Flossary Revisions, 4-24, 4-26
Data Item Glossary Revisions, 8-34
Data Modeling, 4-3, 4-5
Design Aid Exit/ Save, 4-59
Detail Spec. Over Titles, 3-37
Detail Spec. Types, 3-36
Developer's Workbench, 3-80
Edit and Promote, 3-97
field Selection List, 4-67, 4-69
File Selection List, 4-69
Format Keyword Maintenance, 4-72
FRF & JD Edwards World Msg File, 4-31
Function Key/ Opt Definition, 4-73
Item Maintenance, 4-52, 4-53
Item Master Information, 4-54, 4-56, 4-57, 4-58, 4-70, 4-76, 4-78, 4-79
Item Search, 4-47, 4-48, 4-65
JD Edwards World Passthru Network, 7-17
JD Edwards World Visual Debug, 6-125, 6-127, 6-129
Library List Revisions, 3-17
List of Defined Fields, 4-75
Load Transferred Library, 3-109
Manage Program Paths, 3-87
Manage Projects, 3-93
Next Numbers, 4-28
Next Numbers by Company/Fiscal, 4-29
Object Cross Reference Repository, 4-9, 4-10, 4-11
Pre Promotion Edit Details, 3-100
Pre Promotion Edit History, 3-100
Print Transfer Report, 3-107
Processing Options Revisions, 7-16, 8-10, 8-25, 8-31, 8-34
Project Elements, 3-95
Project Promotion Paths, 3-94, 3-99
Promotion Path, 3-88
Promotion Path Control Files, 3-91
Promotion Path Members, 3-89
Record Formats List, 4-64, 4-66, 4-68
Record Type Titles, 3-41
Remove Member (RMVM), 3-15
Restore Library, 3-107
SAR Log Inquiry, 3-81
SAR Log Transfer, 3-98
Save Library, 3-107
Single D/B Relation Entry, 4-62
Software Scan and Replace, 6-133
Software Transfer, 3-103
Software Versions Repository, 3-45, 3-52, 3-62, 3-63, 3-64, 3-65, 3-72, 3-81, 4-34, 4-60, 4-90, 4-96, 6-124
Start Pass-Through, 7-17
Universal File Converter, 8-8, 8-25
User Defined Code Revisions, 3-40
User Defined Codes Window, 4-11
User Information, 3-18, 7-3, 7-11
User Signon List Revisions, 3-17
Versions List, 8-9
Versions Setup, 7-15
FRF & JD Edwards World Msg File form (P98FRF), 4-31
Function key
  exits from screen design aid, 4-70
Function key exits
  return to previous panel, 4-48
Function key exits from screen design aid, 4-70
Function Key/ Opt Definition form (P9601), 4-73
Function keys, 3-66
data dictionary, 4-21
  flow program/ illustrate file models, 3-67
  function key/ opt definition, 4-73
JD Edwards World command line, 4-70
subfile drop, 4-71
toggle monochrome/ color display, 4-72
Functional server
  interactive program cycle, 6-112
  Functional server error messages
    graphic, 6-110
  Functional server highlights, 6-104
  Functional server interface, 6-110
    graphic, 6-109
  Functional server parameters, 6-110
  Functional server program sections
    graphic, 6-117
  Functional server transaction data
    graphic, 6-109
  Functional server user index, 6-111
  Functional server user space, 6-110
  Functional server/ COPY modules, 6-111
Functional servers, 6-101
  advantages, 6-102
  call parameters, 6-113
  COPY modules, 6-111
  disadvantages, 6-103
  highlights, 6-104
  how they function, 6-103
  parameters, 6-110
  user indexes, 6-111
  user space, 6-110
  what are they, 6-101
Functions
  navigation, 3-64
Functions and options
  delete records, 8-24

G
General aesthetics, 4-62
  alignment, 4-62
  grouping fields, 4-62
  spacing, 4-62
Graphics
  project attributes, 3-86
Group job window
  accessing, 7-4
  advanced functions, 7-8
  entering commands, 7-9
  hidden selections, 7-8
  working with, 7-3
Group job window summary, 7-10
Group jobs, 7-1
  activating suspended, 7-5
  signing off with suspended, 7-6
  terminating, 7-6
  working with non-JD Edwards World, 7-7
Guidelines, 3-92, 6-6
  scan and replace, 6-134
H
HDG
defined, 4-68
Hidden fields, 4-75
How does a file server function?, 6-89
How does a functional server function?, 6-103
How does a user index function?, 6-58

