ORACLE
JD EDWARDS ENTERPRISEONE

JD Edwards EnterpriseOne Tools 8.96 System Administration Guide

April 2006

ORACLE

JD Edwards EnterpriseOne Tools 8.96 System Administration Guide
SKU E1_TOOLS896TSA-B 0406

Contents

About This Documentation Preface

JD Edwards EnterpriseOne implementation guides provide you with the information that you need to implement and use JD Edwards EnterpriseOne applications from Oracle.

This preface discusses:

- JD Edwards EnterpriseOne application prerequisites.
- Application fundamentals.
- Documentation updates and printed documentation.
- Additional resources.
- Typographical conventions and visual cues.
- Comments and suggestions.
- Common fields in implementation guides.

Note. Implementation guides document only elements, such as fields and check boxes, that require additional explanation. If an element is not documented with the process or task in which it is used, then either it requires no additional explanation or it is documented with common fields for the section, chapter, implementation guide, or product line. Fields that are common to all JD Edwards EnterpriseOne applications are defined in this preface.

JD Edwards EnterpriseOne Application Prerequisites

To benefit fully from the information that is covered in these books, you should have a basic understanding of how to use JD Edwards EnterpriseOne applications.

You might also want to complete at least one introductory training course, if applicable.

You should be familiar with navigating the system and adding, updating, and deleting information by using JD Edwards EnterpriseOne menus, forms, or windows. You should also be comfortable using the World Wide Web and the Microsoft Windows or Windows NT graphical user interface.

These books do not review navigation and other basics. They present the information that you need to use the system and implement your JD Edwards EnterpriseOne applications most effectively.

Application Fundamentals

Each application implementation guide provides implementation and processing information for your JD Edwards EnterpriseOne applications.

For some applications, additional, essential information describing the setup and design of your system appears in a companion volume of documentation called the application fundamentals implementation guide. Most product lines have a version of the application fundamentals implementation guide. The preface of each implementation guide identifies the application fundamentals implementation guides that are associated with that implementation guide.

The application fundamentals implementation guide consists of important topics that apply to many or all JD Edwards EnterpriseOne applications. Whether you are implementing a single application, some combination of applications within the product line, or the entire product line, you should be familiar with the contents of the appropriate application fundamentals implementation guides. They provide the starting points for fundamental implementation tasks.

Documentation Updates and Printed Documentation

This section discusses how to:

- Obtain documentation updates.
- Order printed documentation.

Obtaining Documentation Updates

You can find updates and additional documentation for this release, as well as previous releases, on Oracle's PeopleSoft Customer Connection website. Through the Documentation section of Oracle's PeopleSoft Customer Connection, you can download files to add to your Implementation Guides Library. You'll find a variety of useful and timely materials, including updates to the full line of JD Edwards EnterpriseOne documentation that is delivered on your implementation guides CD-ROM.

Important! Before you upgrade, you must check Oracle's PeopleSoft Customer Connection for updates to the upgrade instructions. Oracle continually posts updates as the upgrade process is refined.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Ordering Printed Documentation

You can order printed, bound volumes of the complete line of JD Edwards EnterpriseOne documentation that is delivered on your implementation guide CD-ROM. Oracle makes printed documentation available for each major release of JD Edwards EnterpriseOne shortly after the software is shipped. Customers and partners can order this printed documentation by using any of these methods:

- Web
- Telephone
- Email

Web

From the Documentation section of Oracle's PeopleSoft Customer Connection website, access the PeopleBooks Press website under the Ordering PeopleBooks topic. Use a credit card, money order, cashier's check, or purchase order to place your order.

Telephone

Contact MMA Partners, the book print vendor, at 877 588 2525.

Email

Send email to MMA Partners at peoplebookspress@mmapartner.com.

See Also

Oracle's PeopleSoft Customer Connection, http://www.oracle.com/support/support_peoplesoft.html

Additional Resources

The following resources are located on Oracle's PeopleSoft Customer Connection website:

Resource	Navigation
Application maintenance information	Updates + Fixes
Business process diagrams	Support, Documentation, Business Process Maps
Interactive Services Repository	Support, Documentation, Interactive Services Repository
Hardware and software requirements	Implement, Optimize, and Upgrade; Implementation Guide; Implementation Documentation and Software; Hardware and Software Requirements
Installation guides	Implement, Optimize, and Upgrade; Implementation Guide; Implementation Documentation and Software; Installation Guides and Notes
Integration information	Implement, Optimize, and Upgrade; Implementation Guide; Implementation Documentation and Software; Pre-Built Integrations for PeopleSoft Enterprise and JD Edwards EnterpriseOne Applications
Minimum technical requirements (MTRs) (JD Edwards EnterpriseOne only)	Implement, Optimize, and Upgrade; Implementation Guide; Supported Platforms
Documentation updates	Support, Documentation, Documentation Updates
Implementation guides support policy	Support, Support Policy
Prerelease notes	Support, Documentation, Documentation Updates, Category, Release Notes
Product release roadmap	Support, Roadmaps + Schedules
Release notes	Support, Documentation, Documentation Updates, Category, Release Notes
Release value proposition	Support, Documentation, Documentation Updates, Category, Release Value Proposition
Statement of direction	Support, Documentation, Documentation Updates, Category, Statement of Direction

Resource	Navigation
Troubleshooting information	Support, Troubleshooting
Upgrade documentation	Support, Documentation, Upgrade Documentation and Scripts

Typographical Conventions and Visual Cues

This section discusses:

- Typographical conventions.
- Visual cues.
- Country, region, and industry identifiers.
- Currency codes.

Typographical Conventions

This table contains the typographical conventions that are used in implementation guides:

Typographical Convention or Visual Cue	Description
Bold	Indicates PeopleCode function names, business function names, event names, system function names, method names, language constructs, and PeopleCode reserved words that must be included literally in the function call.
Italics	Indicates field values, emphasis, and JD Edwards EnterpriseOne or other book-length publication titles. In PeopleCode syntax, italic items are placeholders for arguments that your program must supply. We also use italics when we refer to words as words or letters as letters, as in the following: Enter the letter *O*.
KEY+KEY	Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press the W key.
`Monospace font`	Indicates a PeopleCode program or other code example.
“ ” (quotation marks)	Indicate chapter titles in cross-references and words that are used differently from their intended meanings.

Typographical Convention or Visual Cue	Description
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Visual Cues

Implementation guides contain the following visual cues.

Notes

Notes indicate information that you should pay particular attention to as you work with the JD Edwards EnterpriseOne system.

Note. Example of a note.

If the note is preceded by *Important!,* the note is crucial and includes information that concerns what you must do for the system to function properly.

Important! Example of an important note.

Warnings

Warnings indicate crucial configuration considerations. Pay close attention to warning messages.

Warning! Example of a warning.

Cross-References

Implementation guides provide cross-references either under the heading "See Also" or on a separate line preceded by the word *See.* Cross-references lead to other documentation that is pertinent to the immediately preceding documentation.

Country, Region, and Industry Identifiers

Information that applies only to a specific country, region, or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

Example of a country-specific heading: “(FRA) Hiring an Employee”

Example of a region-specific heading: “(Latin America) Setting Up Depreciation”

Country Identifiers

Countries are identified with the International Organization for Standardization (ISO) country code.

Region Identifiers

Regions are identified by the region name. The following region identifiers may appear in implementation guides:

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers may appear in implementation guides:

- USF (U.S. Federal)
- E&G (Education and Government)

Currency Codes

Monetary amounts are identified by the ISO currency code.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like to see changed about implementation guides and other Oracle reference and training materials. Please send your suggestions to Documentation Manager, Oracle Corporation, 7604 Technology Way, Denver, CO, 80237. Or email us at documentation_us@oracle.com.

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions.

Common Fields Used in Implementation Guides

Address Book Number Enter a unique number that identifies the master record for the entity. An address book number can be the identifier for a customer, supplier, company, employee, applicant, participant, tenant, location, and so on. Depending on the application, the field on the form might refer to the address book number as the customer number, supplier number, or company number, employee or applicant ID, participant number, and so on.

As If Currency Code Enter the three-character code to specify the currency that you want to use to view transaction amounts. This code enables you to view the transaction amounts as if they were entered in the specified currency rather than the foreign or domestic currency that was used when the transaction was originally entered.

Batch Number Displays a number that identifies a group of transactions to be processed by the system. On entry forms, you can assign the batch number or the system can assign it through the Next Numbers program (P0002).

Batch Date Enter the date in which a batch is created. If you leave this field blank, the system supplies the system date as the batch date.

Batch Status Displays a code from user-defined code (UDC) table 98/IC that indicates the posting status of a batch. Values are:

Blank: Batch is unposted and pending approval.

A: The batch is approved for posting, has no errors and is in balance, but has not yet been posted.

D: The batch posted successfully.

E: The batch is in error. You must correct the batch before it can post.

P: The system is in the process of posting the batch. The batch is unavailable until the posting process is complete. If errors occur during the post, the batch status changes to *E*.

U: The batch is temporarily unavailable because someone is working with it, or the batch appears to be in use because a power failure occurred while the batch was open.

Branch/Plant Enter a code that identifies a separate entity as a warehouse location, job, project, work center, branch, or plant in which distribution and manufacturing activities occur. In some systems, this is called a business unit.

Business Unit Enter the alphanumeric code that identifies a separate entity within a business for which you want to track costs. In some systems, this is called a branch/plant.

Category Code Enter the code that represents a specific category code. Category codes are user-defined codes that you customize to handle the tracking and reporting requirements of your organization.

Company Enter a code that identifies a specific organization, fund, or other reporting entity. The company code must already exist in the F0010 table and must identify a reporting entity that has a complete balance sheet.

Currency Code Enter the three-character code that represents the currency of the transaction. JD Edwards EnterpriseOne provides currency codes that are recognized by the International Organization for Standardization (ISO). The system stores currency codes in the F0013 table.

Document Company Enter the company number associated with the document. This number, used in conjunction with the document number, document type, and general ledger date, uniquely identifies an original document.

If you assign next numbers by company and fiscal year, the system uses the document company to retrieve the correct next number for that company.

If two or more original documents have the same document number and document type, you can use the document company to display the document that you want.

Document Number — Displays a number that identifies the original document, which can be a voucher, invoice, journal entry, or time sheet, and so on. On entry forms, you can assign the original document number or the system can assign it through the Next Numbers program.

Document Type — Enter the two-character UDC, from UDC table 00/DT, that identifies the origin and purpose of the transaction, such as a voucher, invoice, journal entry, or time sheet. JD Edwards EnterpriseOne reserves these prefixes for the document types indicated:

P: Accounts payable documents.

R: Accounts receivable documents.

T: Time and pay documents.

I: Inventory documents.

O: Purchase order documents.

S: Sales order documents.

Effective Date — Enter the date on which an address, item, transaction, or record becomes active. The meaning of this field differs, depending on the program. For example, the effective date can represent any of these dates:

- The date on which a change of address becomes effective.
- The date on which a lease becomes effective.
- The date on which a price becomes effective.
- The date on which the currency exchange rate becomes effective.
- The date on which a tax rate becomes effective.

Fiscal Period and **Fiscal Year** — Enter a number that identifies the general ledger period and year. For many programs, you can leave these fields blank to use the current fiscal period and year defined in the Company Names & Number program (P0010).

G/L Date (general ledger date) — Enter the date that identifies the financial period to which a transaction will be posted. The system compares the date that you enter on the transaction to the fiscal date pattern assigned to the company to retrieve the appropriate fiscal period number and year, as well as to perform date validations.

JD Edwards EnterpriseOne Tools System Administration Preface

This preface discusses Oracle's JD Edwards EnterpriseOne Tools.

JD Edwards EnterpriseOne Tools

This guide refers to this Oracle product line: JD Edwards EnterpriseOne Tools. In addition to the administrative topics discussed in this guide, essential information describing the setup and design of the system resides in companion documentation. The companion documentation consists of important topics that apply to many or all JD Edwards EnterpriseOne Tools. You should be familiar with the contents of these guides as well. The following companion guides contain information that applies to JD Edwards EnterpriseOne configuration and administration:

- Development Tools: Report Printing Administration Technologies
- Security Administration
- Server and Workstation Administration
- Package Management
- Configurable Network Computing Implementation

See Also

JD Edwards EnterpriseOne Tools 8.96 Development Tools: Report Printing Administration Technologies Guide, "Getting Started with JD Edwards EnterpriseOne Report Printing Administration Technologies"

JD Edwards EnterpriseOne Tools 8.96 Security Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools Security Administration"

JD Edwards EnterpriseOne Tools 8.96 Server and Workstation Administration Guide, "Getting Started with JD Edwards EnterpriseOne Tools Server and Workstation Administration"

JD Edwards EnterpriseOne Tools 8.96 Package Management Guide, "Getting Started with JD Edwards EnterpriseOne Package Management"

JD Edwards EnterpriseOne Tools 8.96 Configurable Network Computing Implementation Guide, "Getting Started with JD Edwards EnterpriseOne Tools Configurable Network Computing Implementation"

CHAPTER 1

Getting Started with JD Edwards EnterpriseOne Tools System Administration

This chapter discusses:

- System Administration overview
- System Administration implementation

System Administration Overview

This guide describes the tools necessary to administer JD Edwards EnterpriseOne and perform system maintenance. This guide also contains instructions on how to use various JD Edwards EnterpriseOne administrative applications to increase the usability of JD Edwards EnterpriseOne software and increase system performance..

System Administration Implementation

In the planning phase of your implementation, take advantage of all Oracle sources of information for JD Edwards EnterpriseOne, including the installation guides and troubleshooting information. A complete list of these resources appears in the preface in *About This Documentation* with information about where to find the most current version of each.

CHAPTER 2

Administering the Data Dictionary

This chapter provides an overview of data dictionary administration and discusses how to update display decimals.

Understanding Data Dictionary Administration

Just as a dictionary contains word definitions, Oracle's JD Edwards EnterpriseOne data dictionary is a central repository that contains data item definitions and attributes. These attributes determine how a data item:

- Appears on reports and forms.
- Validates data entry within an application.
- Assigns column and row descriptions.
- Provides text for field-sensitive help.

The data dictionary is active because changes that you make are automatically reflected in applications without having to recompile the software.

You should assign one or two people to be the data dictionary administrator for each application area in the enterprise. Data dictionary administrators should be experienced with JD Edwards EnterpriseOne and have a comprehensive knowledge of their product area, such as finance or manufacturing. The data dictionary administrator makes all additions, changes, and deletions to data items for the product group. Such changes are reflected in the pristine data dictionary on the enterprise server.

If the setup is similar to the suggested typical customer configuration, then all environments share the same data dictionary. Therefore, the administrator can sign on to any environment to make changes. We recommend that you use the Security Workbench to assign application security on the Data Dictionary application (P92001) to prevent unauthorized users from making data dictionary changes.

Updating Display Decimals in Data Dictionary

Data items that belong to the QTYINV data item class come with the display decimal set at 0 (zero). You can change the display decimal to any number up to 8. For example, if you change the display decimal to 4, instead of seeing 100 you will see 100.0000.

Important! You should change the display decimal value in a CRP environment before any live production data is entered. JD Edwards EnterpriseOne does not have a data conversion feature, so if you change display decimals after users have entered data, the data entered before changing the display decimals will be wrong.

Forms Used to Update Display Decimals

Form Name	FormID	Navigation	Usage
Work With Batch Versions - Available Versions	W98305A	Report Management (GH9111), Batch Versions (P98305)	Work with existing versions of a batch program.
Version Prompting	W98305D	Select the version on the Work With Batch Versions - Available Versions form, and then click Select.	Submit a batch version for processing.

Updating Display Decimals

Enter *GH9111* in the Fast Path. Double-click Batch Versions.

1. On the Work With Batch Versions - Available Versions form, enter *R9200100* in the Batch Application field and then click Find.
2. Select one of these versions and click Select:
   - Update Display Decimals for Class - Proof

     The proof version produces only a report of what the process would do if run in final mode.
   - Update Display Decimals for Class - Final

     The final version actually makes the changes.
3. On the Version Prompting form, click Data Selection and then click Submit.
4. On the Data Selection form, create this statement:

   If BC Data Item Class (F9210) = QTYINV

   Note. QTYINV is the only data item class for which you can modify display decimals. If you modify other data item classes, you must research and test how the modifications affect the software. Also, if you modify which data items reside in the QTYINV data item class, you must research and test how the modifications affect the software.
5. If you changed the Data Selection statement, click OK.
6. On the Processing Options form, complete these fields and then click OK:
   - Enter *1* to run in Update Mode or *0* to run in Proof Mode. The default is 0.
   - Data Item Class
   - New Data Display Decimals
7. On the Printer Selection form, click OK to run the batch application.
8. On each workstation, delete these spec files:

   ```
   qlbltbl.ddb, qlbltbl.xdb
   dddict.ddb, dddict.xdb
   ddtext.ddb, ddtext.xdb
   ```
9. To push the display decimal changes out to users, run R92TAM on the server on which the changes were made.

CHAPTER 3

Understanding JD Edwards EnterpriseOne OMW Administration

This chapter provides an overview of JD Edwards EnterpriseOne Object Management Workbench (OMW) administration.

JD Edwards EnterpriseOne OMW Administration Overview

JD Edwards EnterpriseOne OMW automates many of the object management tasks that users perform in the software. Much of this automation requires careful configuration by the system administrator through the Object Management Workbench Configuration program (P98220).

Use P98220 to configure these features:

Option	Description
Constants	Enables you to set general constants pertaining to JD Edwards EnterpriseOne OMW projects.
SAR System Integration	Enables you to disable SAR system integration with JD Edwards EnterpriseOne OMW and, thus, JD Edwards EnterpriseOne development tools.
Logging System	Enables you to specify the project and object events to be logged. In the event that logging fails, you can also disable development or allow development but disable transfers.
Object Action Notification	Enables you to enable and disable Object Action Notification, which sends a notification message when an action such as checkin or checkout is performed on an object.
Notification Setup	Enables developers to be notified, using subscription, when actions are performed on an object.
Activity Rules	Enables you to add and modify project statuses and object transfer activity rules.
User Roles	Enables you to maintain user roles.
Allowed Actions	Enables you to assign to a user role the actions allowed for each object type during a specific project status.
Save Locations	Enables you to add, modify, and delete the locations where you save objects.

You can find the instructions for performing these JD Edwards EnterpriseOne OMW administrative tasks in the *JD Edwards EnterpriseOne Object Management Workbench Guide.*

See *JD Edwards EnterpriseOne Tools 8.96 Object Management Workbench Guide*, “Configuring JD Edwards EnterpriseOne OMW”.

CHAPTER 4

Working with Servers

This chapter provides an overview of the Work With Servers program (P986116) in Oracle's JD Edwards EnterpriseOne and discusses how to:

- Manage server jobs
- Manage job queues
- Manage JD Edwards EnterpriseOne subsystems

Understanding the Work With Servers Program (P986116)

The Work With Servers program (P986116) in Oracle's JD Edwards EnterpriseOne provides a central location from which system administrators can monitor and control:

- Server jobs
- JD Edwards EnterpriseOne subsystems

As a system administrator, you can use the Work With Servers program to print, view, remove, terminate, release, or hold any jobs that currently reside in a queue on any JD Edwards EnterpriseOne server. Similarly, workstation users can control only those jobs submitted by them. This option is generally restricted to only those jobs associated with a specific user ID.

Also, you can use the Work With Servers program to end and to stop JD Edwards EnterpriseOne subsystems, and to view the status of JD Edwards EnterpriseOne subsystems that are running or are waiting to process jobs.

Managing Server Jobs

This section provides an overview of server jobs and discusses how to:

- Set processing option for Work With Servers.
- Check the status of reports.
- Change the priority and the printer for jobs.
- Print jobs.
- View reports online.
- View the logs for a job.
- Terminate jobs.

- Hold and release jobs.

Understanding Server Jobs

By using the Work With Servers application, system administrators can print, view, and delete job records from the outqueue. They can also terminate, release, or hold any jobs that currently reside in a queue on any JD Edwards EnterpriseOne server. Similarly, using the Submitted Job Search form, users can, in general, control only those jobs submitted by them.

You should use JD Edwards EnterpriseOne security to restrict access to the Work With Servers application. In general, access to this program should be granted only to administrator-level users because the ZJDE0001 version of the Work With Servers program enables users to view and control server jobs for all users. End users should be restricted to the ZJDE0002 version, which is known as the Submitted Job Search form. This version of the application restricts users to viewing and modifying only those jobs that were submitted under their user ID initially. Both programs are located on the System Administration Tools menu (GH9011).

Job Status and Priority

After you submit the report, you can check the status of the job in the queue. Depending on the status of the job, you can perform tasks such as printing or deleting the report, viewing the report output online, and holding the report in the queue.

You can also move the priority of the job to a lower or higher status while the job is at the status of W (Waiting).

Overriding Printer Location for Jobs

You can override the location where the job prints. For jobs with a status of D (Done) and E (Error), you can send the job directly to the default printer without viewing the PDF file online. A status of D means that the processing for the job completed successfully. A status of E means that an error occurred during processing. If you print a job with a status of E, you print an error log to aid you when you troubleshoot the report.

Viewing Reports Online

After the job finishes processing on the server, you can view the report output online. For most jobs, the output is in Portable Document Format (PDF), which can be viewed with Adobe Acrobat Reader. When you view the report output online, the system also creates a PDF file for the report in the following directory on the workstation:

```
\811\PrintQueue
```

You can attach PDF files to email messages; move or copy the files; and, because most current web browsers can read PDF files, post the reports to a web site. Also, you can copy text from Acrobat Reader to the clipboard and paste that text into other applications.

Job Logs

You can view logs that detail the steps taken while the job processed. From the Submitted Job Search form, you can access the `jde.log` and the `jdedebug.log` for the report. These logs are helpful if you need to troubleshoot a report that resulted in error. These logs exist on the machine where the job ran.

The `jde.log` is a general-purpose log used to track error messages generated by JD Edwards EnterpriseOne processing. The `jde.log` tracks any fault that might occur within the software, including whether the sign on is successful. When you are looking for startup errors, you should read the `jde.log` from the top down. For other errors, you should read from the bottom up.

The `jdedebug.log` contains API calls, BSFN logs, and SQL statements, as well as other messages. You can use this log to determine at what time normal execution stopped. The system does not use the`jdedebug.log` to track errors; instead, it uses this log to track the timing of processes.

Terminating Jobs

You can manually terminate a job that is processing. When you terminate a job, you do not delete it; rather, you move the job to the status of E (Error). With the job at the status of E, you can print an error log or delete the job.

Holding and Releasing Jobs

If a job is at the status of W (waiting), you can hold the job. You might choose to hold a job if the job is large enough to affect the performance of the server on which it processes. You can release a job when server performance is not an issue, such as after regular business hours.

Note. If you want to stop a job that is at a status of P (Processing), you must terminate the job. You cannot restart a job after you terminate it; you must resubmit the job to the server.

See Also

JD Edwards EnterpriseOne Tools 8.96 Server and Workstation Administration Guide

Forms Used to Manage Server Jobs

Form Name	FormID	Navigation	Usage
Work With Servers,	W986116A	System Administration Tools (GH9011), Data Source Management, Work With Servers (P986116).	Select a server in which you want to locate a job.
Submitted Job Search	W986110BA	On the Work With Servers form, from the Row menu, select Server Jobs.	Print, terminate, hold, release, or view a job. Users can manage jobs submitted by their user ID initially. Depending on the security level, you can change the User ID field and the Job Queue field to search for other jobs.
Job Maintenance	W986110BC	On the Submitted Job Search form, select a job with which to work and click Select.	Review information about the batch job, modify the priority of the job, or change the printer on which the job will print.
Printer Selection	W986162B	On the Submitted Job Search form, select a job and then select Print from the Row menu.	Override printer-specific information.
View Logs	W986110BD	On the Submitted Job Search form, select the job for which you want to view a log, and then select View Logs from the Row menu.	View the `jde.log` and the `jdedebug.log`.

Setting Processing Option for Work with Servers (P986116)

Although processing options are set up during the JD Edwards EnterpriseOne implementation, you can change processing options each time you run a batch application.

1. Security Flag

Use this processing option to specify how submitted jobs can be viewed. Values are:

Blank

No Security

1

Allow users to view jobs by group.

2

Allow users to view only their own jobs.

Checking the Status of Reports

Access the Submitted Job Search form.

User ID	Change the user ID if you want to work with a report submitted by a different user. You can use a wildcard (*) to find a specific user. The default user ID is the user logged on to the current session.
Job Queue	Enter the name of the logical queue on the server for which you want to view jobs.
Status	Click the search button in the Status field to read the UDCs for status codes in the installation

Changing the Priority and the Printer for Jobs

Access the Work With Servers form.

1. Select a server with which to work and, from the Row menu, select Server Jobs.

   By default, the Submitted Job Search form lists jobs for the User ID for the requesting workstation. Depending on the application security level, you can change the User ID field and the Job Queue field to search for other jobs.

   Note. A job must be at a status of W (Waiting) to change the priority.

2. Select a job with which to work and click Select.
3. On the Job Maintenance form, modify the information in the Job Priority field and click OK. The value that you enter in this field determines how the job will execute based on this priority. Values 0-9 are valid, where 0 is the highest priority.

Printing Jobs

Access the Work With Servers form.

1. From the Row menu, select Server Jobs.
2. On the Submitted Job Search form, select the job that you want to print, and then choose Print from the Row menu.

   The Printer Selection form appears. This form provides printer-specific information as well as information about the format of the report.
3. To print the job, click OK.

Viewing Reports Online

Access the Work With Servers form.

Note. Before you view the report online, verify that you have Adobe Acrobat Reader installed on the workstation.

1. Select a server from the list and then click Select or select Server Jobs from the Row menu.
2. On the Submitted Job Search form, select the job that you want to view and then select View Job from the Row menu.

   Adobe Acrobat Reader displays an online version of the report output.

Viewing the Logs for a Job

Access the Work With Servers form.

1. Select the server that processed the job that you want to view, and click Select, or select Server Jobs from the Row menu.
2. On the Submitted Job Search form, select the job for which you want to view a log, and then select View Logs from the Row menu.

   The View Logs form appears. On this form, you can view the `jde.log` and the `jdedebug.log`.
3. Click OK to view the logs.

Note. If you choose both the`jde.log` and the`jdedebug.log`, the logs open in the same window. To view the logs separately, you must select the logs separately.

Terminating Jobs

Access the Work With Server form.

1. Select a server from the list or use the query by example row to select a specific server.
2. Click Select or select Server Jobs from the Row menu.
3. On the Submitted Job Search form, select the job to terminate, and then select Terminate from the Row menu.

Note. A job must be at a status of P (processing) to terminate the job.

---

4. Click Find to update the detail area.

   The status of the job changes to E (error).

Holding and Releasing Jobs

Access the Work With Servers form.

1. Select a server from the list or use the query by example row to select a specific server.
2. Click Select or select Server Jobs from the Row menu.

   The Submitted Job Search form appears.
3. To hold a job, select the job and then select Hold from the Row menu.
4. Click Find to update the detail area.

   The status of the job changes to H (Hold).
5. To release a job, select the job and then select Release from the Row menu.

   The job must be at the status of H (Hold).
6. Click Find to update the detail area.

   The status of the job changes to reflect the position of the job in the queue, for example, W (Waiting), S (In Queue), or P (Processing).

Managing Job Queues

This section provides an overview of job queues and discusses how to:

- Add a job queue.
- Copy a job queue.
- Change the status of a job queue.
- Override a job queue.

Understanding Job Queues

Each JD Edwards EnterpriseOne server instance starts a queue kernel process that manages batch processes across operating system platforms. The process keeps track of all jobs that are submitted and controls the order in which the jobs run.

JD Edwards EnterpriseOne uses two tables to maintain queue records:

- Job Control Status Master table (F986110), which maintains records on the status of each job submitted to a queue.
- Queue Control Status Master table (F986130), which stores the names of each queue, such as QBATCH, the name of the server on which the queue runs, the port number for the server instance, the queue status and type, and the maximum number of active jobs allowed.

Note. Since F986130 is a system table, be sure to account for it when you map objects using Object Configuration Manager (OCM).

The following list summarizes how the software, using the queue kernel, manages a UBE that you launch:

- Starts queue kernel when the server instance starts.
- Verifies that a record exists in the F986130 table for the queue to which the job is submitted. If the job is intended for a non-EnterpriseOne queue, verifies that the native queue (for example, iSeries) exists.
- Inserts job record into the F986110 table.
- Sends a message to the queue kernel that the new job exists.
- Adds the job to a wait list.
- Schedules the job or submits it to the native queue.
- Starts the job.
- Runs the job.
- Updates the job record in the F986110 table upon receiving a message from the UBE process that the job is complete.
- Removes the job from the list of active jobs.
- Schedules another job.

The queue kernel also follows a subroutine in scheduling jobs. The following list summarizes the subroutine that the queue kernel follows:

- Verifies that jobs in the queue are waiting to be run.

- Verifies that the number of jobs waiting to be run is less than the maximum number of jobs allowed for the queue.
- Takes the highest priority job from the wait list and updates its status to S (Submitted).
- Removes the job from the wait list and adds it to the active list.

Administering Job Queues

Use the Job Queue Maintenance application (P986130) to define and manage job queues. This application enables you to dynamically administer job queues. You can use this application to create, modify, copy, delete, or change the status of job queues, regardless of platform. For example, you can use this application to add a queue record to the Queue Control Status Master table (F986130). You can also revise an existing queue record. For example, you might want to change the maximum number of jobs that can run in a queue.

In addition, when you set up job queues, you can define a default queue in which to submit jobs.

Overriding a Job Queue

When you prepare to submit a batch application, you can change the values of the parameters that define the submission by overriding the job queue. Overriding the job queue means that you change the job queue to which the job is submitted on the server.

To override the job queue for a batch version, you launch the Batch Versions application (P98305), select a batch version, and access the Advanced Version Prompting form (W98305I). The override queue must be one that is available for the server and port.

In working with the Advanced Version Prompting form, you can override the job queue only if the queue kernel is active and if the batch version is mapped to run on the server. If the batch version is mapped to run locally, you cannot override the job queue, even if the queue kernel is active, unless you select the Override Location option.

Note. Overriding the job location means that you change the machine that will run the batch application. For example, a batch application might run locally by default. You can override the processing location to a server, and the batch application will run on the server. Conversely, you can change the processing location from a server to a workstation.

JD Edwards EnterpriseOne displays a Verify Overriding the Job Queue form if the job runs locally and you do not override the processing location.

The status of the queue kernel and the default processing location for the batch application determine the way the Override Job Queue option appears in the Advanced Version Prompting form. The following table summarizes the queue kernel status and processing location combinations that can occur, and the effect each combination has on the Override Job Queue option:

Queue Kernel Status	UBE Processing Location	Status of Job Queue Override Option
Inactive	Local or server	Not visible
Active	Local	Visible but disabled
Active	Local, but Override Location option chosen	Enabled
Active	Server	Enabled

Prerequisites

Before you work with job queues, you must activate the queue kernel. To do so, perform these tasks:

- Make sure that the server's jde.ini file contains these settings:

```
[JDENET_KERNEL_DEF14]
krnlName=QUEUE KERNEL
dispatchDLLName=jdekrnl.dlldispatchDLLFunction=_DispatchQueueMessage@28max
NumberofProcesses=1numberofAutoSartProcesses=0
[DEBUG]
QKLog=0
```

  Where a value of 0 means that only an error log is generated. You can change the setting to 1 if you need to generate debug logs for troubleshooting purposes.

```
[NETWORK QUEUE SETTINGS]
QKActive=1QKOnIdle=300
```

  Where a value of 1 means that the queue kernel is active and a value of 300 sets the queue kernel on idle time to 300 seconds.

- Add the following setting to the client jde.ini file:

```
[NETWORK QUEUE SETTINGS]
QKActive=1
```

Forms Used to Manage Job Queues

Form Name	FormID	Navigation	Usage
Work With Job Queues	W986130A	Batch Processing Setup menu (GH9013), Job Queues (P986130).	Add a job queue and change the status of a queue.
Job Queue Revisions	W986130B	On the Work With Job Queues form, click Add.	Add information for a new job queue. Revise or copy a job queue.
Work With Batch Versions - Available Versions	W98305A	Report Management (GH9111), Batch Versions (P98305)	Override a job queue.
Version Prompting	W98305D	On the Work With Batch Versions - Available Versions form, select a version, and then click Select.	Access the Advanced Version Prompting form. Enter data selection or data sequencing.
Advanced Version Prompting	W983051	On Version Prompting, select Advanced from the Form menu.	Select the Override Job Queue option.
Job Queue Search	W986130C	On Version Prompting, click Submit.	Find and select the job queue that you want to override.

Adding a Job Queue

Access the Work With Job Queues form.

1. Click Add.
2. On the Job Queue Revisions form, in the Host field, enter the name of the server on which the queue will run.
3. Enter the name of the queue in the Job Queue field.
4. In the Job Queue Status field, enter *01* if you want the queue to be active, or *02* if you want the queue to be inactive.
5. In the Queue Type field, define whether the queue is a JD Edwards EnterpriseOne queue or a non-EnterpriseOne queue.

   A non-EnterpriseOne queue works only on the iSeries server.
6. In the Maximum Batch Jobs field, define the maximum number of jobs that can run in the queue.
7. In the Port Number field, specify the port number for the server instance on which the queue will run.
8. In the Default Queue field, check the box for the default queue, or leave it blank for a non-default queue.

Copying a Job Queue

Access the Work With Job Queues form.

1. On the Work With Job Queues form, find the queue that you want to copy and click Copy.
2. On Job Queue Revisions, you can modify any of these fields:
   - Host
   - Job Queue
   - Job Queue Status
   - Queue Type
   - Maximum Batch Jobs
   - Port Number
   - Default Queue
3. Click OK to complete the copy.

Changing the Status of a Job Queue

Access the Work With Job Queues form.

1. On the Work With Job Queues form, find the queue whose status you want to change.
2. From the Row menu, select Change Status.

   JD Edwards EnterpriseOne changes the status of the queue from Active to Inactive or from Inactive to Active, depending on its previous status.

Overriding a Job Queue

Access the Work With Batch Versions - Available Versions form.

1. On the Work With Batch Versions - Available Versions form, find a version of a job that you want to submit and click Select.
2. On the Version Prompting form, select Advanced from the Form menu.

3. On the Advanced Version Prompting form, select the Override Job Queue option and click OK.

   Note. If the queue kernel is not active, this option is not visible.

4. In the Version Prompting form, select either, both, or neither of these options and click Submit:
   - Data Selection
   - Data Sequencing
5. On the Job Queue Search form, find the name of an available queue for the host and port name.
6. Select the queue that you want to override to and click Select.
7. Complete the data selection and sequencing and the processing options required to submit the job and select a printer, if necessary.

Managing JD Edwards EnterpriseOne Subsystems

This section provides an overview of JD Edwards EnterpriseOne subsystems and discusses how to:

- Locate subsystems running on a server.
- Review job records for subsystems.
- Terminate subsystems.

Understanding JD Edwards EnterpriseOne Subsystems

Within JD Edwards EnterpriseOne, subsystems are defined as continuously running batch jobs that run independently of, and asynchronously with, JD Edwards EnterpriseOne applications. Subsystem jobs function within the logical process of the operating system or the queue defined for the server platform. You can configure JD Edwards EnterpriseOne to use one or more subsystems.

The term *subsystem* is an industry-wide generic term that usually indicates a system that is a subprocess to an operating system. On iSeries server platforms, a subsystem is a logical process that is used to run system jobs, whether they are JD Edwards EnterpriseOne or other application jobs. For UNIX, JD Edwards EnterpriseOne subsystem is functionally equivalent to a daemon. On UNIX and Windows server platforms, system jobs are processed in queues; these queues are functionally equivalent to subsystems on the iSeries platform.

How JD Edwards EnterpriseOne Uses Subsystems

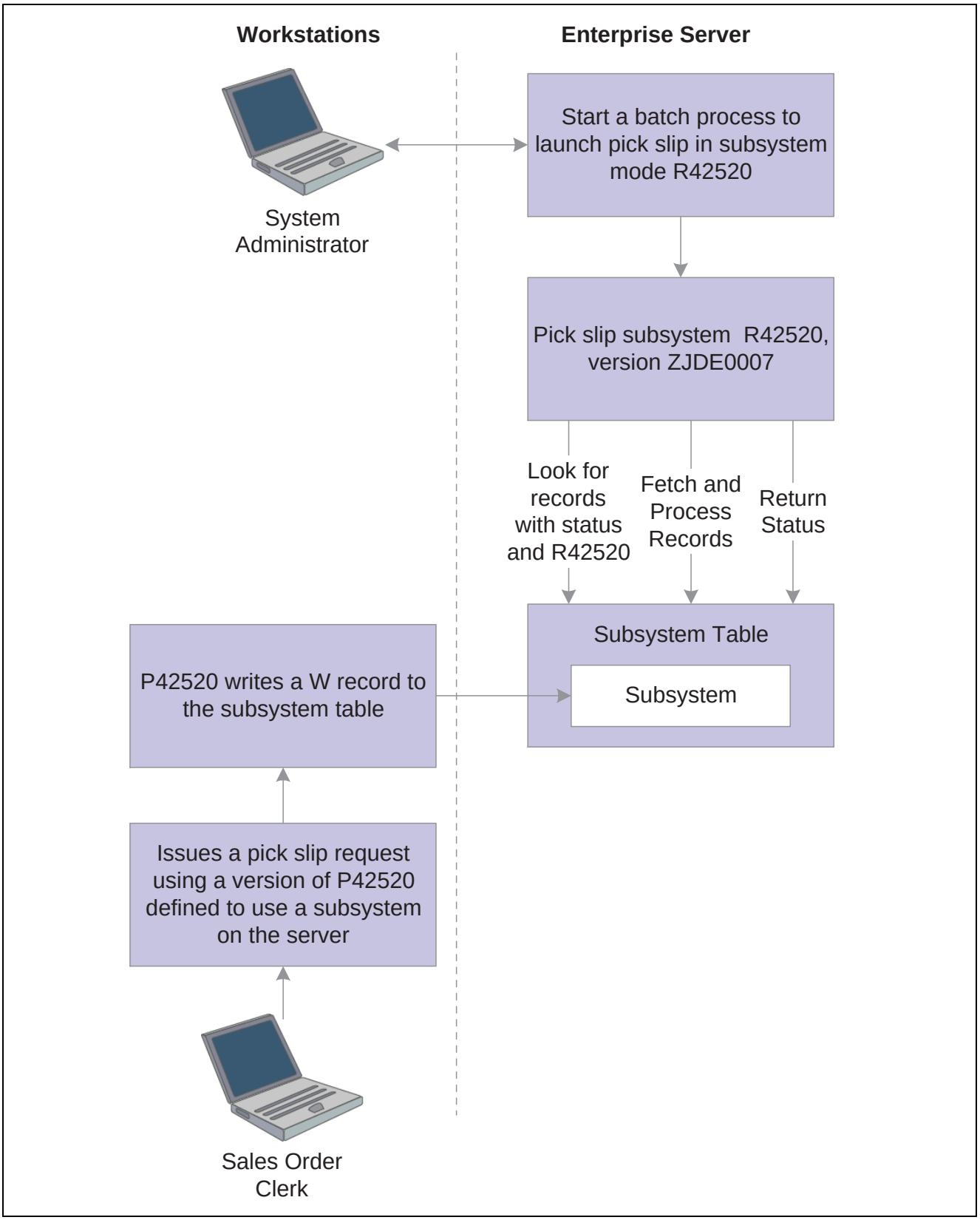
Some JD Edwards EnterpriseOne applications are designed to use subsystems to complete needed work. For example, you can instruct Sales Order Processing to print pick slips through a JD Edwards EnterpriseOne subsystem. You activate a subsystem through the processing options of a batch application. Then you create a specific version of the batch application, using that processing option to run the application in subsystem mode.

You must manually start subsystems to minimize the consumption of system resources. When started, JD Edwards EnterpriseOne subsystems run continuously, looking for and processing requests from JD Edwards EnterpriseOne applications. Subsystems run until you terminate them.

Typically, you use subsystem jobs running on the enterprise server to off load processor resources from the workstation. Instead of queuing requests and running them in batches at specified times of the day, you can direct the requests to a subsystem, where they are processed in realtime. For example, you might be running the Sales Order Entry application on a workstation and want to print pick slips. If you are using a version of pick slips that has the Subsystem Job function enabled, the request is executed by a subsystem job. The pick slip request is routed to and processed by the subsystem job on the defined enterprise server. As a result, no additional processing resources are required from the workstation machine to actually print the pick slip.

When an application issues a request for a job to run in a subsystem, it places a record in the Subsystem Job Master table (F986113). These records are identified by subsystem job name and contain status and operational indicators. Embedded in the record is key information that allows the subsystem to process the record without additional interaction with the requesting application. The continuously running subsystem monitors the records in this table. If the subsystem finds a record with its process ID and appropriate status indicators, it processes the record and updates the status accordingly.

This illustration displays the logical sequence of events associated with subsystems:



Logical sequence of events associated with subsystems

Enabling Subsystems

To prevent excessive processing overhead during server startup and to prevent unnecessary uses of processor resources for subsystem jobs that might be in use, you must manually start subsystems. Generally, the system administrator or manager-level user is responsible for this task. To manually start subsystems, a version of a JD Edwards EnterpriseOne batch process with a processing option set to enable the use of subsystems is run.

As described, the way that you initially control the creation and start-up of these subsystems and queues depends on the server platform.

Platform (Subsystem or Queue)	Description
iSeries (JDENET)	One iSeries subsystem is used for JD Edwards EnterpriseOne. This subsystem is started automatically when you issue the JD Edwards EnterpriseOne startup command STRNET. The subsystem name is version-specific. For example, for release 8.11, the subsystem name is psft811. To process requests that are destined for JD Edwards EnterpriseOne subsystems, you must define a specific job queue running under the JDENET subsystem. For example, a job queue might be named QBATCH. User requests for JD Edwards EnterpriseOne subsystem-defined batch jobs are executed by the job queue that is based on definition in the iSeries user profile.
UNIX (jdequeue)	One or more queues can exist for JD Edwards EnterpriseOne. These queues can be named the same or differently. You define queues by parameters in the startup shell scrip `RunOneWorld.sh`. To process requests that are destined for subsystems, you must define one or more queues. For example, a jdequeue might be named QBATCH. User requests for subsystem-defined batch jobs are executed by the job queue, based on the process ID.
NT (`jde.ini` settings)	One or more queues can exist for JD Edwards EnterpriseOne. These queues must have the same name. You define queues using settings in the `jde.ini` file. To process requests that are destined for subsystems, you must define the name and number of queues in the [NETWORK QUEUE SETTINGS] section of the `jde.ini` file. For example, a jdequeue might be named QBATCH. User requests for subsystem-defined batch jobs are executed by the job queue, based on the process ID.

System administrators can display all of the subsystems that are running on a server by using the Subsystem Jobs application (P986113). Use this application to:

- Locate a list of subsystems that are running on a server.
- Locate a list of subsystem records that are unprocessed (not available for iSeries servers).
- Locate the current record that a subsystem is processing (not available for iSeries servers).
- Stop or delete any subsystem.

Subsystem Job Records

Multiple JD Edwards EnterpriseOne processes write records to the Subsystem Job Master table (F986113). Each record has a status code that identifies subsystem request types and operational status. You can use Work With Server Jobs to view the records in this table.

Terminating Subsystems

You can use Work With Server Jobs to terminate subsystems. The following two methods of termination are available:

- Stopping a subsystem job causes it to terminate after it completes processing the current record. Additional unprocessed records in the Subsystem Job Master table (F986113) will not be processed, and no new records can be written. Essentially, the unprocessed records will be lost; that is, the process that initiated the record is not notified that the record was not processed.
- Ending a subsystem job causes it to terminate after processing all of the existing subsystem records. No new records can be written to the Subsystem Job Master table (F986113).

Forms Used to Manage JD Edwards EnterpriseOne Subsystems

Form Name	FormID	Navigation	Usage
Work With Servers	W986116A	System Administration Tools (GH9011), Data Source Management, Work With Servers (P986116).	Select a server in which you want to locate a subsystem.
Work With Subsystems	W986113A	On the Work With Servers form, from the Row menu, select Subsystem jobs.	Review the status and type of the subsystem. Stop or end a subsystem.
View Jobs	NA	On the Work With Subsystems form, from the Row menu, select View Jobs.	Review server jobs and job types.

Locating Subsystems Running on a Server

Access the Work With Servers form.

1. On the Work With Servers form, select a server from the list or use the query by example row to select a specific server.
2. From the Row menu, select Subsystem Jobs.
3. On the Work With Subsystems form, select one of these options:
   - &Processes

     A process is a subsystem that is waiting for work. It is identified by an S (subsystem job) value in the Job Type field.
   - &Waiting Jobs

     Waiting jobs are report jobs that are queued for a subsystem. They are identified by an R (subsystem record) value in the Job Type field.

     All currently running subsystems are displayed. Report number and version identify the running subsystems.

4. Review these fields in the detail area to note the type and status of the subsystem:

Field	Description
Job Type	Indicates the subsystem type. Values are: • R Subsystem record. • S Subsystem job.
Job Status	Indicates the status of the subsystem job or record. Values are: • W Subsystem record waiting. • P Subsystem record processing. • E Subsystem record to end the job. • R Subsystem job running.

Reviewing Job Records for Subsystems

Access the Work With Subsystems form.

1. After locating a subsystem job, on the Work With Subsystems form, click Find.
2. Select a record in the detail area, and then select View Jobs from the Row menu.
3. On the View Jobs form, click Find.

   A list is displayed for all server jobs in the Subsystem Job Master (F986113) with an R (subsystem job running) job type.

Terminating Subsystems

Access the Work With Subsystems form.

1. Select the running subsystem that you want to stop.
2. To stop a subsystem, from the Row menu, select Stop Subsystem.

   Note. If you are viewing Waiting Jobs from Work With Server Jobs or if you are viewing subsystem jobs by selecting the View Jobs from Work With Server Jobs, the Stop Subsystem selection is disabled from the Row menu selection.

3. To end a subsystem, from the Row menu, select End Subsystem Job.

   Note. If you are viewing Waiting Jobs from Work With Subsystems, the End Subsystem selection is disabled from the Row menu selection.

4. On End Subsystem Job, click OK.

CHAPTER 5

Setting Up PIM Synchronization for JD Edwards EnterpriseOne CRM Users

This chapter provides an overview of Personal Information Management (PIM) synchronization and discusses how to:

- Set the processing option for the PIM Server.
- Set up users for PIM synchronization.
- Export PIM user records.
- Import JD Edwards EnterpriseOne PIM synchronization records into Intellisync Server.

Understanding PIM Synchronization

Oracle's JD Edwards EnterpriseOne provides the ability to synchronize user data between third-party PIM systems and JD Edwards EnterpriseOne CRM users. With this synchronization, CRM users can share and manage personal information such as contacts, appointments, and tasks stored in IBM Domino or Microsoft Exchange PIM systems. For example, as appointments are created in the CRM system, corresponding objects are created in the PIM system database. If users update or delete items in either the CRM or PIM system, the changes are synchronized in both locations.

JD Edwards EnterpriseOne relies on a third-party software product, Intellisync Server, to enable data synchronization between JD Edwards EnterpriseOne CRM and PIM systems. Intellisync Server uses these items to synchronize PIM data:

- CSV file.