I
IBM pass-through
setting up access to remote locations, 7-15
using with group jobs, 7-16
working with, 7-15
Identifiers
file information, 3-48
maintenance on a logical file, 3-48
member, 3-45
member relationship and compiling information, 3-47
processing a screen, 3-48
type, use, and associated systems, 3-46
where members are maintained, 3-49
Implementing a file server, 6-92
Index line (COORIX)
error message, 6-117
Initiating
source debugger, 6-124, 6-126
Input specifications, 5-5
Interactive program cycle using a functional server, 6-112
Internal RPG subroutines within
JD Edwards World programs, 6-11
Item description
graphic, 4-88
Item Maintenance form (P92700), 4-52, 4-53
Item Master Information form (P928011), 4-54, 4-56, 4-57, 4-58, 4-70, 4-76, 4-78, 4-79
Item Search form (P928200), 4-47, 4-48, 4-65

J
J98INIT
defining access for a user profile, 3-17
J98INITA
defining access for a user profile, 3-17
JD Edwards World group job window summary, 7-10
JD Edwards World open application architecture
graphic, 6-108
JD Edwards World Passthru Network form (B98P), 7-17
JD Edwards World Visual Debug form (P3701), 6-125
JD Edwards World Visual Debug form (P93701), 6-127, 6-129
Job groups
creating new, 7-4
Join logical files, 3-63

K
Key lists
searching for, 6-93
Keywords, 8-21
business unit, 8-21
check data dictionary, 8-21
data dictionary default, 8-21
dates, 8-21
default constant, 8-21
initialize, 8-21
next number, 8-21
user defined code lookup, 8-21

L
Libraries
CLTSEC, 3-7
copying, 3-18
creating, 3-6, 3-11
creating common, 3-11
creating development object, 3-13
data, 3-7
install, 3-7
JD Edwards World, 3-6
JDEINSTAL, 3-7
JDFDATA, 3-7
JDFINS, 3-7
JDOBJ, 3-7
JDFSRC, 3-6
object, 3-7
plans, 3-7
security, 3-7
source, 3-6
understanding development source, 3-13
Library List Revisions form (P0094), 3-17
List of Defined Fields form (P92540), 4-75
Literal field
adding, 4-56
Load Transferred Library form (P98312), 3-109
Loading the transfer library, 3-109
Locate
a project, 3-93
promotion path, 3-88
Locating a data item name, 4-15
Locating the next numbers facility, 4-28
Locating the rebuild FRF and JD Edwards World Msg file form, 4-31
Logic
data integrity, 6-106
user interface, 6-106
Logical file
sample, 4-37
Logical files, 3-62
join, 3-63
Logical Files
defined, 4-41
Logical files with omits
sample, 4-38
Logical files with selects
sample, 4-38

M
Manage Projects form (P92413), 3-93
Manage Promotion Paths form (P92403), 3-87
Member identifiers, 3-45
Menu flow, 4-34
Merge functions for PTFs and reinstalls, 4-41
Message file
JD Edwards World, 4-30
locating rebuild FRF & JD Edwards World Msg, 4-31
rebuilding, 4-30
Miscellaneous items, 6-7
key list (KLIST), 6-7
naming conventions, 6-7
Modules
copy, 3-63

N
Naming conventions, 3-52
Naming conventions for objects, 3-54
Next Numbers by Company/Fiscal form (P00021), 4-29
Next numbers facility
locating, 4-28
working with, 4-27
working with by company & fiscal year, 4-29
Next Numbers form (P0002), 4-28
Non-group mode
changing to, 7-6

O
Object cross reference repository
working, 4-9
Object Cross Reference Repository form (P980014), 4-9, 4-10, 4-11
Open application architecture
graphic, 6-108
Opt
defined, 4-64, 4-94
Optional files workbench, 3-62
Ordering of Fields
defined, 4-41
Output specifications, 5-7
OVERLAY
defined, 4-52