  The CSV file contains the PIM server records for each JD Edwards EnterpriseOne user that you set up for PIM synchronization. You generate this file from JD Edwards EnterpriseOne after you set up users for PIM synchronization.

- Synchronizing template.

  This template determines the types of information that are synchronized between the JD Edwards EnterpriseOne and PIM systems.

The process for setting up PIM synchronization is:

1. Enter each JD Edwards EnterpriseOne user's PIM system information into JD Edwards EnterpriseOne.
2. Export this information from JD Edwards EnterpriseOne into a CSV file.
3. Using Intellisync Server Bulk User Import, import the CSV file, along with a synchronizing template, into the Intellisync Server.

Note. JD Edwards EnterpriseOne provides a standard synchronizing template for PIM synchronization.

User Defined Codes for PIM

JD Edwards EnterpriseOne uses these user defined codes (UDCs) to categorize the data shared with PIM systems:

UDC Type	UDC Codes and Definitions
01|PH (Phone Type)	HOM: PIM-Home <blank>: PIM-Business FAX: PIM-Fax CAR: PIM-Car
01|RT (Relation Type)	A: PIM-Assistant S: PIM-Spouse C: PIM-Child
01|AT (Address Type)	H: PIM-Home W: PIM-Work
01|ET (Electronic Address Type)	E: PIM-Email Address I: PIM-Web Address
01|SG (PIM Sync Category)	PIMSG: PeopleSoft This UDC is used with MS Exchange only. ACTTY: APPOINTMT TASK: TODOTASK CALTY: PERSONAL **Note.** The ACTTY, TASK, and CALTY codes apply only to JD Edwards EnterpriseOne 8.12 and later releases.

Note. You should not change the Description Line 2 values of PIM UDCs.

Understanding the 01|SG UDCs

The 01|SG UDCs were created solely for PIM synchronization and are hard-coded, which means that you can only change the Description Line 1 values of these UDCs. The PIM server uses the default value in the Description Line 1 of the 01|SG UDCs to categorize events, tasks, and contacts in JD Edwards EnterpriseOne.

This table describes how the PIM server uses the UDCs in 01|SG to categorize PIM data:

01|SG UDC	Description
PIMSG	The PIM server uses the Description Line 1 value of PIMSG to categorize the data that is synchronized between JD Edwards EnterpriseOne and Microsoft Exchange or Domino. The default value of Description Line 1 is "PeopleSoft." You can modify this value. In Microsoft Outlook, appointments, tasks, and contacts have a Categories field. When PIM users create appointments, tasks, or contacts in Microsoft Outlook, they must enter a value in the Categories field that is the same as the value in Description Line 1 of PIMSG. The PIM server uses this value to categorize PIM data in JD Edwards EnterpriseOne. In JD Edwards EnterpriseOne, there is no equivalent Categories field for appointments, tasks, and contacts. However, when PIM users create these items in JD Edwards EnterpriseOne, the PIM server associates the value in Description Line 1 of PIMSG with these items. When users open these appointments, tasks, and contacts in Microsoft Outlook, the Categories field is populated with the value from the Description Line 1.
CALTY	The Description Line 1 value in the CALTY UDC specifies the JD Edwards EnterpriseOne calendar type that is being used by PIM users. JD Edwards EnterpriseOne contains different types of calendars, such as PERSONAL and CRM. When importing information from the PIM server, JD Edwards EnterpriseOne uses the value that you enter in the Description Line 1 of the CALTY UDC to determine which calendar the information belongs. The default value for Description Line 1 is PERSONAL.
ACTTY	JD Edwards EnterpriseOne contains several event types, such as MEETING, CALL, DOWNTIME, and APPOITMNT. The 01|AC UDC type contains a code for each event type. The PIM server uses the value in the Line Description 1 of ACTTY to categorize events sent from the PIM system to JD Edwards EnterpriseOne. The default value is APPOITMNT.
TASK	JD Edwards EnterpriseOne contains several task types, such as TODOTASK, RESEARCH, and FOLLOWUP. UDCs for these tasks are defined in the 01|AC UDC type along with the event types. The PIM server uses the value in the Line Description 1 of the TASK UDC to categorize tasks sent from the PIM system to JD Edwards EnterpriseOne. The default value for Description Line 1 is TODOTASK.

Prerequisite

Install the PIM Sync Servlet.

See *EnterpriseOne Tools 8.96 PIM Sync Installation and Configuration* guide for your platform and application server.

Setting the Processing Option for the PIM Server

If all users use the same type of PIM system (Microsoft Exchange or IBM Domino), you can set a processing option to specify the type of PIM server to set up synchronization with.

Note. Do not set this processing option if both PIM servers are used.

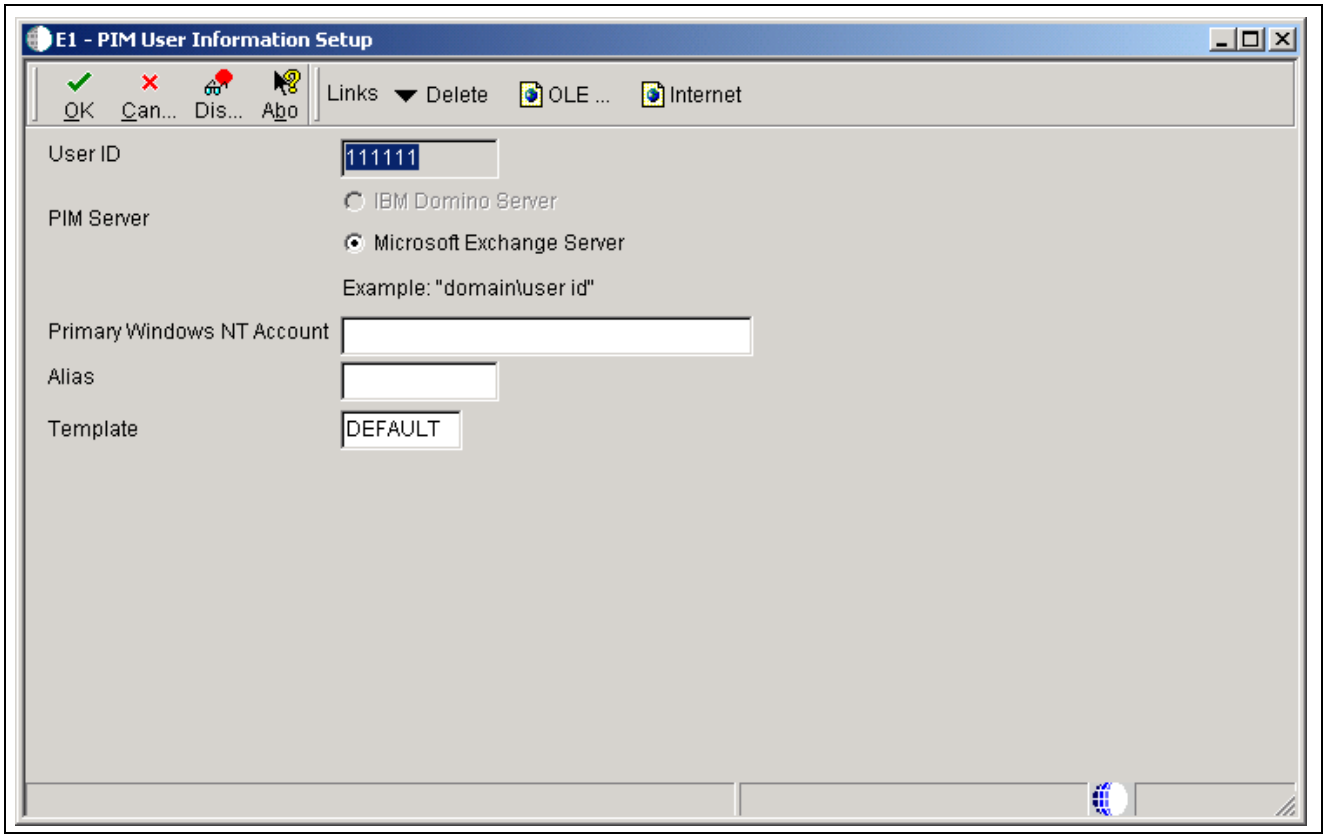
From the System Administration Tools menu (GH9011), select User Management. Right-click User Profiles, select Prompt for, and then select Values.

On the Processing Options form, select the PIM Server Setup tab and then enter the appropriate value for the PIM server:

- Enter *D* for IBM Domino Server.
- Enter *X* for Microsoft Exchange Server.

Setting Up Users for PIM Synchronization

Access the E1 - PIM User Information Setup form.From the System Administration Tools menu (GH9011), select User Management, User Profiles. On the Work with Users/Roles form, select a user record and then select PIM User Setup from the Row menu.



E1 - PIM User Information Setup form

1. Select the user's PIM server type:
   - IBM Domino Server
   - Microsoft Exchange Server

   Note. If you specified the PIM server type in the PIM Server Setup processing option, you cannot select this option here.

2. For IBM Domino Server, complete these fields:

Field	Description
Mail File Location	Enter the user's Lotus Notes mail database file. Specify the full name relative to the Domino Data Directory, without the .nsf extension, for example `mail\UserMailDatabase`.
Address Book Name	Enter the user's Lotus Notes address book database name. Specify the full name relative to the Domino Data Directory, without the .nsf extension, for example `contacts\UserAddressBook`.
Address Book Server	Enter the user's Lotus Notes address book database server.

3. In the Template field, click the search button to specify the Intellisync Server synchronizing template:
   a. On E1 – PIM Template Revisions, complete these fields:

      Template

      Enter a name for the template. You can name the template based on the role or function of the JD Edwards EnterpriseOne users that you associate this template to, for example you can name the template CRMHR template for human resources users or DEFAULT as the name of the default template for all users.

      Template File

      Enter a fully qualified pathname to the synchronizing template file. This must be a pathname that Intellisync can recognize.

   b. After you create the template, double-click the template to select it and then click OK.
4. For Microsoft Exchange Server, complete these fields:
   - Primary Windows NT Account

     Enter the user's Windows NT account user name and domain name. The Intellisync server user name must be unique, for example Domain\SampleUser1.
   - Alias

     Because an Intellisync account can only reference one mailbox, you must enter an alias to specify which mailbox to use.
   - Template

     See step 3 for instructions on how to specify the template.

Exporting PIM User Records

After you set up users for PIM synchronization, you must export each user record to a CSV file. The CSV file contains each user's PIM system information. Subsequently, you must import this file into the Intellisync Server to complete the synchronization setup.

This section discusses how to:

- Export all user records.
- Export a selected list of PIM user records.

Exporting All User records

Access the Work With User/Role Profiles form. From the System Administration Tools menu (GH9011), select User Management, User Profiles (P0092).

1. On the Work With User/Role Profiles form, click Find.
2. From the Form menu, select PIM Export All Users.
3. On the Report Output Destination form, make sure that the Export to CSV option is selected and click OK.

Exporting a Selected List of PIM User Records

Access the Work With User/Role Profiles form. From the System Administration Tools menu (GH9011), select User Management, User Profiles (P0092).

1. On the Work With User/Role Profiles form, click Find and then select the user records that have been set up for PIM synchronization.

   To determine which users have been set up for PIM synchronization, scroll to the PIM Server column. Records that have a value in this column have been set up for PIM synchronization.
2. From the Row menu, select PIM Export User(s).
3. On the Report Output Destination form , make sure that the Export to CSV option is selected and click OK.

   The system generates and displays a CSV file. Use this file to import PIM synchronization records into Intellisync Server.

Importing PIM Synchronization Files into Intellisync Server

To complete the PIM data synchronization with JD Edwards EnterpriseOne, use the Intellisync Server Bulk User Import tool to import these items into Intellisync Server:

- CSV file
- Synchronizing template

Note. The synchronizing template must match the template that you associated with JD Edwards EnterpriseOne users when you set up users for PIM synchronization.

See *EnterpriseOne Tools 8.96 PIM Sync Installation and Configuration*for information on how to import these files.

CHAPTER 6

Enabling Mobile Client Functionality

This chapter provides an overview of distributed next numbers for mobile clients and discusses how to set up distributed next numbers for mobile clients.

Understanding Distributed Next Numbers for Mobile Clients

Any of JD Edwards EnterpriseOne mobile applications that assign unique numbers using the number facilities (namely the X0010 and X00022 business functions) must have their parameters registered as distributed next numbers. These parameter sets are identified during application development. For applications that call X0010, these parameters include a system code, next numbering index, company, document type, century, and fiscal year. For applications that call X00022, the parameter is an object name.

Setting Up Distributed Next Numbers for Mobile Clients

This section discusses how to:

- Activate mobile client functionality.
- Define the starting next number.
- Enable distributed next numbers.
- Set up distributed next numbers by system code.
- Map the JD Edwards EnterpriseOne user ID to the Distributed Next Numbers application (P950411).
- Set up distributed next numbers by application.
- Schedule updates to distributed next numbers applications.

See Also

Setting Up System Next Numbers in the *JD Edwards EnterpriseOne Address Book 8.12 Guide*

Activating Mobile Client Functionality

Access the Work with EnterpriseOne System Control form. On the JD Edwards EnterpriseOne enterprise server, access the P99410 application.

1. On the Work with EnterpriseOne System Control form, click Add.
2. On the EnterpriseOne System Control Revisions form, enter each of the following system codes, one at a time, in the Data Item field:

- *SY90CA*
- *CRMMSL*
- *SY49*

The F99410 table must contain records for these system codes before you can enable mobile client functionality.

3. Click Yes.
4. Click OK.
5. Repeat these steps to add each system code.

Defining the Starting Next Number

On the JD Edwards EnterpriseOne enterprise server, access the Next Number Revisions application (P0002) through the JD Edwards EnterpriseOne web client.

1. On the Work with Next Numbers form, click Add.
2. Enter *42E* in the System field.
3. In the Use field, enter a description of the next number parameter.
4. In the Next Number field, enter the starting next number, for example *1*.
5. Select the Check Digit Used option if you want JD Edwards EnterpriseOne to append a random digit to the next number for uniqueness, and then click OK.

Enabling Distributed Next Numbers

On the JD Edwards EnterpriseOne enterprise server, access the P00098 application through the JD Edwards EnterpriseOne web client.

1. On the Work With Enable Distributed Next Numbers form, click Add.
2. Create separate rows for each of these product codes:

Product Code	Order Type Next Number	Enabled
42E	1	Y
47	2	Y
32	2	Y
01	1	Y

3. Create additional rows for the following tables:

Object Name	Enabled
F0111 (Address Book - Who's Who)	Y
F01112 (Related Person)	Y
F0115 (Address Book - Phone Numbers)	Y

Object Name	Enabled
F01151 (Electronic Address)	Y
F0450 (Payee Control)	Y
F42140 (Customer Master Commission Information)	Y
F42150Z1 (Header Commission Unedited Transaction File)	Y
F42160Z1 (Detail Commission Unedited Transaction File)	Y
F90CA06A (Competitor_KillSheet Table)	Y
F90CB010 (Lead Table)	Y
F90CB020 (Opportunity Table)	Y
F90CB02E (Opportunity - Employee Table)	Y
F90CB02J (Opportunity - Sales Cycle Table)	Y
F90CB030 (CRM Address Table)	Y
F90CB043 (Qualification Object Table)	Y
F90CB05A (Sales Cycle Table)	Y
F90CB05C (Sales Cycle Notification Ledger Table)	Y
F90CB060 (Forecast Table)	Y

4. Click OK.

Setting Up Distributed Next Numbers by System

On the JD Edwards EnterpriseOne enterprise server, access the P00023A application through the JD Edwards EnterpriseOne web client.

1. On the Work with Distributed Next Numbers by System Code form, select ZJDE0001.
2. Click Add.
3. On the Distributed Next Number form, for each JD Edwards EnterpriseOne mobile user, create the following additional rows for the different system codes, and then click OK:

Third-Party User ID	System Code	OrderType Next Number	Allotment
Enter the JD Edwards EnterpriseOne user ID of a mobile client	42E	1	500
	47	2	500
	32	2	500
	01	1	Assign a large enough allotment of next numbers so that users can enter enough new records before synchronizing with JD Edwards EnterpriseOne. **Note.** The mobile client is not allowed to add more records than specified by this allotment. When the mobile client synchronizes with the server, the user is granted a new allotment of next numbers.

Mapping the JD Edwards EnterpriseOne User ID to the Distributed Next Numbers Application (P950411)

On the JD Edwards EnterpriseOne enterprise server, access the P950411 application.

1. On the Work with Third Party User ID Map form, for each mobile user, click Add.
2. On the Third Party UserID Map Detail form, accept the default value (Synchronization Application) in the UserID Type field
3. In the User ID field, enter the user ID of the mobile client.
4. In the Environment Name field, enter the JD Edwards EnterpriseOne environment that you are currently signed on to.
5. Accept the default value in the Third-Party User ID field, and then click OK.

Setting Up Distributed Next Numbers by Application

On the JD Edwards EnterpriseOne enterprise server, access the P00023A application through the JD Edwards EnterpriseOne web client.

Note. You must make sure that you perform the steps in this task in the order that they appear. Particularly, you must enter the server user ID before you enter the JD Edwards EnterpriseOne user IDs.

---

1. On the Work with Distributed Next Numbers by Object Name form, select ZJDE0002 and click Add.
2. Enter the following information into separate rows:

Third-Party User ID	JD Edwards EnterpriseOne Table	Allotment
SERVER	F0111 (Who's Who)	6250
SERVER	F01112 (Related Person)	2500
SERVER	F0115 (Phone Numbers)	1250
SERVER	F01151 (Electronic Address)	1250

Note. The SERVER value must be entered in upper case.

3. Click OK.
4. Select ZJDE0002 and click Add.
5. For each JD Edwards EnterpriseOne mobile user, create rows for each of these JD Edwards EnterpriseOne tables:

Third-Party User ID	JD Edwards EnterpriseOne Table	Allotment
Enter the JD Edwards EnterpriseOne user ID of a mobile client.	F0111	Assign a large enough allotment of next numbers so that users can enter enough new records before synchronizing with JD Edwards EnterpriseOne. **Note.** The mobile client is not allowed to add more records than specified by this allotment. When the mobile client synchronizes with the server, the user is granted a new allotment of next numbers.
	F01112	
	F0115	
	F01151	
	F0450	
	F42140	
	F42150Z1	
	F42160Z1	
	F90CA06A	
	F90CB010	
	F90CB020	
	F90CB02E	
	F90CB02J	
	F90CB030	
	F90CB043	
	F90CB05A	
	F90CB05C	
	F90CB060	

6. Click OK.

Scheduling Updates to Distributed Next Numbers Applications

You should schedule the Distributed Next Number Update UBE (R0002D1) to run at a regular interval. This UBE is typically set up to run nightly.

CHAPTER 7

Setting Up Viewable and Configurable Portlets

This chapter provides an overview of viewable and configurable portlet types and discusses how to initialize viewable and configurable portlets.

Note. The SRM, HTML, URI, and IURI portlets that are discussed in this chapter are available in the JD Edwards Collaborative Portal and Oracle Portal.

Understanding Viewable and Configurable Portlet Types

In JD Edwards Collaborative Portal and Oracle Portal, you can set up viewable and configurable portlets for these types of portlets:

- Isolated URI (IURI)

  This type of portlet is the most flexible because it is not actually integrated into the Portal. IURI portlets allow you full access to the browser frame and to take existing web content and display it in the Portal.

- URI

  This type of portlet always requires interaction between the Portal and a web server. URI portlets can be written as CGI, Java servlets, or HTML pages, and they can reside on a server other than the Portal server. URI portlets tend to be slower than the other portlets.

- HTML

  This type of portlet requires no interaction with an external server, so it provides quick response with very little external resource load. HTML portlets can contain references to applets, ActiveX controls, and images, but the HTML is taken directly from JD Edwards EnterpriseOne database, so the content is static. You can take advantage of pass-through functions, such as RunOWApp and addRunOWAppFI with HTML portlets as well. These functions enable you to launch JD Edwards EnterpriseOne applications.

- Supplier Relationship Management (SRM)

  This portlet provides a fully configurable shell portlet that you can modify without Java programming. You can select the queries to display in the portlet at runtime using a wizard interface.

Viewable portlets can be initialized only to an existing portlet type, but may be initialized by anyone. After initializing a viewable portlet, users cannot modify its definition.

Configurable portlets enable you to initialize a portlet to an existing portlet type or create a new portlet definition. In addition, when you initialize a configurable portlet, you define permissions to determine the type of access users have to the configurable portlet. You can grant either View or View,Config access. View permissions allow users to use a configurable portlet, but prevents them from modifying the definition of the portlet. View,Config permissions allow users to modify the definition and the permissions of a portlet. A user who has View,Config permissions basically has the same permissions of a system administrator. Therefore, you should carefully consider the users to which you grant View,Config permissions.

If a user only has view access to a configurable SRM portlet, the user may still edit the portlet if it contains alert parameters that can be personalized. In this case, the SRM portlet displays an Edit icon that users can select to modify the alert parameters.

Initializing Viewable and Configurable Portlets

This section discusses how to:

- Initialize viewable HTML, URI, IURI portlets.
- Initialize configurable HTML, URI, IURI portlets.
- Initialize a viewable SRM Portlet.
- Initialize a configurable SRM portlet.

Prerequisites

If you are using the JD Edwards Collaborative Portal, make sure that the WebClient_Portal.war file that contains these portlets has been deployed on the IBM WebSphere Portal.

If you are using the Oracle Portal, make sure that the webclient_Oracle_Portal.war file has been deployed as a portal provider and that the provider has been registered with the Oracle Portal.

Initializing Viewable HTML, URI, IURI Portlets

You can initialize viewable HTML, URI, IURI portlets only to an existing portlet type.

Access the portal page that contains the viewable HTML_URI_IURI Component portlet.

1. In the portlet, click the Configure icon.
2. Under Configure Portlet Choice, select an existing portlet from the drop-down list box and then click the Done button.

   The initialized portlet appears on the Portal page.

Initializing Configurable HTML, URI, IURI Portlets

Access the portal page that contains the Configurable_HTML_URI_IURI_Component portlet.

1. In the portlet, click the Configure icon.
2. If you want to use an existing portlet definition, perform these steps:
   a. Select the Select an existing portlet option.
   b. Select an existing portlet from the drop-down list box, and then click the Done button.
3. To create a new portlet definition, select the Create a new portlet of portlet type option, and then select HTML, URI, or IURI from the drop-down list box.
4. Click the Next button and then complete the fields according to the type of portlet selected:

Portlet Type	Fields	Description
HTML	Name	Enter a unique ID for the portlet. The maximum field length is 10 characters. This ID must be unique across the system.
	Title	Enter a name for the portlet. This is the name that appears when the system provides a list of portlets for a user to select from. The maximum field length is 30 characters.
	HTML Code	Enter or edit existing HTML for this component.
URI	Name	Enter a unique ID for the portlet. The maximum field length is 10 characters. This ID must be unique across the system.
	Title	Enter a name for the portlet. This is the name that appears when the system provides a list of portlets for a user to select from. The maximum field length is 30 characters.
	Support Cookies	Indicates whether the component can support cookies.
	Component URL	Enter a URL to existing web content. Select one of the following protocols from the drop-down list box and then enter the remaining path for the URL: • http:// • https:// Select none if the URL is relative to the context of the portlet.
IURI	Name	Enter a unique ID for the portlet. The maximum field length is 10 characters. This ID must be unique across the system.
	Title	Enter a name for the portlet. This is the name that appears when the system provides a list of portlets for a user to select from. The maximum field length is 30 characters.
	Width	Select the percent or pixels option, and then enter the appropriate value to determine the width of the component IFRAME.
	Height	Indicates the height in pixels of the component IFRAME.
	Scrolling	Select Yes to always display scrollbars in the IFRAME. Select No to never display scrollbars. Select Auto to allow the IFRAME to display scrollbars if they are needed.
	Component URL	Enter a URL to existing web content. Select one of the following protocols from the drop-down list box and then enter the remaining path for the URL: • http:// • https:// Select none if the URL is relative to the context of the portlet.

5. Click the Next button.
6. In the Add Permissions region, select the type of permission that you want to assign from the Permissions drop-down list box.
   - View

     Select this option to grant only view access to the portlet.
   - View,Config

     Select this option to grant users the capability to re-configure the portlet definition and permissions.
7. In the Who group box, select the option according to the users to which you want to assign permissions and complete the fields accordingly:
   - User
   - Role
   - *PUBLIC
8. Select the Update button to add the permissions record.

   The record appears in the Edit/Remove Permissions region.
9. In the Edit/Remove Permissions region, you can modify or remove permissions.
10. Select the Done button to complete the portlet initialization.

Initializing a Viewable SRM Portlet

You can initialize viewable SRM portlets only to an existing portlet.

Access the portal page that contains the viewable SRMComponent portlet.

1. In the portlet, click the Configure icon.
2. Under Supplier Relationship Management Portlet Choice, select an existing portlet from the drop-down list box.

   The initialized portlet appears on the portal page.

   An initialized viewable SRM portlet may contain an edit icon if the portlet contains an alert that a user can personalize.

Initializing a Configurable SRM Portlet

Access the Portal page that contains the Configurable_ SRMComponent portlet.

1. In the Configurable_SRMComponent portlet, click the Configure icon.
2. If you want to use an existing portlet definition, perform these steps:
   a. Select the Select an existing portlet option.
   b. Select a portlet from the drop-down list box, and then click the Done button.
3. To create a new SRM portlet definition, select the Create a new portlet option, and then click the Next button.
4. For a new portlet definition, perform the necessary steps to select and configure the SRM alerts.

   See Configuring the SRM Portlet Using the Configuration Wizard in the *JD Edwards EnterpriseOne 8.12 Supplier Relationship Management Collaboration Guide.*

After you select the SRM alerts and configure the alert parameters, the portal displays the page that you use to define permissions for the portlet.

5. In the Add Permissions region, select the type of permission that you want to assign from the Permissions drop-down list box.
   - View

     Select this option to only allow users to view the portlet.
   - View,Config

     Select this option to grant users the capability to re-configure the portlet definition and permissions.
6. In the Who group box, select the option according to the users to which you want to assign permissions and complete the fields accordingly:
   - User
   - Role
   - *PUBLIC
7. Click the Update button to add permissions.

   The record appears in the Edit/Remove Permissions region.
8. In the Edit/Remove Permissions region, you can modify permissions or remove permissions.
9. Click the Done button to complete the portlet initialization.

CHAPTER 8

Dropping Indexes from a Table

This chapter provides an overview of the Index Selection Tool (P95150) and discusses how to use the Index Selection Tool.

Understanding the Index Selection Tool

Database administrators drop indexes from tables in a physical database to increase system performance. Instead of using the actual database application to drop table indexes, you can use the Index Selection Tool (P95150). The advantage of using P95150 to drop indexes is that Oracle's JD Edwards EnterpriseOne keeps track of the indexes that are dropped, even if the indexes have been restored or rebuilt. For example, during a software update, the system restores indexes to tables that are affected by the update. Typically, after an update, you would have to perform another analysis to determine the indexes that need to be re-dropped. However, JD Edwards EnterpriseOne retains this information so that you can easily find the indexes that were previously dropped, reducing the time it takes to drop the indexes again.

Note. P95150 should only be used by system administrators or database administrators. Use Security Workbench (P00950) to apply the proper security to this application.

See *JD Edwards EnterpriseOne Tools 8.96 Security Administration Guide*, "Using Security Workbench".

Flagging indexes

The P95150 program enables you to locate and display the table indexes in a tree structure. You expand the tree to view and flag the indexes that you wish to drop. You can flag a single index or multiple indexes at a time. The P95150 program only allows you to drop indexes that are not unique or primary to the table since dropping these types of indexes would disrupt the integrity of the table.

When you open a table in P95150, the program displays the table and all of its columns in a tree. Each item, or node, in the tree has an icon next to it. The Tree Node Key tab contains a description of each of these icons, which include:

- Table
- Data Source Override
- Primary Index
- Unique Index
- Unflagged Index

  The Unflagged Index is the only type of index that you can flag or drop. The system does not permit you to drop a primary or unique index.
- Index Flagged for Drop

- Column in Index

Determining the Data Source

A table can reside in multiple data sources, so you must determine the data source from which you want the table index to be dropped. When you initially open a table in P95150, by default, the system displays a table in a Default node. A table in the Default node is not associated with a data source. If you flag and then try to drop an index from this node, the program prompts you to specify a data source. If you specify a data source before flagging and dropping an index, the program displays another node besides the Default node. This node is named after the data source that you selected and contains the same tree as the Default node. When you drop an index from the data source node, the index is automatically dropped.

Index Selection Logs

The P95150 program contains an option that enables you to view the history of actions that have been performed using the tool. You can view the following information in the index selection logs:

- Table name
- Data Source
- Index ID
- Log Description
- User
- Machine
- Date Updated
- Time Updated

Using the Index Selection Tool

This section discusses how to:

- Add a table to the Index Selection Tool.
- Select a data source.
- Drop a single index from a table.
- Drop multiple indexes from a table.
- Restore indexes to a table.
- View the Index Selection Logs.

Forms Used with the Index Selection Tool

Form Name	FormID	Navigation	Usage
Machine Search & Select	W95150H	In the Solution Explorer Fast Path, enter P95150.	Provide the location of the data source master table that the Index Selection Tool validates data sources against.
Index Selection Tool	W95150A	On Machine Search & Select, select a row and then click Select.	Flag, drop, and restore table indexes.
Choose OCM or Regular Data Source	W95150D	On Index Selection Tool, click the search button in the Data Source field.	Choose the method in which you want to select a data source.
Select Data Source By Object Configuration Manager	W95150C	On Choose OCM or Regular Data Source, select the "Select data source by OCM (recommended)" option and click OK.	Select the data source that the table is mapped to by OCM.
Database Data Source Search and Select	W986101WB	On Choose OCM or Regular Data Source, select the "Select data source from list of all data sources" option and click OK.	Select a data source from which you want to drop table indexes.
Add Table to Index Selection Tool	W95150B	On Index Selection Tool, click the Add button.	Select a table for flagging or dropping indexes.
Index Selection Logs	W95150I	On Index Selection Tool, from the Form menu, select Index Selection Logs.	View the Index Selection Logs.

Adding a Table to the Index Selection Tool

Access the Machine Search & Select form.

1. Click the search button in the Data Source column to find the data source master table that you want the Index Selection Tool to validate data sources against.

   Once you select a table, the system automatically displays the machine in which the table resides in the Machine Name column.

2. Click Select to continue.
3. On the Index Selection Tool, click the Add button.
4. On the Add Table to Index Selection Tool form, select a table and click OK.

   The left pane displays the table in a default node tree.

Selecting a Data Source

Before you drop an index from a table, you can select a specific data source location for the table. When you use this method, the Index Selection Tool tree displays the data source in a node for the table.

Access the Index Selection Tool form.

1. Select a table in the tree and then click the Add Data Source button in the center column.
2. On Choose OCM or Regular Data Source, click one of these options and then click OK to continue:
   - Select data source by OCM (recommended)

     Use this option to ensure that the table exists in any data source that you select.
   - Select data source from list of all data sources

     If you select this option, you must select a data source from the Database Data Source Search and Select form, which lists all available data sources.
3. If you selected the "Select data source by OCM (recommended)" option, on Select Data Source by Object Configuration Management, click one of these options and then click the search button in the Data Source column:
   - List data sources that the table is specifically mapped to by OCM

     Use this option to select from a list of data sources that the table has been specifically mapped to.
   - List data sources that the table is specifically and by "DEFAULT" mapped by OCM

     Use this option to select from a list of data sources that the table has been specifically mapped to, as well as mapped to by default.
4. Click Select.

   The Index Selection Tool displays the data source in a new node under the table.

Dropping a Single Index from a Table

Access the Index Selection Tool form.

The instructions in this section assume that you have already selected a table and a specific data source from which you want to drop the table index.

1. Expand the data source node to view the indexes for the table.
2. To flag an index, select an index and then click the Flag/Unflag Index button.
3. To drop a single index, select a flagged index and then click the Drop Single Index button.
4. On Drop Single Index, click one of these options and then click OK:
   - Use proxy user and password
   - Specify user and password

     This option activates the Owner ID and Password fields, which you must complete before clicking OK.
5. On the dialog box, click OK to continue.

Dropping Multiple Indexes from a Table

Access the Machine Search & Select form.

The instructions in this section assume that you have already selected a table and a specific data source from which you want to drop indexes.

1. From the data source node, select an index, and then click the Flag/Unflag Index button. Repeat until you have flagged all of the indexes that you want to drop from the table.

2. To drop multiple indexes, select the data source node that contains the indexes that are flagged to be dropped, and then click the Update Table Indices button.
3. On Update Table Indices, click one of these options and then click OK:
   - Use proxy user and password
   - Specify user and password

     This option activates the Owner ID and Password fields, which you must complete before clicking OK.

Restoring Indexes to a Table

Access the Index Selection Tool form.

1. Locate the table and data source that you want to restore indexes to.
2. Expand the data source node to view the indexes that have been dropped.
3. If you want to restore a single index, select the index and then click the Create Single Index button.
4. If you want to restore all of the indexes to the table, select the appropriate data source node and then click the Update Table Indices button.

Viewing the Index Selection Logs

Use the Index Selection Logs to view all of the actions that you have taken when working with table indexes.

Access the Work with Index Selection Logs form.

1. Enter a table name in the Table Name field or enter an asterisk (*) to view all the tables that have been acted upon in the Index Selection Tool.
2. Click Find.

   The program displays the records for all actions taken based on the search criteria. Each record contains this information:
   - Table Name
   - Data Source
   - Index ID
   - Log Description
   - User
   - Machine
   - Date Updated
   - Time Updated

CHAPTER 9

Using Interactive Versions for Applications

This chapter provides an overview of interactive versions for applications and discusses how to work with interactive versions.

Understanding Interactive Versions for Applications

In JD Edwards EnterpriseOne applications, a version is a user-defined set of specifications. These specifications help to control how interactive applications run. Interactive versions are associated with applications (usually as a menu selection) and always run on a workstation.

Interactive versions for applications contain processing options with different sets of data for each version. These processing options are passed to the application when it runs.

Versions enable you to modify the behavior of applications because they exist independently of the application. Typically, administrators control the creation, modification, and location of the actual version tables. When you upgrade JD Edwards EnterpriseOne software applications to a new release level, you can apply the existing versions to the new applications.

When a user starts an interactive application, the user might have the option to select from a list of versions. A user only has this option if the application designer attached processing options to the application. If the system administrator sets the version for blind execution, when the user starts the application, the application uses the saved processing option values for the version without prompting you for new processing option values. If the system administrator sets the version for Prompt for Values, the user will be prompted to enter processing option values. Depending on how you assign security to your JD Edwards EnterpriseOne software applications, end users can select or create different versions based on business requirements.

For example, on the System Administration Tools menu (GH9011), the Interactive Versions option (P983051) does not have processing options attached, so a version does not exist for the application. However, the Work With Servers application (P986116) has processing options attached so that the system administrator must attach a version for the application. Otherwise, the application uses the default. For each interactive application, the system administrator can set up multiple versions that contain different processing options values for each version.

How Processing Options Affect Versions

The processing options that you define in versions are a set of parameters that alter how an application runs. They are similar to initialization (.ini) files and command-line arguments for a traditional executable. These processing options let you specify the options that you want when you open an application. For example, you can specify the appearance of a form, show or hide a field, change the default status for order activity rules, and set default information to appear in a field. Depending on how the developer coded the application, the following functionality could be available:

- Change the functionality of an application. For example, you can set up processing options to select or deselect logic in order holds. You also can specify whether you want to automatically print pick slips after you enter an order.
- Change default values. For example, in Sales Order Entry you can set up processing options to set default values for document type values (such as Sales Order or Sales Quote) or line type (such as stock or nonstock item).
- Control the display of forms. For example, you can set processing options to hide or show a cost field, a price field, or a commission field.

Not all JD Edwards EnterpriseOne applications have processing options. If the Prompt For Values option on the Edit menu is grayed out, either no processing options are associated with the application or the system administrator has disabled the processing options. To use versions with an application, you must first attach processing options to the interactive application.

The system administrator can secure a version for an application. In this case, the Prompt For Version option on the Edit menu appears grayed out. When a user opens a secured version of an application from the Work With Interactive Versions application, a security message appears to alert the user that she or he does not have access to the version.

How Interactive and Batch Versions Differ

Interactive versions have processing options. Batch versions have processing options, data selection and sequencing, and template overrides. You do not check in and check out interactive versions, whereas batch versions have local specifications that must be checked in and checked out.

Working with Interactive Versions

This section provides an overview of how to work with interactive versions and discusses how to:

- Work with version detail for interactive versions.
- Copy an interactive version.
- Create an interactive version.

Understanding How to Work with Interactive Versions

When you work with interactive versions, you change processing options, version detail, and copy or create versions. You can also review information such as the date the version was last modified and the user who performed the modification. Interactive versions must be associated with a menu selection in order to run the specified version.

In JD Edwards Solution Explorer, you can display the available versions of an interactive application by choosing Prompt for Version from the Edit menu or by choosing the Interactive Versions (P983051) menu selection. You can filter the versions that you want to display to show all versions or only your versions. To filter the version display, select the Display option from the Form menu.

Working with Version Detail for Interactive Versions

Access the Work With Interactive Versions form. Enter *IV* in the Fast Path.

Note. Depending on your security level and the level of security for the version, you might not be able to change the version detail information.

1. Enter an application ID in the Interactive Application field and click Find.

   For example, to locate a version for the Sales Order Entry application, enter P42101.
2. In the grid, select a version with which to work, and then from the Row menu, select Version Detail.
3. On the Interactive Application Version Details form, you can modify the information in these fields:

Field	Description
Version Title	Enter information that describes the use of a version in this field. This field should describe the use of a version. For example, an application for generating pick slips might have a version called Pick Slips - Accounting and another version called Pick Slips - Inventory Management.
Prompting	Enter a value to determine whether the processing options for the version are disabled, run with blind execution, or chosen by the user at run time.
Security	Enter a value to determine the security for the version, ranging from no security to full security. This security is based on the user and is not related to the application security. Depending upon your security level and the level of security for the version, you might not be able to work with version detail.

4. Review the additional information that appears on the form, as needed.

Copying an Interactive Version

You can copy an existing version and then tailor its information to fit your needs. The copied version inherits the processing option values of the existing version.

When you copy a version, you should add security to the new version. Security settings range from none (anyone has the authority to modify or run the version) to full (only the person who last modified the version can modify and run the version).

Note. Depending on your security level and the level of security for the version, you might not be able to copy the version.

Access the Work With Interactive Versions form. Enter *IV* in the Fast Path.

1. Enter an application ID in the Interactive Application field and click Find.
2. In the grid, select a version with which to work, and then click Copy.
3. On the Copy Interactive Application Version form, complete these fields and then click OK:

Field	Description
New Version	Enter a unique identifier for this version of the application in this field.
Version Title	Enter information that describes the use of a version in this field.
Security	Enter a value to determine the security for the version, ranging from no security to total security.

4. On the Interactive Version Design form, click one of these buttons:

Option	Description
Revise Version	Click this option to access the Version Detail form and modify this version. See Chapter 9, "Using Interactive Versions for Applications," Working with Version Detail for Interactive Versions, page 50.
Processing Options	Click this option to change the processing options for the version.
Run Local Web	Click this option to run the version.

5. Click OK when you are finished modifying the interactive version.

Creating (Adding) an Interactive Version

You can create (add) a new interactive version that is not based on an existing version. When you create an interactive version, you should add security to the new version. Security settings range from none (anyone has the authority to modify or run the version) to full (only the person who last modified the version can modify and run the version).

Access the Work With Interactive Versions form. Enter *IV* in the Fast Path.

1. Enter an application ID in the Interactive Application field.

   For example, to add a version for the Sales Order Entry application, enter P42101.
2. Click Add to create a new version.
3. On the Version Add form, complete these fields, and then click OK:

Field	Description
Version	Enter a unique identifier for this version of the application in this field.
Version Title	Enter information that describes the use of a version in this field.
Prompting Options	Enter a value to determine how the version assigns processing options, such as no processing options or blind execution, or prompts the user to select options at runtime. Blank is not a valid value when you add a version. All versions for interactive applications must have processing options attached.
Security	Enter a value to determine the security for the version, ranging from no security to full security.

4. On the Interactive Version Design form, click one of these options:

Option	Description
Revise Version	Click this option to access the Version Detail form and modify this version. See Chapter 9, "Using Interactive Versions for Applications," Working with Version Detail for Interactive Versions, page 50.
Processing Options	Click this option to change the processing options for the version.
Run	Click this option to run the version.

5. Click OK when you are finished adding the interactive version.

CHAPTER 10

Working with User Defined Codes

This chapter provides overviews of user defined codes (UDC), UDC and UDC type customization, UDC tables, and discusses how to:

- Customize UDCs.
- Translate UDCs into alternate languages

Understanding User Defined Codes

Most forms in JD Edwards EnterpriseOne contain fields. Some fields enable you to enter any value, and some require you to select from a list of values. A user defined code (UDC) is one value in a set of values that is assigned as valid for a field. You can use UDCs to categorize your data and make sure that users provide consistent input on forms. Because users can select only values from the list, UDCs simplify, standardize, and validate the data that is contained in fields.

From any JD Edwards EnterpriseOne application, you can identify fields that have UDCs attached to them by using the visual assist button that appears when you tab into or click in a field. If you do not know the value to enter in a field with a user defined code attached to it, click the visual assist button, which accesses the Select User Define Code form. This form displays all values contained in the user defined code tables for this field. You can then select the value to use.

JD Edwards EnterpriseOne provides predefined UDCs, but many of the UDCs that you will use are unique to your enterprise, and your needs are likely to change. Therefore, the system lets you change, add, and delete UDCs to meet the needs of your enterprise. When you upgrade JD Edwards EnterpriseOne software, your customized UDCs will remain.

UDCs, UDC Types, and Category Codes

A UDC is one value in a set of values that you have assigned as valid for a field. A UDC is made up of two parts. The first part is the code, which consists of the characters that you enter in a field. The second part is the description, which is text that describes what the code means. For example, on the Work With Addresses form, you can enter A in the Search Type field to designate an Applicant; this code becomes part of the data stored with the record.

A UDC type is the complete set of UDCs that are allowed as values for a field. A UDC type is made up of a code type, which is its two-character name, and a description. Examples of UDC types are ST - Search Type and UM - Unit of Measure. UDC types are sometimes referred to as UDC lists.

Each UDC type is associated with a JD Edwards EnterpriseOne product code. You can identify any set of UDCs by its product code and its code type. For example, the Address Book (product code 01) list of search types (code type ST) is referred to as UDC 01/ST.

Throughout the system, you will see references to category codes. Category codes are UDC types that JD Edwards EnterpriseOne provides for you to customize according to your needs. You can change the code type and the description, and you can redefine the UDCs as appropriate for your purposes. For example, you might see a UDC type called Category Code 01. You can change its description and define the UDCs within it to suit your business needs.

Example: User Defined Codes in Address Book

JD Edwards EnterpriseOne systems use UDCs. For example, Address Book uses a field called Search Type to classify the entries in the address book. When you click the visual assist button in the Search Type field on the Work With Addresses form, a list of the search types appears. These search types are UDCs. Some of the search types include the following:

- A - Applicants
- C - Customers
- V - Suppliers
- E - Employees

You can use these UDCs to classify your address book entries, and you can add or change UDCs to accommodate your needs. For example, if you need to categorize some of your address book entries as students, you can add a UDC to the list, such as S - Students.

UDCs are also used to supply values for the following:

- State and province codes
- Units of measure
- Document types
- Languages

When you click the visual assist for a field and the system displays the Select User Defined Code form, you know that you are working with UDCs.

Understanding UDC and UDC Type Customization

JD Edwards EnterpriseOne provides many UDC types that contain predefined UDCs. Some of the UDCs are hard-coded, which means that certain applications depend on specific values, so you should not change these UDCs. However, if a UDC is not hard-coded, you can change it to suit your business needs.

Many of the UDCs that you need to use are unique to your enterprise, and your needs are likely to change. Therefore, you can change, add, and delete UDCs. UDCs enable you to customize JD Edwards EnterpriseOne software to meet your needs without having to write complex programs or modify the system code. When you upgrade JD Edwards EnterpriseOne software, your customized UDCs remain.

You can change, add, and delete UDCs in these ways:

- Change the code or the description of a UDC in an existing UDC type. For example, in UDC 01/ST, a medical institution might change the UDC for C - Customers to P - Patients to more accurately describe the category.
- Add UDCs to an existing UDC type. For example, in the Search Type list, you might add a UDC for S - Students.

- Delete UDCs from a UDC type. For example, if you want to prevent users from choosing a UDC, you can delete it from the UDC type.

You can also change, add, and delete UDC types in the following ways:

- Change the code type or the description of an existing UDC type, which is useful if you want to customize one of the generic category code lists for your needs.
- Create a new UDC type and add UDCs to it. For example, an educational institution might create a UDC type called MA - Major Field of Study to classify its students, and it might define the following UDCs:
- LA - Liberal Arts
- MA - Mathematics
- CS - Computer Science
- EN - Engineering
- MD - Medicine

Consequences of Customizing UDCs

Because UDCs can significantly affect the integrity of your data, you should customize them only as part of a coordinated plan within your enterprise. When you add or change a UDC, you are affecting the set of values against which the system validates the data entry. However, you are not affecting the actual content of any existing data records. By changing UDCs within a working production environment, you might affect the integrity of your data. For example, you use Address Book to enter address book records, and you use search types to classify those records. Suppose that you select a search type of C - Customers to classify some of the records that you enter. Later, you decide to change that UDC from C - Customers to P - Patients. Any address book records that you entered with the original UDC value of C will still contain that value. When Address Book displays these records, you will see an error in the Search Type field because C is no longer a valid value.

Understanding UDC Tables

You use the User Defined Codes program (P0004A) to create and customize UDCs and UDC types. The program stores UDC information in the following tables:

- User Defined Code Types (F0004)
- User Defined Codes (F0005)

Prerequisite

Every UDC belongs to a UDC type. Verify that a UDC type exists where you will add the UDC.

See Chapter 10, “Working with User Defined Codes,” Adding a UDC Type, page 62.

Customizing User Defined Codes

A UDC is made up of two parts. The first part is the code, which consists of the characters that you enter in a field. The second part is the description, which is text that describes what the code means. You can change both the code and the description. For example, the UDC list of search types contains the code C, which designates Customers. A medical facility might change this code and description to P for Patients.

This section discusses how to:

- Change a UDC.
- Add a UDC.
- Delete a UDC.

Change a UDC

Access the Work With User Defined Codes form. On the JD Edwards EnterpriseOne web client, from a System Setup menu for your product, access the P0004A program.

1. Complete the Product Code and User Defined Codes fields and click Find.

   For example, to display the list of Address Book search types, which is UDC 01/ST, type 01 in the Product Code field and ST in the User Defined Codes field.
2. Select the code that you want to modify and click Select.
3. On the User Defined Codes form, modify any of these fields and click OK:

Field	Description
Codes	A user defined code (98/SY) that identifies a system. Valid values include: 01 Address Book 03B Accounts Receivable 04 Accounts Payable 09 General Accounting 11 Multicurrency
Description 01	A code that identifies the table that contains user defined codes. The table is also referred to as a UDC type.

Field	Description
Special Handling	A code that indicates special processing requirements for certain user defined code values. The value that you enter in this field is unique for each user defined code type. The system uses the special handling code in many ways. For example, special handling codes defined for Language Preference specify whether the language is double-byte or does not have uppercase characters. Programming is required to activate this field.
Hard Coded	A code that indicates whether a user defined code is hard-coded. Valid values are: Y The user defined code is hard-coded N The user defined code is not hard-coded A check mark indicates that the user defined code is hard-coded.