P
Parameters
call, for the functional server, 6-113
control, 6-89
functional servers, 6-110
returned, 6-92
Performance issues, 6-135
Pick list
placing fields on a form using, 4-70
Placing fields on a form using a pick list, 4-70
Pre Promotion Edit Details form (P92431), 3-100
Pre Promotion Edit History form (P9243), 3-100
Prefix standards, 4-45
Prepare
SAR system, 3-106
Print Transfer Report form (P98312), 3-107
Printing a report, 8-29
Printing the transfer library report, 3-107
Process overview - placing selected fields, 4-78
Process overview - revising the field definition, 4-79
Processing Options Revisions form (P98312), 7-16, 8-10, 8-25, 8-31, 8-34
Production and development examples, 3-8
basic development environment, 3-9
basic production environment, 3-8
common shared library, 3-9
no source in production environment, 3-9
one source and object library, 3-10
Profiles
defining access for using J98INIT, 3-17
understanding user, 3-16
Program and file names, 3-61
Program being debugged
executing, 6-125
Program environment
determining, 6-124
Program generator
creating JDESRC, 3-14
creating JDESRC without, 3-16
Program overview, 6-1
Program specifications, 5-3
Program Structure, 6-11
Programs and IDs
0031 (crossover rules), 8-15, 8-19
00312 (add cross over instructions), 8-20
B98P (JD Edwards World passthru network), 7-17
CALL (call program), 7-8
CHGLIBL (change library list), 7-7
CPYF (copy file), 3-14, 3-20
CRTLIB (create library), 3-14
CRTSRC (create source physical file), 3-16
G9362 (developer’s workbench), 3-80
G9841 (universal file converter), 8-8, 8-25
P0002 (next numbers), 4-28
P00021 (next numbers by company/fiscal), 4-29
P00051 (detail spec. types), 3-36
P00051 (user defined code revisions), 3-40
P0092 (user signon list revisions), 3-17
P0095 (language list revisions), 3-17
P08332 (single d/b relation entry), 4-62
P3701 (JD Edwards World visual debug), 6-125
P48002 (detail spec. title type), 3-37
P48002 (record type titles), 3-41
P81QM (user defined codes window), 4-11
P90630 (Copy DD, VO, DW, UDC, SVR, Menus), 3-111
P92001 (data item glossary revisions), 4-24, 4-25, 8-34
P9201 (data dictionary), 4-16, 8-33
P9202 (data field descriptions), 4-27
P9204 (data field alias), 4-22
P92102 (data file design aid), 4-35, 4-38
P9240 (promotion path), 3-88
P92401 (promotion path members), 3-89
P92402 (promotion path control files), 3-91
P92403 (manage program paths), 3-87
P92411 (project promotion paths), 3-94
P92411W (project promotion paths), 3-99
P92412 (project elements), 3-95
P92412 (edit and promote), 3-97
P924124 (software transfer), 3-103
P924127 (control files copy), 3-103
P92413 (manage projects), 3-93
P9242 (SAR log transfer), 3-98
P9243 (pre promotion edit history), 3-100
P92431 (pre promotion edit details), 3-100
P924310 (create new screen), 4-51
P924320 (record formats list), 4-64, 4-66, 4-68
P92422 (file selection list), 4-69
P92424 (field selection list), 4-67, 4-69
P92437 (format keyword maintenance), 4-72
P92450 (list of defined fields), 4-75
P92450 (design aid exit/ save), 4-59
P92700 (item maintenance), 4-52, 4-53
P928011 (item master information), 4-54, 4-56, 4-57, 4-58, 4-70, 4-76, 4-78, 4-79
P928200 (item search), 4-47, 4-48, 4-65
P928400 (field definition report), 4-91
P93701 (JD Edwards World visual debug), 6-127, 6-129
P9601 (function key/ opt definition), 4-73
P980014 (cross reference), 3-66
P980014 (Object Cross Reference Repository), 4-9, 4-10, 4-11
P98009 (CASE profiles), 3-78
P98009 (CASE Profiles), 3-72
P9801 (software versions repository), 3-45, 3-52, 3-62, 3-63, 3-64, 3-65, 3-72, 3-81, 4-34, 4-60, 4-90, 4-96, 6-124
P98042 (data modeling), 4-3, 4-5
P9810 (SAR log inquiry), 3-81
P98300 (versions list), 8-9
P98300 (versions setup), 7-15
P98312 (build transfer library), 3-106
P98312 (load transferred library), 3-109
P98312 (print transfer report), 3-107
P98312 (processing options revisions), 7-16, 8-10, 8-25, 8-31, 8-34
P98810 (software scan and replace), 6-133
P98FRF (FRF & JD Edwards World Msg file), 4-31
P99630 (copy DD, VO, DW, UDC, SVR, Menus), 3-22
RMVM (remove member), 3-15
RSTLIB (restore library), 3-107
SAVLIB (save library), 3-107
STRPASTHR (start pass-through), 7-17
Project
define, 3-106
defining, 3-92
promote, 3-106
promoting, 3-97, 3-102
to locate, 3-93
to promote, 3-102
Project Elements form (P92412), 3-95
Project Promotion Paths form (P92411), 3-94
Project Promotion Paths form (P92411W), 3-99
Project SABs
to assign, 3-95
Project updates
promoting, 3-105
Promote
project, 3-102, 3-106
Promoting a project, 3-97, 3-102
Promoting project updates, 3-105
Promotion path
define, 3-106
validating, 3-106
validating, 3-99
Promotion Path Control Files form (P92402), 3-91
Promotion Path form (P9240), 3-88
Promotion Path Members form (P92401), 3-89
Promotion paths
defining, 3-87
to add, 3-88
to assign, 3-94
to define for control tables, 3-91
to define for source code members, 3-89
to locate, 3-88
to validate, 3-99
understanding, 3-85
Promotion paths and projects
working with, 3-85
PTFs
merge functions, 4-41
PUTOVR
defined, 4-52
Q
Quantifier
defined, 4-5