Adding a UDC

Add a UDC to a UDC type when none of the existing codes is appropriate for your needs. For example, if you need to identify the entries in the address book that are your business partners, you can add a search type B - Business Partners to UDC 01/ST.

Access the Work With User Defined Codes form. On the JD Edwards EnterpriseOne web client, select the appropriate program for changing UDCs.

1. Complete the Product Code and User Defined Codes fields and click Find. In the User Defined Codes field, enter the UDC type for which you want to add the UDC.
2. On the Work With User Defined Codes form, click Add.
3. On the User Defined Codes form, scroll to the last empty row of the detail area.

Important! Be sure to add each new code on the *last* detail row so that you do not inadvertently overwrite a blank code, which might appear in the first detail row. A blank code might have only a period in the Description field.

---

4. Enter a code in the Codes field. To allow a blank as a valid value, leave this field blank.
5. Enter a description in the Description 1 field. To allow a blank as a valid value, type any character (such as a period) in the last space in this field.
6. Complete the Special Handling and Hard Coded fields, and then click OK.

Deleting a UDC

You can delete UDCs from a UDC type, but do so with caution. Only delete UDCs as part of a coordinated plan within your enterprise. For example, you might delete the F - Facilities UDC from the list of search types if you do not want users to select that UDC.

If you delete a UDC, the system deletes only the code from the UDC type. UDC values in existing records are not deleted.

Important! Do not delete hard-coded UDCs because JD Edwards EnterpriseOne applications might depend on them. Hard-coded UDCs have the value Y in the Hard Coded field on the Work With User Defined Codes form.

Access the Work With User Defined Codes form. On the JD Edwards EnterpriseOne web client, from a System Setup menu for your product, access the P0004A program, or enter *UDC* in the Fast Path field.

1. Complete the Product Code and User Defined Codes fields and click Find.
2. On the Work With User Defined Codes form, select the UDC from the detail area that you want to delete and click Delete.

   Important! Ensure that you want to delete this UDC. The only way to replace a deleted UDC is to add it again.

   ---

3. Click OK to confirm that you want to delete the UDC.

Customizing UDC Types

This section provides overviews of how to customize UDC types and add UDC types and discusses how to:

- Change a UDC type.
- Add a UDC type.
- Delete a UDC type.

Understanding How to Customize UDC Types

A UDC type is the complete set of UDCs that is allowed for a field. A UDC type is made up of a code type, which is its two-character name, and a description. Examples of UDC types are search types and units of measure. UDC types are sometimes referred to as UDC lists.

Each UDC type is associated with a JD Edwards EnterpriseOne product code. You can identify any set of UDCs by its product code and its code type. For example, the Address Book (product code 01) list of search types (code type ST) is referred to as UDC 01/ST.

You can change the code type and the description of an existing UDC type to meet your needs. Typically, you would change only the description so that it provides a meaningful description of the UDCs within the UDC type. For example, to classify your customers according to how much business they provide, you can change the description for Category Code 01 to Customer Volume. Then, you can customize the individual UDCs within that UDC type to describe the following classifications for your customers:

- H - High-volume customer
- M - Medium-volume customer

- L - Low-volume customer

You can change the code type, but you should do so with caution. If you change a code type, you could invalidate any existing records that use the original code type.

You can also follow this procedure to see a complete list of UDC types for a product code.

Understanding How to Add UDC Types

Add a UDC type when you need to categorize your data using UDCs and when none of the existing UDC types is appropriate. For example, an educational institution might add a UDC type called "Major" to categorize its students by any of the following fields of study:

- LA - Liberal Arts
- MA - Mathematics
- CS - Computer Science
- EN - Engineering
- MD - Medicine

Note. When you add a UDC type, you also must modify the JD Edwards EnterpriseOne applications that use the UDC type. See Understanding Edit Controls in the *Form Design Aid Guide* for more information about associating a UDC type with a field.

Because modifying a JD Edwards EnterpriseOne application might require significant effort, whenever possible, you should change an existing UDC type (such as a category code) instead of adding a new UDC type. See Changing a UDC Type in this guide.

Prerequisite

Delete all individual UDCs from the UDC type.

See Chapter 10, "Working with User Defined Codes," Deleting a UDC, page 60.

Changing a UDC Type

Access the Work With User Defined Codes form. From a System Setup menu for your product, select the appropriate program for changing UDCs; or enter *UDC* in the Fast Path field.

1. Select Code Types from the Form menu.
2. On the Work With User Defined Code Types form, complete the Product Code field and click Find.

   The system displays the UDC types that exist for that product code.
3. Select the UDC type that you want to change and click Select.
4. On the User Defined Code Types form, change the values in any of these fields and click OK:

Field	Description
Code Types	Enter a code that identifies the table that contains user defined codes. The table is also referred to as a UDC type. **Important!** It is suggested that you do not change code types. If you change a code type, you might invalidate existing records that use the original code type.
Description	Enter a description for the UDC type.
Code Length	Enter the length of the user defined code. It cannot be greater than 10 characters.
2nd Line (Y/N)	Determines if the Select User Defined Code form will display a second line of description. Valid values are: Y Enables the second line of description. M For maintenance only for second line display. This capability is seldom used, but has applicability in areas such as inventory product codes. The M value will not display the second line of description in the Select User Defined Code form. N Enables the Select User Defined Code form to display only one line of description.
Numeric (Y/N)	Determines whether a user defined code is numeric or alphanumeric. Valid values are: Y Indicates that the code is numeric and should be right-justified. N Indicates that the code is alphanumeric and should be left-justified.

Adding a UDC Type

Access the Work With User Defined Codes form. From a System Setup menu for your product, select the appropriate program for changing UDCs; or enter UDC in the Fast Path field.

1. Select Code Types from the Form menu.
2. On the Work With User Defined Code Types form, complete the Product Code field and click Find.
3. Click Add.
4. On the User Defined Code Types form, scroll to the last empty row of the detail area, complete these fields, and then click OK:

Deleting a UDC Type

You can delete a UDC type, but you should do so with caution. JD Edwards EnterpriseOne applications and the integrity of the data within your database might depend on the existence of UDCs and UDC types. Only delete UDC types as part of a coordinated plan within your enterprise.

Important! Do not delete UDC types that contain hard-coded UDCs because JD Edwards EnterpriseOne applications might depend on them. Hard-coded UDCs have the value Y in the Hard Coded field on the Work With User Defined Codes form.

Access the Work with User Defined Codes form. From a System Setup menu for your product, access the P0004A program, or enter *UDC* in the Fast Path field.

To delete a user defined code type:

1. On the Work With User Defined Codes form, select Code Types from the Form menu.
2. On the Work With User Defined Code Types form, complete the Product Code field and click Find:
3. On the Work With User Defined Code Types form, select the code type that you want to delete and click Delete.

   Important! Ensure that you want to delete this code type. The only way to replace a deleted UDC type is to add it again.

4. Click OK to confirm that you want to delete the code type.

Translating User Defined Codes into Alternate Languages

This section provides an overview on how to translate UDCs into alternate languages and discusses how to:

- Translate UDC type descriptions into alternate languages.
- Translate UDC descriptions into alternate languages.

Understanding How to Translate User Defined Codes into Alternate Languages

Multinational enterprises can translate the descriptions for both UDCs and UDC types into alternate languages. The system displays the descriptions in the language designated by the user's language preference. For example, you can provide a translated description for this UDC:

- Code: E
- English Description: Employees
- Spanish Description: Empleados

In this way, users can select the same UDCs, regardless of their language preference.

The User Defined Code Alternate Descriptions program (P0004D) stores the translated descriptions in the following tables:

- User Defined Codes - Alternate Language Descriptions (F0004D)
- User Defined Codes - Alternate Language Descriptions (F0005D)

Translating UDC Type Descriptions into Alternate Languages

Access the Work With User Defined Codes form. From a System Setup menu for your product, access the P0004A program, or enter UDC in the Fast Path field.

1. Select Code Types from the Form menu.
2. On the Work With User Defined Code Types form, complete the Product Code field and click Find.
3. Select the UDC type that you want to translate and then select Language from the Row menu.
4. On the UDC Alternate Languages form, enter information in a blank row for these fields and click OK:

Field	Description
L	Select the user defined code (01/LP) that specifies the language to use on forms and printed reports. Before you specify a language, a code for that language must exist at either the system level or in your user preferences.
Description	Type the translated description into this field.

Translate UDC Descriptions into Alternate Languages

Access the Work With User Defined Codes form. From a System Setup menu for your product, access the P0004A program, or enter UDC in the Fast Path field.

1. On the Work With User Defined Code form, complete the Product Code and User Defined Codes fields and click Find.
2. Select the code that you want to translate and then select Language from the Row menu.
3. On the UDC Value Alternate Descriptions form, enter the following information in a blank row in the grid, and then click OK:

Field	Description
L	Select the user defined code (01/LP) that specifies the language to use on forms and printed reports. Before you specify a language, a code for that language must exist at either the system level or in your user preferences.
Description	Type the translated description into this field.

CHAPTER 11

Working with User Overrides

This chapter provides an overview of user overrides and discusses how to work with user overrides.

Understanding User Overrides

This section provides an overview of user overrides and discusses:

- Search hierarchy
- Cached override information

User Overrides Overview

User Overrides enable you to change the appearance of an application to fit the needs of your business. For some user overrides, such as an object linking and embedding (OLE) attachment to a form, no consequences exist when you upgrade your software because your user overrides are merged into the new release. For other user overrides, such as grid or toolbar formats, the system gives you the options of reconciling your user overrides with the new software or deleting your user overrides.

A user override changes only the appearance of an application; it does not affect the application's functions. You can set overrides by user ID, role, or the keyword *PUBLIC. If you set your override by user ID, only that user is affected when he or she signs on to any workstation in the enterprise. If you set your overrides by role, those users who are set up within User Profiles to be members of that group are affected. If you set your user overrides by *PUBLIC, all users in the enterprise are affected. After you create an override for a role or *PUBLIC, each employee inherits that override, regardless of the workstation he or she signs on to. Even with role or *PUBLIC overrides, each employee can further customize a version to fit individual needs. For example, if someone in a department has trouble seeing text on a form, he or she can switch to a larger font. This change applies only to the individual, not to the entire group or company.

Oracle's JD Edwards EnterpriseOne stores these modifications in the User Overrides Table (F98950). Because the system tracks the overrides by your user ID or role, the modifications will appear on any workstation that you sign on to. User overrides enable you to make the following modifications:

- Resequence the grid.
- Change the sort order of rows and columns.
- Freeze columns and rows.
- Move and resize columns and rows.
- Change the magnification and font size.
- Add charts and graphs to an application, and embed third-party products that support OLE automation.

The following overrides are local only; they can be considered workstation preferences. The system stores these overrides on your workstation; therefore, they are accessible only from that workstation:

- Sizing of the parent and child windows.
- Changing the parent window.
- Changing fonts on a form.
- Maximizing the form.
- Turning on the exit bar.

See Also

JD Edwards EnterpriseOne Tools 8.96 Foundation Guide, "Using the JD Edwards EnterpriseOne Web Application User Interface"

Search Hierarchy

During the execution of an application, the system uses a search hierarchy to locate a user override. The system searches by user and group for each unique combination of application, form, version, and language in the following order:

Hierarchy	Description
User ID	When you access a specific application, the system searches first for an override for the application under your user ID.
Role	If the system does not find an override under your user ID for the application, it then searches for it at the role level. For example, if you are in the Accounts Payable role, the system searches for an override for that role.
*PUBLIC	If the system does not find an override for the application at the role level, it searches for it under *PUBLIC. If no override is found at the *PUBLIC level, the system uses the No Overrides default value.

Cached Override Information

The first time that a user opens an application form, the system reads the User Overrides Table (F98950) and creates a disk cache on the workstation. This table contains form-specific information such as menus, buttons, and formats. This cache improves network performance because multiple database fetches are not required to retrieve individual form elements.

However, if a system administrator or the user modifies user overrides with the User Overrides Revision application (P98950), the system writes the override information directly to the F98950 table, and not to the cached table. Because the system always reads overrides from the cached information, any modified user overrides cannot become effective until the user exits and reenters the system, causing the cached table to be refreshed.

For example, assume that you want to modify a journal entry by adding tabs and then associating those tabs with your user overrides. You would create the tab and then use the P98950 application to associate the tab with your user profile. You would not immediately see any records of the journal entry form because the user override is stored in the User Overrides Table, and the system is looking at the cached information. This process does not affect your ability to create and use local form changes, or workstation preferences, that are not stored in the User Overrides Table.

If for some reason you have two users who share the same user ID, be aware that the system does not share user override records. If both users sign on to the system at about the same time, the first of these two users to sign on will see the user overrides; the second user to sign on will not. Furthermore, the first user to make an override change during simultaneous sessions takes control of the F98950 record, and all other users signed on to the same user ID are locked out.

Setting Up User Overrides

This section provides an overview of how to create user overrides and discusses how to:

- Create user overrides.
- Change an individual user override to a group user override.
- Fix user overrides after a form change.

Understanding How to Create User Overrides

You can create user overrides for a user ID, a group, or for *PUBLIC. After you create a user override, your user override will be available on any workstation that you sign on to in the enterprise.

If a form for which you have created user overrides has changed after upgrading your software, the system attempts to merge your user overrides with the changed form. You might need to reset your user overrides; at the least, you will need to verify that your user overrides are still intact.

Creating User Overrides

You can create individual user overrides in which the changes that you make to an application reside on an enterprise server and are associated with your user ID. This override is available to you at any workstation that you sign on to. To create a group override, you first create an individual override, and then you change that override to a group override, thereby making the override available to employees within a group or to the entire company (*PUBLIC).

1. Access the application for which you want to create an override (for example, Standard Voucher Entry).
2. Modify the application (for example, rearrange columns or rows on the grid). When you exit the application, the preferences that you just set up are stored in User Overrides under your user ID.

Changing an Individual User Override to a Group User Override

Enter *P98950* in the Fast Path.

Note. The User Overrides application (P98950) is available on the Microsoft Windows client and the web client.

---

1. On the Work With User Overrides form, enter a user ID in the User/Role field, and then click Find to locate and select the individual override record that you wish to make available to a group.

   Note. Be sure that you select the correct type of override. The two types available are GF for grid tab format overrides, and HC for menu and toolbar overrides. For example, to deploy a grid tab to other users, you need to select a GF record; to deploy a changed menu bar or toolbar, you need to select an HC record.

2. On the Work With User Overrides form, scroll to the right to see the override type, and then click Copy.
3. On the Copy Overrides form, in the User/Role field, enter either a valid role, which has been set up in User Profiles, or *PUBLIC to copy the user override to a role or the entire company.

   If you want to change a JD Edwards EnterpriseOne demo version and want the copied version to use the same user overrides as the demo version, do not change the User/Role, but name the version to represent your custom version.
4. In the Version field, enter the version name to copy user overrides set up for one version to another version.

   Note. User Overrides does not create versions.

5. In the Language field, enter a valid language code to select the user override language for the specified user and application.
6. If you copied and modified the version, delete the individual user record that you copied.

   Deleting this record ensures that when you log on, you are viewing the overrides for the sign on role, not the override that is specific to your user ID.

   Note. The system creates a record for each form that you modify.

Fixing User Overrides after a Form Change

When you install a package on your workstation, you might have a discrepancy between the changed forms included with the new package and the grid, menu, or toolbar user overrides that you made before the installation. For example, a new column might have been added to a grid for which you have user overrides. After the package installation, the first time that you access the changed form, the system detects the discrepancy between the newly installed form and your existing user overrides. The system asks if you want to fix your user overrides to include the new column or delete your user overrides altogether. If the system cannot fix the discrepancy between the changed form and your user overrides, the system automatically deletes your user overrides.

The system performs this fix only for grid, menu, or toolbar user overrides because all other user overrides, such as an OLE attachment to a form, do not interfere with changes to a form.

The first time that you access a form after a package installation and have a discrepancy between the newly installed form and your user overrides, a message box appears. This box prompts you to either delete your user overrides for that form or have the system try to fix your user overrides to match the changed form.

On the message box that appears, perform one of these tasks:

- To delete your user overrides, click Delete.

  The system deletes your user overrides for that form. You can add your overrides again by following the process for creating user overrides.
- To try to fix your user overrides, click Fix.

The system tries to merge the changes from the newly installed form with your user overrides for that form. If successful, verify that the form works properly with your user overrides. If any errors exist with the grid formats or menu and toolbar customization after the system tries to fix the discrepancy, you should delete your user overrides for that form. On the Work With User Overrides form, select your overrides and then click Delete.

If the system is unable to merge the changes with your user overrides, the system automatically deletes your user overrides for that form. You can add your overrides again by following the process for creating user overrides.

CHAPTER 12

Working with Vocabulary Overrides

This chapter provides an overview of vocabulary overrides and discusses how to:

- Access Vocabulary Overrides from System Administration Tools.
- Access Vocabulary Overrides from JD Edwards EnterpriseOne Object Management Workbench (OMW).
- Create interactive vocabulary overrides.
- Create batch vocabulary overrides.
- Review vocabulary overrides.
- Reset a vocabulary override.
- Reset all vocabulary overrides in an application (interactive and batch).

Understanding Vocabulary Overrides

Vocabulary Overrides (P9220) is an application that you can use to change the text that appears on forms and reports. You can specify both form columns and row headings, provide customization for multiple languages and industries, and retain your overrides with the next software update.

Because the Vocabulary Overrides application (P9220) affects the user interface throughout JD Edwards EnterpriseOne, it is important that you secure this application from most users. When you work with vocabulary overrides for an interactive or batch application, the Vocabulary Overrides application simulates an application checkout from the central objects repository, just as if you checked out the application by using JD Edwards EnterpriseOne OMW. This checkout is done so that, while you are working on the application in the Vocabulary Overrides application, no one can check out the application.

Note. When the OMW line is written for the Vocabulary Overrides application, the system does not bring down specifications to the requesting workstation. Instead, the requesting workstation accesses the relational database tables directly.

After you make vocabulary override changes, use an update package to push these changes to the users.

Creating Vocabulary Overrides

You can create vocabulary overrides to customize the interactive and batch applications. After you make vocabulary override changes, use an update package to push these changes to the users. For example, you could create vocabulary overrides for the Verify OCM report. After you make vocabulary override changes, you should use an update package to push these changes to the users.

Note. When you create a vocabulary override for a report, the override occurs at the version level. When you run the version, the vocabulary override appears on the report instead of the data dictionary description. The vocabulary override does not affect the base report specifications or any other version of the report.

Forms Used to Work with Vocabulary Overrides

Form Name	FormID	Navigation	Usage
Work With Vocabulary Overrides	W9220H	Application Development (GH902), Object Management, Vocabulary Overrides (P9220)	Locate an interactive or batch application to which you want to apply vocabulary overrides. Review existing vocabulary overrides.
Object Management Workbench	W98220A	In Solution Explorer, enter *OMW* in the Fast Pathfield.	Locate an interactive or batch application to which you want to apply vocabulary overrides.
SAR Requirement	W559220A	On the Work With Vocabulary Overrides form, select an application and click Select.	This form only appears if the system administrator set up the processing option for vocabulary overrides to require a software action request (SAR) number for overrides.
Interactive Vocabulary Overrides	W9220A	On the Work With Vocabulary Overrides form, select the Interactive option and click Find. Select an application, and click Select.	Create and reset interactive vocabulary overrides.
Batch Vocabulary Overrides	W9220B	On the Work With Vocabulary Overrides form, select the Batch option and click Find. Select a batch version and click Select.	Create and reset batch vocabulary overrides.
Overridden Data Item Search	W9220E	On the Work With Vocabulary Overrides form, from the Form menu, select Data Item Search.	Review data items that contain vocabulary overrides.

Accessing Vocabulary Overrides from JD Edwards EnterpriseOne Solution Explorer

Access the Work With Vocabulary Overrides form.

1. Select the Interactive or Batch option and click Find.

   Use the query by example fields to refine the search.

2. Select an application and click Select.

If the application that you selected is checked out or is in the save location, the system displays the following error message: This object is currently in use by a project in JD Edwards EnterpriseOne OMW (either through check out or in the save location) and is, therefore, unavailable.

You must create vocabulary overrides for this application at another time or contact the users of the application to check in, erase their checkout, or delete the object from the save location.

3. If the SAR Requirement form appears, enter a SAR number in the SAR Number field.

   This form appears if the system administrator set up the processing option for vocabulary overrides to require a SAR number for overrides.

4. Click OK.

   The Interactive Vocabulary Overrides form or the Batch Vocabulary Overrides form appears. All of the interactive forms or batch versions associated with the application that you chose appear in the detail area. You can expand any row that has a plus (+) sign on the left side.

   The Vocabulary Overrides application essentially checks out this application in JD Edwards EnterpriseOne OMW so that while you are working on the application in Vocabulary Overrides, no one else can check out the application. After you finish creating overrides, Vocabulary Overrides erases the checkout in JD Edwards EnterpriseOne OMW.

Accessing Vocabulary Overrides from JD Edwards EnterpriseOne OMW

Access the Object Management Workbench program (P98220).

1. On Object Management Workbench, add the object to an OMW project.
2. Select the interactive or batch application, and then click the Design button.

   The Interactive Application Design form or the Batch Application Design form appears.

3. On the Design Tools tab, click Vocabulary Overrides.

   The system displays the following warning: Warning! You are now accessing Vocabulary Overrides. This application will override currently checked in objects. You must have authority to make changes.

4. If you have authorization to make vocabulary override changes, click OK.
5. If the SAR Requirement form appears, enter a SAR number in the SAR Number field.

   This form only appears if the system administrator sets the processing option that requires a SAR number for vocabulary overrides.

6. On the Work with Vocabulary Overrides form, click Select.

   The Interactive Vocabulary Overrides form or the Batch Vocabulary Overrides form appears. All of the interactive forms or batch versions associated with the application appear in the detail area. You can expand any row that has a plus (+) sign on the left side.

   The Vocabulary Overrides application essentially checks out this application in OMW so that while you are working on the application in Vocabulary Overrides, no one else can check out the application. After you finish creating overrides, Vocabulary Overrides erases the checkout in OMW.

Creating Vocabulary Overrides

Access the Interactive Vocabulary Overrides form.

1. To work with a language other than the domestic language, on Interactive Vocabulary Overrides, complete the Language field, and then click Find.
2. Enter a language code.

   Leave this field blank if you are creating vocabulary overrides in the domestic language.
3. Double-click the + button next to one of the forms listed in the detail area.

   The form expands, displaying the types of text that are available on that form, such as find/browse text, control text, grid column text, exit text, and text variables.
4. Double-click the + button for one of the types of text.

   The type of text expands, displaying all of the text that you can override.
5. To create a vocabulary override, change the text in the Description column for a particular item.

   Click OK when you finish creating overrides.

   Note. Some descriptions for data items contain carriage returns and new-line characters. To create a vocabulary override for these descriptions (indicated with an icon to the left of the row), select the data item row and, from the Row menu, select Extended Text Revision.
6. On the Extended Text Revision form, change the text in the field and click OK.

The Vocabulary Overrides application essentially checks out this application in OMW so that while you are working on the application in Vocabulary Overrides, no one can check the application out. After you finish creating overrides, Vocabulary Overrides erases the checkout in OMW.

To actually see the description change applied to the application, you must first retrieve the specifications for the application to the local client machine and run it. Do this by clicking either the Check Out or Get button in OMW.

Creating Batch Vocabulary Overrides

Access the Batch Vocabulary Overrides form.

1. To work with a language other than the domestic language, on Batch Vocabulary Overrides, complete the Language field, and then click Find.
2. Enter a language code.

   Leave this field blank if you are creating vocabulary overrides in the base (domestic) language.
3. Double-click the + button next to one of the versions listed in the detail area.

   The version expands, displaying the types of text that are available on that version, such as page header and group sections.
4. Double-click the + button next to one of the types of text.

   The type of text expands, displaying all of the text that you can override.
5. To create a vocabulary override, change the text in the Description column for a particular item.
6. Click OK when you finish creating overrides.

The Vocabulary Overrides application essentially checks out this application in OMW so that while you work on the application in Vocabulary Overrides, no one can check the application out. After you finish creating overrides, Vocabulary Overrides erases the checkout in OMW.

Reviewing Vocabulary Overrides

Access the Work With Vocabulary Overrides form.

You can use vocabulary overrides to review every location where someone has overridden a data item. You can view the override locations from a form or from a report.

1. On the Work With Vocabulary Overrides form, from the Form menu, select Data Item Search.
2. On the Overridden Data Item Search form, enter a data item to search for and then click OK.
3. Click one of these options to select a scope for the application search:
   - Interactive Application
   - Batch
   - Both
4. Select one of these options for the output results:
   - Interactive

     If you view the search results by using the interactive application, the Data Item Locator form appears when this search is complete. This form displays a list of all of the applications in which the data item appears.
   - Printed Report

     If you view the search results by using the printed report, an Adobe Acrobat Portable Document Format (.pdf) file is created, which you can view or print.
5. From the Form menu, select Run Report.

Resetting a Vocabulary Override

Access the Interactive Vocabulary Overrides form or the Batch Vocabulary Overrides form, depending on the type of application in which you want to reset a vocabulary override.

You can reset vocabulary overrides to the original data dictionary definition. If you need to reset multiple vocabulary overrides to the default data dictionary definition, JD Edwards EnterpriseOne provides an automated process that resets overrides at the interactive form level, the batch version level, and the interactive and batch application level. When you reset vocabulary overrides at the form level, you reset all vocabulary overrides on a specific form—for example, the Work with Addresses form (W01012B) in the Address Book application. When you reset vocabulary overrides at the application level, you reset all vocabulary overrides on all forms or versions in an entire interactive or batch application—for example, the Address Book application (P0101) or the Print Mailing Labels report (R01401).

1. Double-click the + button in the row header for one of the forms or versions in the detail area, and then double-click the + button in the row header for a type of text on the form or a type of section in the version.

   The detail area expands to display the data items associated with the type of text or section.
2. Select the data item that you want to reset, and then, from the Row menu, select Reset Description.

   Note. The Reset Description menu option is inactive if a vocabulary override does not exist for the data item.

   ---
3. Click OK to return to the Work With Vocabulary Overrides form.

   If you click Cancel to return to the Work With Vocabulary Overrides form *after* you reset a vocabulary override, you *do not* cancel the action. The data item remains at the default data dictionary definition.

Resetting All Vocabulary Overrides on a Form (Interactive and Batch)

Access the Work With Vocabulary Overrides form.

1. Click one of these options and then click Find:
   - Interactive
   - Batch
2. Select an application and click Select.

   Depending on the type of application, either the Interactive Vocabulary Overrides form or the Batch Vocabulary Overrides form appears. The detail area displays forms for interactive applications and versions for batch applications.
3. From the Form menu, select Reset by Application for interactive applications or Reset by Batch for batch applications.

   The software clears all vocabulary overrides from the *entire* application and resets the data items to the base definitions. If no base definition exists for a data item, the software resets the data item to the default data dictionary definition.

Important! When you select either the Reset by Application or the Reset by Batch menu option, the decision is final; the software does not provide a confirmation box or a proof mode.

CHAPTER 13

Using the Scheduler Application

This chapter provides an overview of the Scheduler application (P91300) and discusses how to:

- Work with the Job Scheduler.
- Use advanced scheduling options.
- Review the job schedule.
- Work with the Scheduler server.
- Work with daylight savings rules.
- Run Scheduler reports.

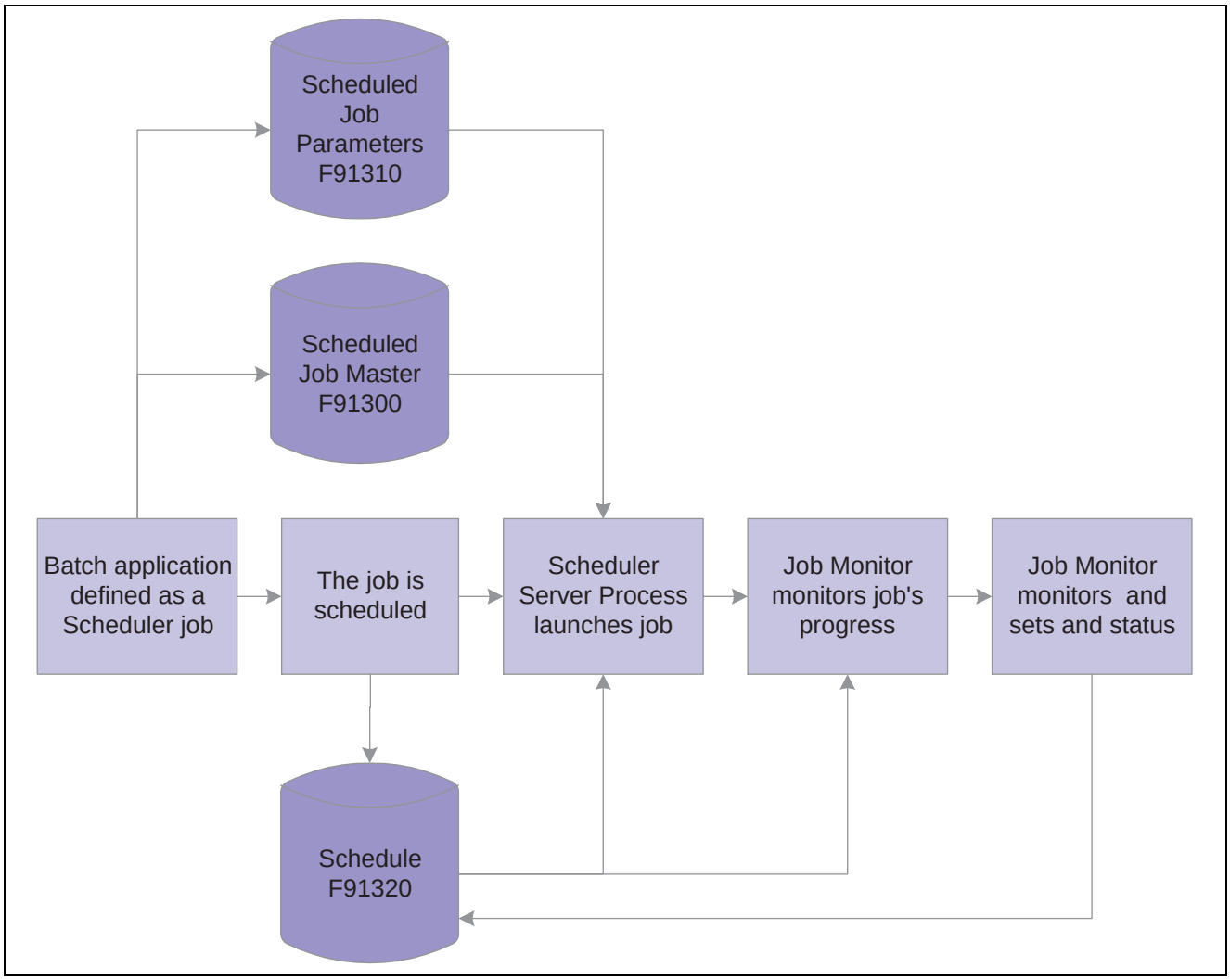
Understanding the Scheduler Application

Occasionally, you might want to run batch jobs that take up a great deal of machine resources or that require users to be signed off after normal working hours. You also might want the flexibility to run jobs at scheduled intervals during the day or even periodically throughout the month or year.

The Scheduler application enables you to schedule batch jobs to run after hours or periodically throughout the day, according to a schedule that you define. You can schedule jobs by time (daily, weekly, monthly, yearly) or based on a specified period. You also can set up the scheduler to restart a job in the event of a job failure.

You can specify the server on which you want the job to run, as well as the time zone, regardless of the locale. The Scheduler system uses a modified version of Universal Coordinated Time (UCT), which counts the number of minutes, not seconds.

The following graphic illustrates the scheduling process:



The scheduling process

When you define a scheduled job, the parameters of that job are stored in the Scheduled Job Master table (F91300).

After the job is scheduled, the system writes records to the Schedule table (F91320), indicating each time that the job should be launched. As the job runs, the Job Monitor monitors the progress of the job.

When the job ends, the Job Monitor assigns an end status to the job and updates the record of the job in the Job Schedule table to indicate that the job either ended successfully or in error.

Working with the Job Scheduler

This section provides an overview of the Job Scheduler in Oracle's JD Edwards EnterpriseOne and discusses how to:

- Set processing options for the Job Scheduler.
- Schedule a job.
- Schedule a recurring job.

- Revise a scheduled job.
- Review all jobs or local jobs.

Understanding the Job Scheduler

When you schedule a batch process to run through the Scheduler, you can also add a recurrence pattern to the job, which means that the job will restart at the intervals that you define, such as once a week, once a month, or once a year. You can also specify how many times you want the job to run before it ends, or you can define a date after which the job will no longer run.

You schedule jobs in the local time of the server on which the job will run. For example, when you schedule a job, you might select the version that specifies the eastern time zone to run jobs in eastern standard time (EST).

Prerequisite

To use a server's time zone, you must first specify the time zones that you want to use. To do this, copy the Scheduler processing options (version ZJDE0001 on the Work with Versions form), and modify them according to your needs. If you use more than one time zone, you should modify the processing options to display the Work with Versions form each time that you invoke the Schedule Jobs application. That way, you can select the correct version for the time zone in which you want to schedule the job.

See Also

Chapter 13, "Using the Scheduler Application," Scheduling a Recurring Job, page 81

Chapter 13, "Using the Scheduler Application," Setting Processing Options for the Job Scheduler, page 80

Forms Used to Work with the Job Scheduler

Form Name	FormID	Navigation	Usage
Work With Versions	W983050B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300)	Locate the version that specifies the time zone in which the scheduled job will run.
Work With Scheduled Jobs	W91300B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300) On the Work With Versions form, click the version and then click Add.	Access forms to schedule a job.
Scheduling Information Add	W91300A	On the Work With Scheduled Jobs form, click Add.	Schedule a job.
Scheduling Advanced Options	W91300I	On the Scheduling Information Add form, select Advanced Options from the Form menu.	Enter the user and machine information for the scheduled job.
Recurring Scheduling Information Revisions	W91300C	On the Scheduling Information Add form, select Recurrence from the Form menu.	Schedule a recurring a job.

Setting Processing Options for the Job Scheduler

Processing options enable you to specify the default processing for programs and reports.

Display

Although processing options are set up during a JD Edwards EnterpriseOne implementation, you can change processing options each time you run a program.

View Local Time

Local Time Zone

Enter ’1’ to adjust time for daylight savings. Enter ’0’ to never adjust for daylight savings.

Use this Daylight Savings Rule when adjusting for daylight savings.

Process

Although processing options are set up during JD Edwards EnterpriseOne implementation, you can change processing options each time you run a program.

Maximum number of job schedule records

Defaults

Although processing options are set up during JD Edwards EnterpriseOne implementation, you can change processing options each time you run a program.

Default Job Type

Number Of Job Occurrences

Max Number of Job re-submissions

Scheduling a Job

Access the Work With Scheduled Jobs form. On the Work with Versions form, double-click the version that specifies the time zone in which the scheduled job will run.

Note. If you use only one time zone, you might not be prompted to select a version. In this case, the Work With Versions form does not appear. You can delete the Work With Versions form in the menu properties for P91300. By default, GH9015/P91300 prompts for the version.

---

1. On the Work With Scheduled Jobs form, click Add.
2. On the Scheduling Information Add form, in the Scheduled Job Name field, enter a name that uniquely identifies to the system and the user of a scheduled job.

   Use this name to indicate the job function, such as *Monthly Close* or *Nightly Back Up.*
3. In the Scheduled Job Status, determine the status of the scheduled job.

   As long as the status is active, the Scheduler determines if the job should be submitted to the server for execution. When the scheduled end date for the job has been reached, the status changes to *Not Active.* To stop the Scheduler from considering the job for submission, you can change the status to *Not Active* (or suspended) at any time prior to the end date. You can reactivate the job if you want the Scheduler to include the job again, but you can reactivate a job only if the end date is in the future.
4. In the Scheduled Batch Application, specify the object name of the report that the Scheduler submits to the server.
5. In the Scheduled Version, specify the version of the report that is scheduled to run. A version identifies a specific set of data selections and sequencing settings that the batch job uses.
6. In the Scheduled Start Date/Time, determine the next date on which the Scheduler submits the scheduled job to the server for execution, and then click OK.

Scheduling a Recurring Job

Access the Recurring Scheduling Information Revisions form.

1. Select one of these options, and complete the accompanying fields that appear after you select an option:

- By Time

  Run the job every *n* days or every weekday.

  Run the job at the specified time interval. For example, run the job every 40 minutes or every eight hours.

- Daily

  Run the job at the specified interval of days or every weekday. For example, run the job every seven days or every weekday.

- Weekly

  Recur every *n* weeks on Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, or Saturday.

  Run the job at the specified weekly interval on the specified day of the week. For example, run the job every two weeks on Monday.

- Monthly

  Day *n* of every *x* months or the [first, last, and so on] [day, Sunday, Friday, and so on] of every *n* months.

  Run the job on the specified day of every *n* months or on a specific day of every *n* months. For example, run the job on day one of every four months or on the last Friday of every second month.

- Period

  Day *n* of every *x* periods or the [first, last, and so on] [day, Sunday, Friday, and so on] of every *n* periods.

- Company

- Yearly

  Every [month name] [date] OR

  The [first, last, and so on] [day, Sunday, Friday, and so on] of [month name] OR

  Day *x* of the [first, last, and so on] period OR

  The [first, last, and so on] [day, Sunday, Friday, and so on] of the [first, last, and so on] period.

  Schedule the job at the specified time of the year. For example, you might want to run the job on the last day of December of each year, every January 1, on the first Monday of June, on day 15 of the last period, or on the second Tuesday of the fourth period.

2. Specify when you want the Scheduler to stop submitting the job by selecting one of these options:
   - No end date
   - End after
   - End by

     Enter the month, day, and year on which you want the job to expire.
3. Click OK.

Revising a Scheduled Job

Access the Work With Versions form.

You might want to revise the information for a job. For example, you might want to change the job status to *Active* or *Not Active,* enter a new batch process as the scheduled job, or change the job start date and time.

1. On the Work With Versions form, select the version that specifies the time zone in which the job will run and click Select.

2. On the Work with Scheduled Jobs form, select the job that you want to revise, and then select Job Revisions from the Row menu.
3. On the Scheduling Information Revisions form, modify the following fields as necessary and click OK:
   - Scheduled Job Status
   - Scheduled Batch Applications
   - Scheduled Version
   - Scheduled Start Date/Time
4. If you want to remove recurrence from a scheduled job, select Remove Recurrence from the Form menu.
5. If you want to add category codes to the scheduled job, select Category Codes from the Form menu.
6. On the Scheduler Category Codes form, if you want to revise the advanced options for this job, select Advanced Options from the Form menu.

   The Scheduling Advanced Options form appears.

Note. You can also activate or inactivate a job by selecting the job on the Work With Scheduled Jobs form, and then selecting Change Status from the Row menu.

Reviewing All Jobs or Local Jobs

Access the Work With Scheduled Jobs form.

If necessary, you can review all of the jobs in all time zones, or local jobs only. Depending on the view that you are currently using, the system protects the other choice. For example, if you are currently viewing local jobs, the system protects the Local Jobs choice and enables you to select only All Jobs.

1. On the Work With Scheduled Jobs form, select Display from the Form menu.
2. Click either All Jobs or Local Jobs.

Using Advanced Scheduling Options

This section provides an overview of the advanced scheduling options and discusses how to:

- Override the environment.
- Override job expiration specifications.
- Define when the Scheduler resubmits a job.
- Override batch application specifications.
- Add values to a report interconnect.

Understanding Advanced Scheduling Options

You can use advanced options to override the job properties, such as the location where the job will run and the environment in which it will run. You can also use advanced options to specify whether you want the system to resubmit a job if it ends in error or if you want to change job expiration specifications.

You can define whether to log errors to the jde.log or jdedebug.log, and whether you want the system to override printer locations and job queues. You can also add a report interconnect to a job if you want to pass parameters to it.

Note. To restore the default values in the advanced options application, click the default button on the Scheduling Advanced Options forms.

Forms Used to Work with Advanced Scheduling Options

Form Name	FormID	Navigation	Usage
Work With Scheduled Jobs	W91300B	Report Management (GH9111), Job Scheduler (GH9015), Schedule Jobs (P91300)System On the Work With Versions form, click the version and then click Add.	Access forms to schedule a job.
Scheduling Advanced Options	W91300I	On the Work With Scheduled Jobs form, select the job, and then select Advanced Options from the Row menu.	Override the environment, job expiration, and batch application specifications. Define when the Scheduler resubmits a job.

Overriding the Environment

Access the Scheduling Advanced Options form. Select the Launch Overrides tab.

A situation might exist in which you need to override an environment. For example, you would override the environment if the environment in which you want to run the job is not available or is different from the environment that you were logged into when you scheduled the job.

Overriding Job Expiration Specifications

Access the Scheduling Advanced Options form. Select the Job Expirations tab.

Job expiration specifications ensure that servers do not become overloaded with unexpired jobs. If necessary, you can override job expiration specifications so that the job never expires, or expires after a certain number of minutes.

For example, suppose you schedule a job to run at midnight and another job for 1:00 a.m., but the server goes down and probably will not come back up again before the jobs are scheduled to run. In this case, you can specify that the first job, which you scheduled to run at midnight, expires in 30 minutes (12:30 a.m.), so that if the server does not come back up within 30 minutes, the job expires.

Defining When the Scheduler Resubmits a Job

Access the Scheduling Advanced Options form.

You can define when the Scheduler resubmits jobs. This feature is useful if a job ends in error, for example, because the Scheduler will submit the job after a certain period of time.

To avoid the use of system resources, you can limit the number of times that a job can be resubmitted. You also can have the Scheduler check for connection errors or runtime errors when the job runs. Connection errors occur when the system fails to connect to the server to submit the job. Runtime errors occur when the server on which the job is running places the job in an error state. You can set up the system to monitor for both cases.

To define when the system resubmits jobs:

1. On the Scheduling Advanced Options form, select the Job Resubmission tab.
2. Specify the number of minutes that elapse before the job continues or terminates, and then activate one of the following options:
   - Let the job continue.
   - Terminate the job.
   - Terminate the job and resubmit.
3. If you want to terminate the job and resubmit it after a certain period of time, select the terminate the job and resubmit after option, and enter the number of minutes that you want to elapse before the system resubmits the job.
4. Select one of the following options that apply when the job ends in error:
   - Do nothing.
   - Resubmit immediately.
   - Resubmit after.
5. Specify whether you want the system to check for connection errors, runtime errors, or both.
6. In the Max Number of Job Resubmissions field, specify the maximum number of times that you want the job to be resubmitted.
7. Click OK.

Overriding Batch Application Specifications

Access the Scheduling Advanced Options form.

You can specify if you want errors written to the jde.log. If you want errors written to the jdedebug.log, you can set the trace level to determine what types of errors to include in the log.

You can also override the printer at which a report is printed. This feature is useful if a specific printer is down or if you want to print a report to a printer other than the default. You can specify whether a job should be printed immediately upon completion or held in the job queue to be printed later.

Also, you can override the queue to which the output of a submitted job is sent. If you want to pass parameters to a particular batch job, you can attach a report interconnect through Batch Application Specifications.

To override batch application specifications:

1. On the Scheduling Advanced Options form, select the Batch Application Overrides tab.
2. Select one or more of the following options:
   - Jde.log
   - Jdedebug.log

     If you select jdedebug.log, you must also select the jde.log option.
3. Complete these fields:

- UBE Logging Level

  If you select jdedebug.log, you can set a trace level to log certain levels of errors.

- Printer Name

  Enter the name of the printer to which you want to print the report that the job generates. If you want to use a default printer, specify *DEFAULT*.

- Print Immediate

  If you want the job output to be sent to the printer immediately, select this option.

- Job Queue

  Enter the name of the job queue to which you want the job output sent.

4. Click OK.

Adding Values to a Report Interconnect

Access Work With Scheduled Jobs form.

You can add values to be passed through a report interconnect into a batch process when that batch process is launched. The batch process must first contain a report interconnect.

1. On the Work With Scheduled Jobs form, select the job and then select Advanced Options from the Row menu.
2. On the Scheduling Advanced Options form, select Parameters from the Form menu.
3. On the Report Interconnect form, enter the values that you want to pass to the batch process when the process runs.
4. Click OK.

Reviewing the Job Schedule

This section provides an overview of how to review scheduled job information and discusses how to:

- Review all job schedules.
- Change the launch status of a job.
- View job details.
- Set the job status manually.
- Reset the job schedule.

Understanding Job Schedule Reviewing

When you schedule a job that includes a recurrence pattern, the system creates a set of schedule records, or instances, for the job in the F91320 table. The F91320 table indicates the times and dates that the job will run. You can review these instances and their statuses, and also change the scheduled job information. For example, you can change the location at which you want a job to process, delete a job instance, or override any advanced functions.

Note. Because the F91320 table is also used for audit information, you can modify or delete only jobs that have not yet run.

See Also

Chapter 13, "Using the Scheduler Application," Revising a Scheduled Job, page 82

Forms Used to Review the Job Schedule

Form Name	FormID	Navigation	Usage
Work With Versions	W983050B	Job Scheduler (GH9015), Schedule Jobs	Locate a version that you want to review.
Work With Scheduled Jobs	W91300B	On the Work With Versions form, select a version and then click Select.	Review all scheduled jobs for a version.
Work With All Job Schedule	W91300M	On the Work With Scheduled Jobs form, select All Schedules from the Form menu.	Review all scheduled jobs for a version.
Job Schedule	W91300F	On the Work With Scheduled Jobs form, select the job, and then select Job Schedule from the Row menu.	Change the launch status of a job instance. Reset the job schedule.
Job Maintenance	W986110BC	On the Job Schedule form, select the job, and then select View Detail from the Row menu.	View details about a job, as well as the job queue, the priority in which the job will run, and the location of the report printer. From this form, you can also change the job priority or the location where the report will print.
Manually Set Job Status	W91300L	On the Job Schedule form, select the job instance for which you want to manually set the job status, and then select Set Status from the Row menu.	Set the job status manually.

Reviewing All Job Schedules

Access the Work With All Job Schedule form.

When you review all of the job schedules, you can view all instances of jobs that have been launched. You can even revise a job by choosing a job instance and then selecting Revise Job from the Row menu.

You can filter the job instances that you want to review by launch date, start date, and time. For example, you can review all of the job instances for today's date by entering that date in the Scheduled Start Date field. Or you can review all job instances that were launched on a certain date by entering that date in the Job Launch Date field. You can also filter job instances by scheduled job name, launch status, report name, or scheduled version.

1. On the Work With All Job Schedule form, filter by start date or start time by completing these fields:
   - Scheduled Start Date
   - Scheduled Start Time
2. To filter by launch date or launch time, complete these fields:
   - Job Launch Date
   - Job Launch Time
3. To filter by job name, launch status, report name, or scheduled version, complete these fields:
   - Scheduled Job Name
   - Launch Status
   - Report Name
   - Scheduled Version
4. To view all scheduled jobs in all time zones, select All Time Zones from the Form menu.
5. To view all scheduled jobs in the local time zone, select Local Time Zone from the Form menu.

Changing the Launch Status of a Job

Access the Work With Version form.