R
Reading from a user space, 6-54
Record formats, 4-65
Record Formats List form (P92520), 4-64, 4-66, 4-68
Record type codes
creating, 3-39
Record type titles
defining, 3-41
verifying, 3-42
Record Type Titles form (P48002), 3-41
Record types
adding, 3-36
changing, 3-37
copying, 3-21
Records
copying, 3-20
deleting, 8-20
Reinstall
merge functions, 4-41
Related Record
defined, 4-65, 4-94
Remove Member form (RMVM), 3-15
Report
compiling, 4-96
printing, 8-29
scan and replace, 6-134
Report design aid, 4-83
accessing, 4-90
Report design standards, 4-88
general aesthetics, 4-89
JD Edwards World standards/ record formats, 4-88
RDA features, 4-88
Report formats, 4-86
Reports
Field Definition, 4-91
Print Install Records, 3-109
Software Scan and Replace, 6-134
Universal File Converter, 8-29
Restore Library form (RSTLIB), 3-107
Restoring the transfer library from tape, 3-107
Retrieving data from a user index, 6-64
Returned parameters, 6-92
Review an RPG program's source, 6-16
RPG program's source
review, 6-16
Rules for creating development environments, 3-5
selecting types to log, 3-79
SAR log, 3-77
SAR log inquiry
accessing, 3-80
selection exits, 3-82
summary, 3-83
SAR Log Inquiry form (P9810), 3-81
SAR Log Transfer form (P9242), 3-98
SAR logging
setting up user input options, 3-78
SAR system
setup, 3-39
to prepare, 3-106
SARs
updating, 3-97
Save Library form (SAVLIB), 3-107
Saving the transfer library to tape, 3-107
Scan and replace, 6-134
guidelines, 6-134
report, 6-134
working with, 6-133
Screen design aid, 4-43
summary, 4-82
working with, 4-47
Screen design standards and tips, 4-60
alpha fields, 4-61
default cursor, 4-61
description fields, 4-61
fold area, 4-61
line 24, 4-60
title, 4-60
window, 4-61
SDA Exit/ save function key, 4-59
Search
conducting, 4-9, 4-10
Searching for key lists, 6-93
Selected fields
placing, 4-78
Selecting database fields, 4-66
Selecting types of SAR information to log, 3-79
Selection exits
data modeling, 4-6
fields, 4-7
software versions repository, 3-67
where used, 4-7
Selection Exits
defined, 4-52
Selection exits from the scheduling workbench, 3-36
processing options, 3-36
Seq No
defined, 4-67
Setting the break point
source debugger, 6-127
Setting up universal file converter, 8-8
Setting up user input options for SAR logging, 3-78
Signing off with suspended group jobs, 7-6
SAR information
Single D/B Relation Entry form (P08332), 4-62
Software Versions Repository form (P9801), 3-81
Software Scan and Replace form (P98810), 6-133
Software Scan and Replace report (P98810), 6-134
Software Transfer form (P924124), 3-103
Software versions repository
accessing, 3-45
selection exits, 3-67
Software Versions Repository
working with, 3-43
Software Versions Repository form (P9801), 3-45, 3-52, 3-62, 3-63, 3-64, 3-65, 3-72, 4-34, 4-60, 4-90, 4-96, 6-124
Source debugger, 6-123
continue execution, 6-128
execute the program, 6-127
features, 6-129
initiating, 6-124, 6-126
set the break point, 6-127
Standards
field name, 4-46
prefixes, 4-45
screen design, 4-60
Start Pass-Through form (STRPASTHR), 7-17
Start/End Lines
defined, 4-65, 4-94
Subfile
defined, 4-51
Subfile boundaries
changing, 4-77
Subfile Clear
defined, 4-52
Subfile Fold
defined, 4-52
Subfile portion of screen
defined, 4-6
Subroutines, 6-1
common, 6-2
internal RPG, within JD Edwards World programs, 6-11
standard, 6-1
Summary of CASE profiles, 3-75
Summary of JD Edwards World attention MENU window functions, 7-12
Summary of screen design aid, 4-82
Summary of the SAR log inquiry, 3-83
Suspended group jobs
signing off with, 7-6
System integration, 2-1
features, 2-1