You can change the launch status of a job. For example, you might need to put a job on hold or reschedule a job.

1. On the Work With Versions form, select the time zone in which the job will run, and then click Select.
2. On the Work With Scheduled Jobs form, locate the job that you want to change.
3. From the Row menu, select Job Schedule.
4. On the Job Schedule form, select the job instance for which you want to change the status, and then enter a new status in the Launch Status field.

   Enter *1* for a scheduled status or *50* for hold.
5. Click OK.

Viewing Job Details

Access the Job Schedule form.

1. On the Job Schedule form, select the job, and then select View Detail from the Row menu.
2. On the Job Maintenance form, complete the following fields, if necessary, and click OK:
   - Job Priority
   - Printer Name

Setting the Job Status Manually

Access the Work With Scheduled Jobs form.

As a system administrator, you can change the status of jobs if the Scheduler is not updating the launch status or if the Job Monitor is disabled.

Note. If you need to stop a job, select Work with Servers from the Form menu on the Work with Scheduled Jobs form.

Important! You should secure other users from accessing the Set Status option. Only the JD Edwards EnterpriseOne administrator should have access to this option.

1. On the Work With Scheduled Jobs form, select the job, and then select Job Schedule from the Row menu.
2. On the Job Schedule form, select the job instance for which you want to manually set the job status, and then select Set Status from the Row menu.
3. On the Manually Set Job Status form, complete the Scheduled Launch Status field.

Resetting the Job Schedule

Access the Job Schedule form.

If you configure a job schedule and then change your mind, you can remove the changes and regenerate the job schedule by using the previously defined recurrence pattern. The job schedule is reset to the way it was before you made the changes.

1. On the Job Schedule form, select Reset Schedule from the Form menu.

   The system displays the following warning message: This will remove any custom changes to this job's schedule and regenerate the schedule using the recurrence pattern. Are you sure you want to continue?
2. Click Yes to confirm resetting the job's schedule.

Working with the Scheduler Server

This section provides an overview of the Scheduler server and discusses how to:

- Stop or restart the Scheduler server.
- Pause the job launcher or job monitor.
- Reset the Scheduler server.
- Refresh the Scheduler server settings.
- Modify the Scheduler server and monitor sleep time.

Understanding the Scheduler Server

The Scheduler server is a process that performs two distinct functions: it launches all jobs at the scheduled times, and it monitors each job's progress and ending state. These functions are started by a JDENET message, as defined in the following kernel type in the jde.ini file:

```
[JDENET_KERNEL_DEF10]
dispatchDLLName=jdekrnl.dll
dispatchDLLFunction=_JDEK_DispatchScheduler@24
maxNumberOfProcesses=1
beginningMsgTypeRange=2001
```



```
endingMsgTypeRange=2256
newProcessThresholdRequests=0
numberOfAutoStartProcesses=1
```

The Scheduler launches batch processes in a server, environment, and user combination, based on the information in the F91300 table. After the Scheduler is started, JDENET keeps it in a wait state by calling the Scheduler dispatch function every minute with an idle message. This idle message enables the Scheduler process to check whether it should launch a job or monitor the jobs that are running. In addition, JDENET sends the Scheduler any message sent from the workstation (for example, messages that new job schedules were added).

You can stop, reset, restart, and refresh the Scheduler server. For example, if the server goes offline, it needs to be reset. You can also modify the server and monitor sleep time, specifying how many seconds you want JDENET to wait until it checks to see if it needs to initialize, or wake up, the Scheduler server.

You also might encounter situations for which you need to activate or deactivate the Job Launcher or Job Monitor. For example, you might need to take down the servers to which you submit jobs and for which you want to avoid unnecessary connection errors when jobs are submitted.

You can also change the jde.ini file to enable the Scheduler to restart automatically by changing the`numberOfAutoStartProcesses` line. If you enable this feature, and the server on which the Scheduler server is running comes down, the Scheduler server automatically restarts when the server comes back up, instead of having to be restarted manually. When the Scheduler server restarts, the Scheduler checks the F91320 table to determine if it should restart on that server. If not, the Scheduler shuts down.

JDENET handles the calls to initialize the Scheduler server. The JDENET process either sends a message to initialize the Scheduler to launch a job if it receives a message that table F91320 has changed, or it sends an idle message if no change is detected. For faster response time, you can decrease the number of seconds that you want JDENET to wait until it checks to see if table F91320 has changed.

Note. This application is for administrators only. You should secure users from accessing the Scheduler server application.

Control Record

A control record is a job record in the F91300 table. It is named *SCHEDULE and is hidden from the user. The *SCHEDULE record contains information about the state of the Scheduler processes on the server, and it is the method of communicating to those processes.

For example, when the launch loop starts on the server, it sets a flag in this record to indicate that it is up and running. You can end the launch loop by toggling the corresponding end process flag (such as Job Launcher Status or Job Monitor status) from the Form menu on Scheduler Server Control. The next time that the launch loop fetches the control record, it finds the flag, resets both flags, and ends.

If the system does not find the control record when it is fetched, the record is recreated by P91300 when entering the Scheduler Server Control form. In addition, if the record is corrupt, the aforementioned function is called to recreate it. The sleep times for the job monitor are reset to 15 minutes, and the audit information in this record is updated with the user ID set to SCHEDULER.

Dispatch Function

The dispatch function handles the incoming message from the workstation and starts the requested process. The JDENET process either sends a message to initialize the Scheduler, signals that the F91320 table has changed, or gives an idle message. The idle message is sent every minute unless one of the other messages is sent. When the idle message is sent, the dispatch function checks to see if the launch loop or job monitor needs to be called. If neither does, control is given back to JDENET.

Launch Loop

The launch loop function selects all of the jobs up to the current time. It then loops through the selected records and launches the active jobs if they have not expired. After launching all current jobs, the launch loop fetches all future jobs sorted by start time. If the fetch succeeds, the next select time (NST) is set to the difference between the current time and the start of the next job. If the fetch fails, the NST is set to zero, which indicates that this function should be run the next time that any record is added to or updated by the F91320 table. In addition to launching jobs, the launch loop checks the control record periodically to see if it should exit.

The launch loop also looks for updates of all the schedule instances (F91320 records) and job headers (F91300 records) that it fetches. After the launch loop has processed these records, it then commits any changes and unlocks all of the records.

Job Monitor Loop

The job monitor loop monitors the ending statuses of the launched jobs and relaunches those that end in error if requested to do so by the user. This loop cycles through the internal job list that the job launch loop populates. In addition, it terminates jobs that run too long, if requested to do so. A job cannot be relaunched for more times than specified in the F91300 record of the job.

Like the launch loop, the job monitor loop periodically fetches the control record to see if it should end.

See Also

JD Edwards EnterpriseOne Tools 8.96 Development Tools: Report Design Aid Guide, "Working with Report Interconnects," Defining Report Interconnects

JD Edwards EnterpriseOne Tools 8.96 Development Tools: APIs and Business Functions Guide, "Debugging Business Functions," Using Debug Tracing

Forms Used to Work with the Scheduler Server

Form Name	FormID	Navigation	Usage
Work with Versions	W983050B	Job Scheduler (GH9015), Schedule Jobs (P91300)	Locate the version that specifies the time zone in which the scheduled jobs run.
Work with Scheduled Jobs	W91300B	On the Work with Versions form, select a version and then click Select.	Access the Scheduler Server Control form.
Scheduler Server Control	W91300G	On the Work with Scheduled Jobs form, select Scheduler Server from the Form menu.	Stop, restart, reset, refresh, or modify the Scheduler server. Pause the job launcher and job monitor. Modify the Job Monitor sleep time.

Stopping or Restarting the Scheduler Server

Access the Work with Versions form.

1. On the Work with Versions form, select the version that specifies the time zone in which the scheduled jobs run, and then click Select.
2. On the Work with Scheduled Jobs form, select Scheduler Server from the Form menu.
3. On the Scheduler Server Control form, perform one of these operations and click OK:
   - To stop the server, select Stop Scheduler from the Form menu.
   - To restart the server, select Start Scheduler from the Form menu.

Pausing the Job Launcher or Job Monitor

Access the Scheduler Server Control form.

Note. You might want to pause the job launcher or job monitor, such as when you want to take down the servers to which you submit jobs, and you want to avoid server connection errors that might occur while those servers are down. When you pause the job launcher, the Scheduler stops looking at the F91320 table for jobs to launch. When you pause the job monitor, the Scheduler stops monitoring the status of launched jobs.

---

1. On the Scheduler Server Control form, select Pause Job Launcher from the Form menu to pause the job launcher.
2. To pause the job monitor, select Pause Job Monitor from the Form menu.

Resetting the Scheduler Server

Access the Scheduler Server Control form.

Note. You reset the Scheduler server after you change the status of the Job Monitor or Job Launcher. For example, if you change the status of the Job Monitor, you would select Reset to refresh the settings on the server.

---

1. On the Scheduler Server Control form, select Reset from the Form menu.
2. Click OK.

Refreshing the Scheduler Server Settings

Access the Scheduler Server Control form.

Note. When you refresh the Scheduler server settings, the server refreshes its cache of launched jobs, and closes and restarts all environment and table handles. It is a refresh of the server's internal structures. You might want to refresh the Scheduler server settings if you had to restart the server.

1. On the Scheduler Server Control form, select Refresh from the Form menu.
2. Click OK.

Modifying the Scheduler Server and Monitor Sleep Time

Access the Scheduler Server Control form.

Note. Sleep time is the time that the Scheduler server or monitor is idle.

Scheduler Sleep Time This field indicates the number of seconds that the scheduler server will sleep (or idle). For example, if this field is set to 60 seconds, the Scheduler server checks every 60 seconds to see if it needs to launch or monitor jobs. The default is 60, and it must be greater than zero.

Job Monitor Sleep Time This field indicates the number of minutes the job monitor will pause between job status checks.

Working with Daylight Savings Rules

This section provides an overview of working with daylight savings rules and discusses how to:

- Add daylight savings rules.
- Revise daylight savings rules.

Understanding How to Work with Daylight Savings Rules

Daylight savings rules tell the system how each locale implements its daylight savings time. The Scheduler uses these rules, along with time zone information, to determine when jobs should run on a particular server.

You can add a new daylight savings rule or modify an existing one.

Forms Used to Work with Daylight Savings Rules

Form Name	FormID	Navigation	Usage
Work With Daylight Savings Rules	W00085A	Job Scheduler (GH9015), Daylight Savings Rules (P00085)	Access forms to add a new daylight savings rule or revise an existing rule.
Add Daylight Savings Rule	W00085B	On the Work With Daylight Savings Rules form, click Add.	Add a daylight savings rule.
Daylight Savings Rule Revisions	W00085B	On the Work With Daylight Savings Rules form, select an existing rule and click Select.	Revise a daylight savings rule.

Adding Daylight Savings Rules

Access the Work With Daylight Savings Rules form.

1. Click Add.
2. On the Add Daylight Savings Rule form, complete these fields:
   - Rule Name
   - Description
3. In the Rule Type area, select the method that you want to use to determine a daylight savings rule:
   - By Day of Week Instance
   - By Day of Month
4. Depending on the method that you chose, complete the remaining fields to specify when daylight savings starts and ends, and then click OK.

Revising Daylight Savings Rules

Access the Daylight Savings Rule Revisions form and revise the appropriate fields.

Rule Name Unique name identifying a daylight savings rule. Use daylight savings rules to adjust time for a geographic and political locale.

By Day of Week Instance A code that indicates the method that is used to determine a daylight savings rule.

By Day of Week Instance indicates that daylight savings starts and stops on a certain day of the week for a certain month, such as the first Sunday of April to the first Sunday of October. By Day of the Month indicates that daylight savings starts and stops on a certain day of a certain month, such as April 3 to October 10.

By Day of the Month A code that indicates the method that is used to determine a daylight savings rule.

By Day of Week Instance indicates that daylight savings starts and stops on a certain day of the week for a certain month, such as the first Sunday of April to the first Sunday of October. By Day of the Month indicates that

daylight savings starts and stops on a certain day of a certain month, such as April 3 to October 10.

Running Scheduler Reports

Run the Scheduled Jobs report when you want to review a summary of scheduled jobs and their statuses. You can use processing options to specify whether to run this report based on UCT or local time. You also can adjust for daylight savings time.

If you want to purge records from the F91320 table, run the Scheduler Purge program. You can run the purge program in proof mode or final mode.

See Also

JD Edwards EnterpriseOne Tools 8.96 Development Tools: Report Printing Administration Technologies Guide, "Working with Report Printing Administration," Batch Versions at Submission

Printing the Scheduled Jobs or Purge Scheduled Jobs Report

Access the Work With Batch Versions - Available Versions form.

1. Select a version in the detail area, and then click Select.
2. On the Version Prompting form, select one or both of the following options, and then click Submit:
   - Data Selection
   - Data Sequencing
3. On the Report Output Destination form, select one of these options, and then click OK:
   - On Screen
   - To Printer

CHAPTER 14

Understanding Media Objects and Imaging

This chapter discusses:

- Media objects.
- Imaging.
- Media object queues.
- Media object tables.
- Language considerations for media objects.

Media Objects

Media objects and imaging features in Oracle's JD Edwards EnterpriseOne enable you to attach useful information to an application, including information that might currently exist as a paper-based document. The media objects feature enables you to attach the information to applications, forms and rows, and Object Librarian objects. The imaging feature within media objects gives you flexibility to create a more efficient method of information storage.

This table describes the types of information that you can attach to a grid row or a form:

Text	Media objects provide a word processor that lets you create a text-only attachment. For example, you can use a text attachment to provide specific instructions for a form or additional information about a record.
Image	Images include files such as Windows bitmaps, Graphics Interchange Format (GIF) files, and JPEG files. These files might represent electronically created files, as well as scanned images of paper-based documents.
OLE	Media objects can be files that conform to the OLE standard. OLE enables you to create links between different programs. By using these links, you can create and edit an object from one program in a different program. JD Edwards EnterpriseOne provides the links that you need to attach OLE objects. You attach OLE media objects at the base form level. Media objects attached at this level are attached to a form and not to any data that might appear in the form. You can attach media objects to a detail area or a form, but the files themselves exist in separate directories. The only file information that is included with the application to which the OLE object links is the path to the supporting file. You can only use OLE objects that you properly register and install as OLE objects through JD Edwards EnterpriseOne.

Shortcuts	A shortcut is a link that opens JD Edwards EnterpriseOne application. Within media objects, you can only attach JD Edwards EnterpriseOne shortcuts; that is, you cannot attach Windows shortcuts to media objects.
Uniform Resource Locations (URL) and files	Media objects can be links to web page URLs or other related files. When a developer attaches a URL media object to a control object on a form, the web page appears as part of the form. When a user attaches a URL to a form or Object Librarian object, the media object acts as a link to the URL.

System administrators can also set up templates. A template might include attachments of its own, such as images and shortcuts. For example, you can create a letterhead and a standard form for a memo. You might create a shortcut in the template to provide access to an application that uses data specific to the information that you add to the template.

JD Edwards EnterpriseOne Text Items

Text items are items that you create using the JD Edwards EnterpriseOne media objects word processor. They do not require media object queues. The F00165 table contains both the associated key value of the data record to which the text media object is attached, and the text itself. Text items that originate from applications external to JD Edwards EnterpriseOne (for example, Microsoft Word or WordPad) must be stored as OLE objects.

Imaging

The imaging capabilities available in JD Edwards EnterpriseOne enable you to link to a third-party imaging product. Imaging systems enable you to scan and electronically store paper-based information. For example, this information might include documents such as sales orders, purchase orders, vendor invoices, and product schematics. JD Edwards EnterpriseOne imaging integration includes a media objects viewer and a third-party product that provides scanning and searching interfaces to enable you to find and display images. Implementation of imaging also provides a view of integrated images by using the viewer of the native imaging product.

When you use a third-party vendor, the F00165 table stores the reference to image attachments, but the third-party software controls the search and retrieval of images.

Media Object Queues

JD Edwards EnterpriseOne media object queues enable the storage location of media objects to be tracked by reference rather than physical network location, which simplifies the administration of media location. For example, the location for media objects on the server can change, and the change is reflected in only one place in JD Edwards EnterpriseOne.

You must define a media object queue to identify the pointer to the location where the actual image files or OLE objects reside. Media object queues provide the system administrator with the ability to easily manage the storage of media objects in the software. Within JD Edwards EnterpriseOne, you must set up media object queues to use images that are outside of the imaging product's domain (for example, scanned images). You can set up media object queues for these types of objects:

- Image objects (actual files)

- OLE objects (links to files)
- URLs (internet addresses)

Image Media Objects

Image media objects are individual files that are accessed and viewed by using a third-party imaging product. These objects are stored in locations defined with a name and a network-qualified path. For example, if all of the images for financial applications are stored in a directory on the network called`\\server1\financials\images`, an image media object queue could be defined as:

- Path: `\\server1\financials\images`
- Name: `FIN_IMAGES.BMP`

OLE Media Objects

OLE media objects are individual objects that are created and viewed by using an OLE-compliant application outside of JD Edwards EnterpriseOne. In JD Edwards EnterpriseOne, the OLE object attached to a row or form is actually a link to the OLE object that resides in a media object queue. The distinction between OLE objects and non-OLE objects is important because, other than graphics files, you cannot attach non-OLE objects from JD Edwards EnterpriseOne if they are not compliant. Examples of valid OLE objects are Microsoft Windows OLE-compliant applications such as Word, Excel, PowerPoint, and Visio. Other examples include sound or video files (.wav or .avi extensions).

URL Media Objects

URL media objects are internet addresses that point to web sites that are identified by industry-standard URLs. When defined in the media object table, these addresses can be connected to internet locations.

Media Object Tables

Media object queues typically represent network directory locations for JD Edwards EnterpriseOne media object files, such as OLE objects and images. The two media object tables are F98MOQUE and F98101.

The media object queues are stored in the Media Object Queues table, which, along with the Imaging Constants table, should be located in the system data source. The Media Object Queues table contains the associated key value of the data record to which the media object is attached, the image reference, and the OLE reference. The image reference and the OLE reference are queue names. The queue name is used to access the Media Object Queue table for the location of the OLE object or image.

Media object keys are stored in the F00165 table. Media object characterization properties are stored in the F00166 table. The F00167 table stores information indicating which categories the system activates for any given data structure.

Language Considerations for Media Objects

If you create a custom application that you want to enable for media object language handling, you must include a data item language preference (alias LNGP) in the generic text data structure that you create.

When you design an application, you can allow the end user to add separate and unique media objects to the same record or different records, based on the language chosen.

If language (LNGP) is not a database column, then you define the media object (GT) data structure to include language as part of the data structure. You place a data dictionary control (LNGP) on the application as a filter field, which should then be loaded with the system value for language. When you design the application this way, you attach two separate media objects, based on the language, to the same record.

If language (LNGP) is a database column, then you include LNGP (database) as a filter field, but you must add a separate record to the database table along with its media object attachment. The media object data structure still contains language as part of the key to retrieve the media object attachment. In both cases, the language filter fields (LNGP) must be loaded with the system value for language. LNGP must be built into the key and not associated with the LNGP column in the F00165 table.

For any database table that contains language as part of its key, you can attach media object functionality for records with different languages. For example, you can create one record for English and a copy of the record for French with unique media object attachments. For tables that do not include language as part of the key to that table, you can have media object languages.

CHAPTER 15

Setting Up Media Objects and Imaging

This chapter provides an overview of media object processing and discusses how to:

- Enable JD Edwards EnterpriseOne to use media objects.
- Add a language-specific media object attachment.
- Set up media object queues.
- Set up imaging.

Understanding Media Object Processing

To use media objects, JD Edwards EnterpriseOne requires a set of event rules to process the media objects. This processing includes:

- Tracking where the media object files are stored.
- Tracking which media objects are attached to which objects (rows, forms, and reports).
- Indicating which objects have attachments.
- Creating or viewing attachments.

You can set up JD Edwards EnterpriseOne to use standard processing for media objects, which enables you to bypass all event rules that are required to implement media objects. All of the required information is gathered from a form in Form Design Aid and does not require you to define any event rules. Standard processing provides these benefits:

- Standardizes the usage of media objects across forms.
- For any detail area, places a paper clip button on the row header if a media object is defined for that row.
- For a form, places a button in the status bar if a media object is defined for the form.
- Enables you to attach documents to the form or to a row in the detail area.
- Enables you to double-click the paper clip in a row to view media objects for that row.
- Enables you to click the paper clip in the status bar to view media objects for the form.

If you choose not to use standard processing for a form, you can still develop a system for handling media objects by using existing event rules or event rules that you develop.

JD Edwards EnterpriseOne uses the F00165 table to store link records for media objects and imaging. You must define the media object data structure by using a unique key structure so that the F00165 table can store data correctly. The layout of this table is as follows:

```
GTxxx || F4211Keys || The media object text
```

Where:

GT (generic text) *xxx* is the naming convention used when defining a media object data structure.

F4211Keys is what the system uses to access the unique media object attachment for that particular record. The keys typically match what the unique key would be in the F4211 table for each detail line.

The media object text is the actual text attachment that stores information typed in by the user.

In addition to the media object categories provided by JD Edwards EnterpriseOne, you can define as many as 40 more. Users can associate these categories with a media object to group certain media objects and to enable other users to search for specific media objects. User defined categories reside in the F00166 table and are referenced using each object's unique key. The default titles for these categories are Category Codes 1-30, Dates 1-5, and Numeric 1-5.

Prerequisite

To enable users to see the media object paper clip column on a form, clear the Hide Row Headers option on the grid properties for the form.

Enabling JD Edwards EnterpriseOne to Use Media Objects

In JD Edwards EnterpriseOne OMW, access an application on which you want to enable media objects and open it in Form Design Aid.

1. In Form Design Aid, select Media Objects Setup from the Form menu.
2. On Media Objects Setup, select the Enable Automatic Media Object Functionality option.

   Selecting this option enables imaging and activates the other fields on the form.
3. On Media Objects Setup, select one of these options:
   - Media Objects Only

     Select this option if you do not want to interface with third-party products that include imaging. If you choose this option, you will only be able to use media objects that are defined for and supported from within JD Edwards EnterpriseOne.
   - Document Handling Only

     Select this option if you are developing a form that is enabled for media objects using functionality in event rules and you want to bypass standard processing.
   - Media Objects & Document Handling

     Select this option if you want to enable standard processing later. You must delete all of the event rules for media objects when you choose this option.
4. Click Edit mode or Display mode.

   Edit mode allows the user to make changes; display mode is read-only.
5. Click Define Form Key.

   The System Functions form appears. This form is identical to the parameter definition form used to define system functions in event rules, except that it includes only the Media Object header.
6. On the Function Selection tab, double-click the Media Object Structures folder.

   A list of all currently defined data structures for Media Objects appears.
7. Select the appropriate structure and define it on the Parameter Mapping tab.

Adding a Language-Specific Media Object Attachment

Open the application to which you want to add a language-specific media object attachment.

1. On the application that you want to use, type a language in the filter field.
2. Click Add.
3. Add multiple records if you want the attachment for multiple languages or base.

Setting Up Media Object Queues

This section discusses how to:

- Add a media object queue.
- Define the location of a media object queue.
- Delete a media object queue.

Forms Used to Set Up Media Object Queues

Form Name	FormID	Navigation	Usage
Work With Media Object Queues	W98MOQUEA	System Administration Tools (GH9011), Media Object Administration, Media Object Queues (P98MOQUE)	Add a media object queue. Locate existing queues and delete queues.
Media Object Queue Revisions	W98MOQUEB	On the Work With Media Object Queues form, click Add.	Add information for a new media object queue. Define the location of a media object queue.

Adding a Media Object Queue

Access the Work With Media Object Queues form.