T
Terminating group jobs, 7-6
Tips when using file servers, 6-94
To work with software scan and replace, 6-133
Tracking information if writing variable length records, 6-54
Traditional architecture
graphic, 6-105, 6-107
Transferring individual control table records, 3-111
Type
defined, 4-6, 4-65, 4-94

U
Understanding development source libraries, 3-13
Understanding promotion paths, 3-85
Understanding the data dictionary structure, 4-13
Understanding the SDA exit/save function key, 4-59
Understanding the select all function key, 4-67
Understanding the universal file converter setup, 8-7
Understanding user profiles, 3-16
Unique Keys
defined, 4-40
Universal building blocks of JD Edwards World software, 1-1
Universal file converter, 8-1
business unit, 8-4
calculation of a value, 8-4
data dictionary, 8-4
data dictionary structure, 8-1
data dictionary definition, 8-4
data dictionary generation, 8-4
data dictionary default, 8-4
database considerations, 8-5
dates, 8-4
default constant, 8-4
graphic, 8-2
initialization, 8-4
next number, 8-4
numeric fields, 8-4
setting up, 8-7, 8-9
special processing, 8-4
understanding set up, 8-7
user defined code lookup, 8-4
user responsibilities, 8-5
Universal File Converter form (G9841), 8-8, 8-25
Universal File Converter report (P98300), 8-29
Updating a field in RDA, 4-90
Updating an existing field, 4-48
Updating the SARs, 3-98
Updating/adding fields through SDA, 4-47
Usage
indicator, 6-4
Use
defined, 4-68
User Defined Code Revisions form (P00051), 3-40
User Defined Codes Window form (P81QM), 4-11
User defined help instructions
data dictionary, 4-26
User indexes, 6-57
accessing, 6-112
advantages, 6-58
creating, 6-59, 6-111
Index

functional servers, 6-111
how does it function, 6-58
retrieving data from, 6-64
writing to, 6-62
User Information form (P0092), 3-18, 7-3, 7-11
User interface logic, 6-106
User responsibilities, 8-5
User Signon List Revisions form (P0094), 3-17
User space, 6-49
accessing, 6-112
advantages, 6-49
control fields, 6-116
creating, 6-50, 6-111
functional servers, 6-110
how it functions, 6-50
reading from, 6-54
writing to, 6-53
Using *ALL, 4-57
Using *BOTH, 4-56
Using debugger with an interactive program, 6-123
Using IBM pass-through with group jobs, 7-15

V

Validate
promotion path, 3-99, 3-106
Validating a promotion path, 3-99
Variable length records
tracking information, 6-54
Version
defined, 8-16
Version control, 3-1
Versions List form (P98300), 8-9
Versions Setup form (P98300), 7-15
Video
compiling, 4-60
Video fields
adding, 4-64
record formats list, 4-64

W

W.O. Detail form
accessing, 3-26
What are calculation specifications?, 5-6
What are control specifications?, 5-4
What are extension specifications?, 5-5
What are file description specifications?, 5-4
What are functional servers?, 6-101
What are input specifications?, 5-5
What are output specifications?, 5-7
What are the advantages of using a file server?, 6-88
What are the advantages of using a functional server?, 6-102
What are the advantages of using a user index?, 6-58
What are the advantages of using a user space?, 6-49
What are the disadvantages of using a file server?, 6-88
What are the disadvantages of using a functional server?, 6-103
What are the report design standards, 4-88
What is a file server?, 6-87
What is a user space?, 6-49
Wide Screen
defined, 4-51
Window
defined, 4-51
Windows, 3-64
Work fields, 6-7
option files, 6-8
Work with non-JD Edwards World group jobs, 7-7
Workbench
optional files, 3-62
Working with data field descriptions, 4-27
Working with data item alias revisions, 4-22
Working with data modeling, 4-3
Working with file conversion, 8-25
Working with next number by company and fiscal year, 4-29
Working with object cross reference repository, 4-9
Working with promotion paths and projects, 3-85
Working with screen design aid, 4-47
Working with Software Versions Repository, 3-43
Working with the crossover rules form, 8-15
Working with the data dictionary, 4-16
Working with the data dictionary glossary, 4-22
Working with the data dictionary glossary by file, 8-33
Working with the next numbers facility, 4-27
Working with user defined help instructions, 4-26
Writing to a user space, 6-53