1. Click Add.
2. On the Media Object Queue Revisions form, in the Queue Name field, define the media object queue name where images may be found. The queue name is a symbolic reference only. The media object queue name is one-half of a properly defined media object queue. The other half is the media object path. complete these fields:
   - Queue Name
   - Queue Path On-Line
   - Queue Path Off-line
   - On-Line Access Type
   - Off-Line Access Type
3. In the Queue Path On-Line field, enter a path that points to the location of OLE objects, images, or URLs. The queue path is the second half of a properly defined queue for a media object. The

first half is the name of the media object. A valid queue path for a network location might be `\\server1\share3\images\financial`. To set the queue for a CD ROM, use `cd:` in the pathname; the system automatically substitutes the appropriate drive letter when it resolves the path.

4. In the Queue Path Off-line field, enter a local path that points to the location of OLE objects, images, or URLs. The queue path is the second half of a properly defined queue for a media object. The first half is the name of the media object. A valid queue path when working off-line might be `d:\data\media\images`.

Defining the Location of a Media Object Queue

Access the Work With Media Object Queues form.

1. If an OLE queue does not exist, click Add.
2. On the Media Objects Queue Revisions form, complete these fields:
   - Queue Name
   - Queue Path On-Line
   - Queue Path Off-line
   - Type
   - On-Line Access Type
   - Off-Line Access Type
3. Click OK.
4. If you want to change an existing media object queue, on the Work With Media Object Queues form, click Find to display a list of queue names and their paths.
5. Select the queue name that you want to modify and click Select.
6. On the Media Object Queue Revisions form, change the information in the Queue Path On-Line field to reflect the new location, and then click OK.

Deleting a Media Object Queue

Access the Work With Media Object Queues form.

1. Click Find.
2. Select the queue name that you want to delete.
3. From the Form menu, select Delete.

   Deleting a media object queue deletes only the definition of the queue, not the associated path or objects themselves.

Setting Up Imaging

This section contains an overview of imaging and the flow for imaging systems and discusses how to enable imaging in media objects.

Understanding Imaging

One way to attach images to JD Edwards EnterpriseOne forms and grid rows is to use the Image function of the Media Objects feature; however, this solution is not designed for use with sophisticated document handling systems. See the Customer Connection web site for a complete list of imaging vendors partnered with JD Edwards EnterpriseOne.

The software uses the OLE client/server model to interface with third-party document handling systems, including the OLE client interface and the OLE server. For the currently supported imaging systems, JD Edwards EnterpriseOne meets these minimum design goal tasks:

- Search

  The search mechanism locates a document stored in the indexing system of a document handling system. The search mechanism navigates the storage structures of the document handling system so that the user can find a particular document or set of documents easily.

- Link

  Upon a successful search operation, the link mechanism returns the unique document identifier to JD Edwards EnterpriseOne. This identifier is stored with the transaction.

- View

  The view mechanism passes the unique document identifier to a document viewing mechanism so that the user can view the document.

Customers with requirements for third-party imaging systems other than those that the software currently supports can design custom OLE automation servers for interfacing purposes. The OLE server can be written in any OLE-compliant language. JD Edwards EnterpriseOne has a published set of APIs to enable you to develop compatible middleware applications. The published APIs are described in a Windows help file that is installed with the software.

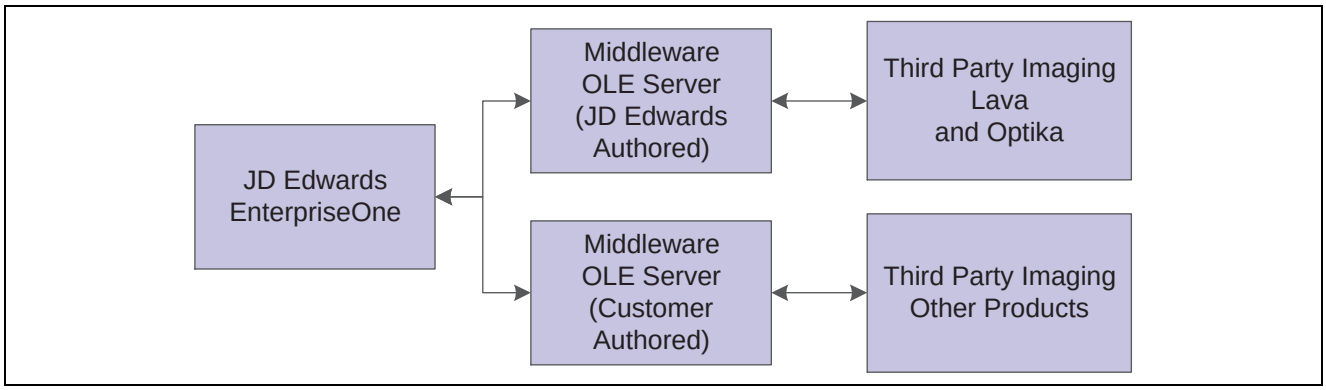
Understanding Flow for Imaging Systems

You can use imaging with a document handling system. The imaging system enables you to automatically scan and catalog documents. The system indexes the images so that you can recall them, based on certain sets of criteria. For example, you might index images according to type, department, and date. You can recall, view, and analyze an image at any given time. For example, in a transaction entry scenario, you might scan a paper-based file when the document enters the mailroom so that a data entry clerk can retrieve the image to use as a source document.

JD Edwards EnterpriseOne can retrieve and view documents based on selection criteria that are defined by the user. A linking system associates the JD Edwards EnterpriseOne transaction to the document for later retrieval and reference. You can attach a transaction identifier with the scanned image in the document handling system to enable a user to access an application directly from the image.

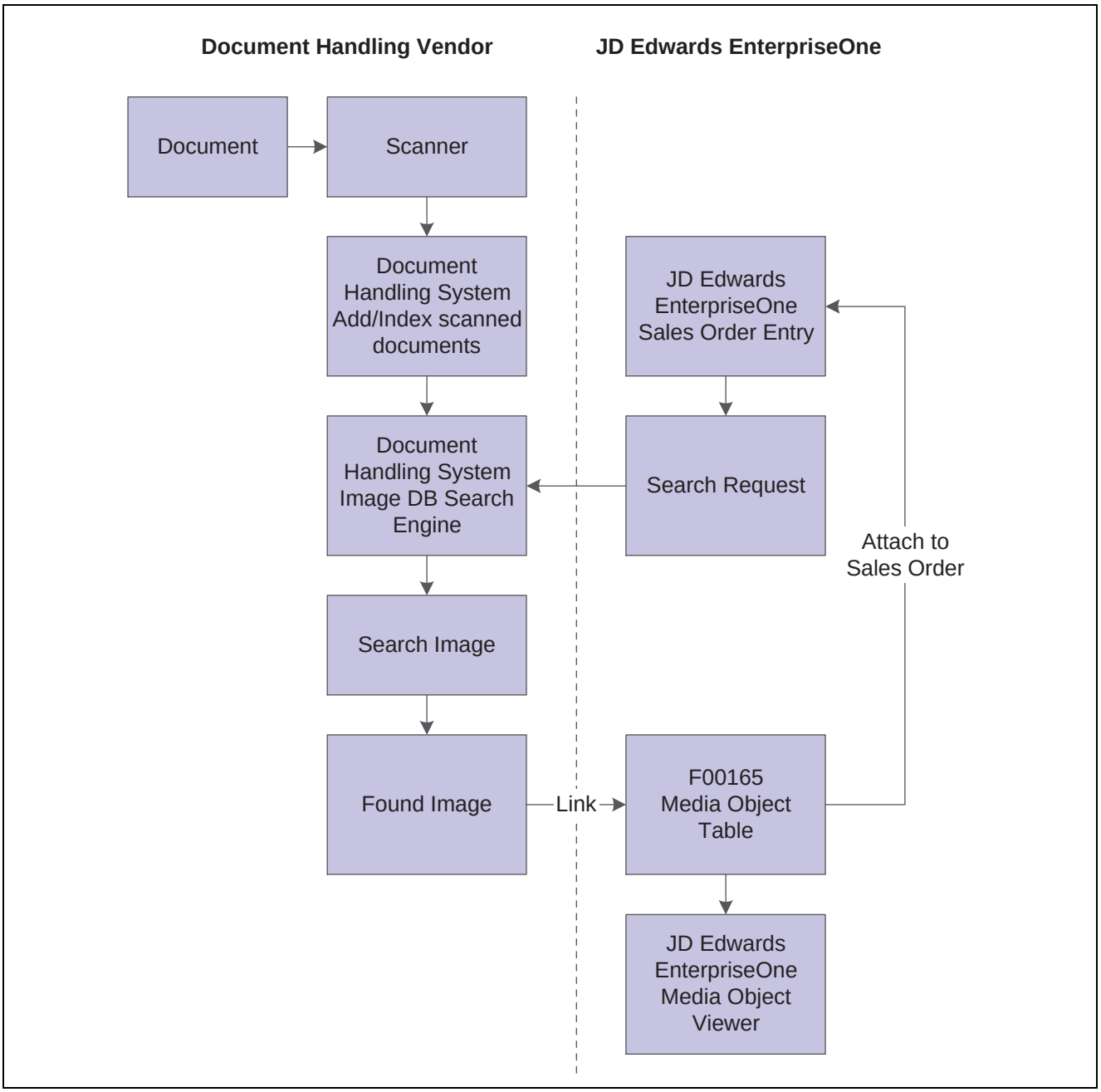
You set up imaging by enabling the imaging at the system level. For an imaging system to be enabled, it must have a registration record in the F98101 table.

This illustration explains how the software supports third-party imaging products through a middleware OLE server layer. Customers also can create their own OLE servers to support additional imaging systems:



JD Edwards EnterpriseOne imaging architecture

This diagram is a typical imaging scenario:



Imaging process flow

Forms Used to Set Up Imaging

Form Name	FormID	Navigation	Usage
Work With Media Object Queues	W98MOQUEA	System Administration Tools (GH9011), Media Object Administration, Media Object Queues (P98MOQUE)	Set up imaging.
Imaging Constants Revisions	W98101A	On the Work With Media Object Queues form, select Imaging from the Form menu.	Enter information about the imaging system and enable the imaging system.

Enabling Imaging in Media Objects

Access the Work With Media Object Queues.

1. From the Form menu, select Imaging.
2. On the Imaging Constants Revisions form, in the Imaging System Vendor field, enter the name of the imaging system vendor that you are using on the system.
3. In the Imaging Vendor Description field, enter a description of the imaging system vendor.
4. If the image type is an OLE, in the Image Program ID/DLL field, enter the program ID that uniquely identifies the imaging system in the system registry. If the image type is a DLL, enter the imaging system DLL name.
5. Select the Imaging Active option to indicate that an imaging system is currently active for the system.
6. Select the OLE option to indicate the type of interface used by the imaging system.

CHAPTER 16

Administering Text Search Indexes

This chapter provides an overview of Verity and discusses how to:

- Define text search indexes.
- Build text search indexes.
- Define text search properties.

Understanding Verity

Verity is the text search engine used by JD Edwards EnterpriseOne. Verity takes information from the tables and associated media objects and builds a text search index. When a text search is initiated, it goes through the text search index and returns database records that match the search criteria. In JD Edwards EnterpriseOne, Verity uses a business view-data source pair to retrieve table data.

Text search indexes exist as folders on a file system, so text search indexes must be accessible using a path name to the enterprise server. Administrators must designate the locations of these folders by specifying the base folder in Text Search Properties.

Defining Text Search Indexes

This section provides an overview of text search indexes and discusses how to:

- Designate data sources for a business view.
- Define media objects for a text search index.
- Define languages for a text search index.

Understanding Text Search Indixes

To create a text search index, define the following items:

- Business view

  To be able to perform a text search over a business view, the developer must select the Text Search option on the business view's properties form in JD Edwards EnterpriseOne OMW. Otherwise, the business view is not available for a text search.

  See *JD Edwards EnterpriseOne Tools 8.96 Development Tools: Data Access Tools Guide*.

- Data source

The data source tells Verity where to search for the data defined in the business view. If you do not select a data source, Verity searches the default data source defined in the OCM mapping.

- Media objects

  To associate media objects with a business view-data source pair, use the Text Search Index Media Object Queue Revisions form. If no queues are specified, then all queues will be scanned.

- Languages

  To build the index for any languages other than your base language, enter the language on the Text Search Index Language Revisions form.

Forms Used to Define Text Search Indexes

Form Name	FormID	Navigation	Usage
Work With Text Search Indices	W95800B	Text Search Administration menu (GH9077), Text Search Indices	View and select text search business views with the associated data sources. Build, clear, and optimize text search indexes.
Text Search Index Revisions	W95800A	On the Work with Text Search Indices form, select a business view.	Edit data sources for each business view.
Text Search Index Media Object Queue	W95800E	On the Work with Text Search Indices form, highlight a data source, then select Media Object Queues from the Row menu.	Edit text search indexes for media object queues
Text Search Index Language Revisions	W95800F	On the Work with Text Search Indices form, highlight a data source, then select Languages from the Row menu.	Edit text search indexes for language.

Designating Data Sources for a Business View

Access the Text Search Index Revisions form.

Text Search Business View Displays the business view selected using the Work with Text Search Indices form.

Text Search Data Source The text search data source defined for the Text Search business view.

Defining Media Objects for a Text Search Index

Access the Text Search Index Media Object Queue Revisions form.

Text Search Business View The business view selected using the Work with Text Search Indices form.

Text Search Data Source The data source selected using the Work with Text Search Indices form.

Media Object Queue Name The name of the media object queue from the table F98MOQUE.

Defining Languages for a Text Search Index

Access the Text Search Index Language Revisions form.

Text Search Business View	The business view selected using the Work with Text Search Indices form.
Text Search Data Source	The data source selected using the Work with Text Search Indices form.
Language	The user defined code indicating the language.
Description	The name of the language.

Building Text Search Indexes

There are four types of builds you can perform:

- Full
- Incremental
- Optimize
- Clear

A full build indexes all the data referenced by a business view and creates a new text search index or replaces an existing text search index. You should schedule full builds regularly, depending on the frequency with which the data changes.

Note. If a user adds a new media object to a specific record in a business view and as a result, exceeds the media object limit defined in the previous full build, the change will only take affect after the next full build is complete. In other words, the incremental build will not add this change in the Verity index for this business view.

Some applications automatically perform incremental builds. An incremental build takes data entered since the last full build and adds it to the end of the text search index. This makes the data available for searching, but each time an incremental build runs, it decreases the efficiency of the text search index. To optimize the efficiency of the text search index, you need to run an optimize build.

An optimize build takes the data entered since the last full build and re-indexes it. This increases text search efficiency and ensures that text searches cover recently added data. An optimize build is not as resource-intensive as a full build, so you can schedule it more frequently.

A clear build removes all data from the text search index. Under normal circumstances you would not clear a text search index, except to free file system storage. Before you can search again, you need to do a full build. However, if you do a search after a clear build, there is no data in the index to return.

You can perform builds manually or schedule them to occur automatically. The manual Build options are off the Report and Row menus. Schedule builds using the JD Edwards EnterpriseOne Scheduler by running the following UBEs:

- R958001 — XJE001 (Full Build).
- R958001 — XJDE002 (Optimize).
- R958001 — XJDE003 (Clear).

Defining Text Search Properties

This section provides an overview of text search properties and discusses how to:

- Add a new text search business view-data source pair.
- Define stop words.
- Define synonyms.
- Define topics.

Understanding Text Search Properties

There are three properties you can define for a Verity search:

- Stop Words

  Stop words are words that are too common to search for. For example, if every service ticket uses the word “broken”, you would not search on “broken”. Similarly, if you only sold automobiles, searching sales receipts for the words “automobile” or “car” would be useless. Instead, you might search for specific makes and models of automobiles. By making “car” and “automobile” stop words, if you entered the search “1998 Ford Mustang Car”, the system would only search for “1998 Ford Mustang”, taking out the common word, “car”. Stop words reduce the required file system storage for text search indexes and improve search performance.

- Synonyms

  Synonyms are words that mean the same thing. For example, in a motorcycle dealership, the words “bike”, “chopper”, and “hog” all refer to a motorcycle. Motorcycle is the thesaurus word, that is the word all the synonyms point to. “Bike”, “chopper”, and “hog” are the synonyms. When a user enters the search word “chopper”, all records containing “motorcycle”, and any of its synonyms are returned.

- Topics

  To define topics you need Verity software not provided with JD Edwards EnterpriseOne. Please reference the documentation Verity provides for more information on topics and topic outline files.

Verity has predefined common stop words and synonyms for each language supported. You only need to define those that are specific to your business.

You can define synonyms and stop words globally or for specific business view-data source pairs.

Forms Used to Define Text Search Properties

Form Name	FormID	Navigation	Usage
Work with Text Search Properties	W95820A	Text Search Administration menu (GH9077), Text Search Properties	View, add, and select text search business views-data source pairs and define maximum results and base folders.
Text Search Properties Revisions	W95820B	On the Work with Text Search Properties form, click Add.	Add a new text search business view-data source pair and define maximum results and base folder.
Work with Text Search Stop Words	W95820E	On the Work with Text Search Properties form, select Stop Words from the Row or Form menu.	When entered from the Row menu, view the stop words defined for a text search business view-data source pair or global stop words. When entered from the Form menu, view and edit stop words already set up in the system.
Work with Text Search Synonyms	W95820C	On the Work with Text Search Properties form, select Synonyms from the Form or Row menu.	When entered from the Row menu, view the synonyms defined for a text search business view-data source pair, or global synonyms. When entered from the Form menu, view and edit synonyms already set up in the system.
Work with Text Search Topics	W95820I	On the Work with Text Search Properties form, select Topics from the Form or Row menu.	When entered from the Row menu, view the topics defined for a text search business view-data source pair, or global topics. When entered from the Form menu, view and edit topics already set up in the system.

Adding a New Text Search Business View-Data Source Pair

Access the Text Search Properties Revisions form.

Text Search Business View The name of the business view of the business view-data source pair from which to build the text search indexes.

Data Source The name of the data source of the business view-data source pair from with to build the text search indexes.

Base Folder The folder to contain the text search index. This folder must be accessible to the enterprise server.

Max Results	The maximum number of matches to be returned from this text search index.

Defining Stop Words

Access the Work with Text Search Stop Words form.

Global	This option defines stop words for all business view-data source pairs.
Business View/Data Source Specific	This option defines stop words for only one business view-data source pair.
Text Search Business View	The business view used to build the text search index.
Text Search Data Source	The database used to build the text search index.
Language	The language defined for the text search index.

Defining Synonyms

Access the Work with Text Search Synonyms form.

Global	This option defines synonyms for all business view-data source pairs.
Business View/Data Source Specific	This option defines synonyms for only one business view-data source pair.
Text Search Business View	The business view used to build the text search index.
Text Search Data Source	The database used to build the text search index.
Language	The language defined for the text search index.

Defining Topics

Access the Work with Text Search Topics form.

Global	This option defines topics for all business view-data source pairs.
Business View/Data Source Specific	This option defines topics for only one business view-data source pair.
Text Search Business View	The business view used to build the text search index.
Data Source	The database used to build the text search index.
Language	The language defined for the text search index.

CHAPTER 17

Setting Up Application Failure Recovery

This chapter provides an overview of application failure recovery and discusses how to:

- Assign an administrator for the application failure recovery applications.
- Grant user access to failed application data.

Understanding Application Failure Recovery

JD Edwards EnterpriseOne web client users to recover data from failed applications due to:

- Catastrophic errors
- Transaction failures
- Session time outs
- Voluntary exits

The Application Failure Recovery program (P95400) enables users to access and recover data from any failed transaction in which they are involved. Using P95400, users can save and copy the data from failed transactions back into the appropriate application to complete the transaction. However, users must be granted permission by an administrator to recover data from applications other than their own. For example, an administrator might give a sales department supervisor the permission to recover data from transactions performed by other users in the department.

To set up application failure recovery, you must first use the Application Failure Administration application (P95410) to assign an application failure administrator. The administrator can grant users permission to recover failed application data from application transactions entered by other users.

See Also

JD Edwards EnterpriseOne Tools 8.96 Foundation Guide, "Using JD Edwards Web Applications and Reports," Recovering Data

Prerequisite

Use the Security Workbench program (P00950) to secure P95410 to system administrators only.

See *JD Edwards EnterpriseOne Tools 8.96 Security Administration Guide*, "Using Security Workbench," Managing Application Security.

You must configure the AppRecovery setting in the [OWWEB] section of the jas.ini file for the system to save the data from a failed application.

See “Parameters and Values for the jas.ini File” in the *JD Edwards EnterpriseOne Tools 8.96 HTML Web Server Installation Guide*.

Assigning an Administrator for the Application Failure Recovery Applications

Access the Work With Application Failure Administrators form. In the JD Edwards EnterpriseOne web client, enter *P95410* in the Fast Path to access the Work with Application Failure Administrators form.

1. Click Add.
2. On the Add Application Failure Administrator form, in the User field, enter the user ID of the individual that you want to assign as administrator, and then click OK.

Granting User Access to Failed Application Data

Access the Work with Application Failure Records form. In the JD Edwards EnterpriseOne web client, enter *P95400* in the Fast Path.

As an application failure administrator, you can allow users to recover failed application data for specific applications. The P95400 application enables you to grant this access to a user, a role, or all users.

1. From the Form menu, select Time Out Subscriptions.
2. On the Work with Time Out Subscriptions form, click Add.
3. On the Add Time Out Subscription form, in the User field, enter the user ID or role that you want to allow access to the failed application data. Enter **Default* to allow access to all users.
4. In the Application Name field, enter the application that the user or role can recover data from, and then click OK.

CHAPTER 18

Using the Universal Table Browser

This chapter provides discusses how to work with the Universal Table Browser.

Understanding the Universal Table Browser

To view the data in tables in different databases, use the Universal Table Browser. This tool lets you verify the existence of data in a table, as well as determine the structure of the table. The Universal Table Browser uses JDEBASE APIs to retrieve data from the database, making it independent of the database that you access.

You can determine whether the data that the Universal Table Browser displays is formatted or non-formatted.

For formatted data, the Universal Table Browser displays the data according to the specifications of the JD Edwards EnterpriseOne data dictionary item. For example, assume that the data item PROC is a numeric field of size 15, with four display decimals. For a value of 56.2185, the Universal Table Browser displays a formatted value (using the data dictionary editing) as 56.2185, even though this value is stored in the database as 562185.

For non-formatted data, the Universal Table Browser displays the data according to the specification of the database and the data item type (such as numeric) from which the data came. For example, assume that the table data item PROC is a numeric field stored in the database. Depending on the database, this field might have a default value size of 32, with a precision of 15 being a numeric data type. Because JD Edwards EnterpriseOne does not store the decimals in the database, a value 56.2185 is stored and displayed in the database as 562185.000000000000000.

The tables that display in the Universal Table Browser contain the query by example (QBE) feature, which functions as it does in any other JD Edwards EnterpriseOne application. For example, you can enter *>50* in the ABAN8 column QBE to display records with an address book number that is greater than 50. You can enter *F** in the ABALPH column QBE to display records with an alpha name that begins with the letter F.

In addition, the column sequence and column width features function as in any other JD Edwards EnterpriseOne application. You can rearrange the columns. For example, you might want to move a column that you use often from the end to the beginning, or move a column next to an associated column. You also can size columns.

Working with the Universal Table Browser

This section discusses how to:

- View the data in tables.
- View column properties in a table.

Viewing the Data in Tables

Access the Universal Table Browser. In Solution Explorer, select the Cross Application Development Tools (GH902) menu, Object Management, Universal Table Browser.

Note. All column and row security that you set up using Security Workbench applies to the Universal Table Browser.

---

1. On Universal Table Browser, select Open Table from the File menu.
2. On the Table and Data Source Selection form, in the Table field, click the search button to select a table.
3. In the Data Source field, click the search button to select a valid data source in which the table resides. This default value is obtained from the Object Configuration Manager (OCM) settings in the current environment.
4. Select the Format Data option if you want the program to display formatted data.

Viewing Column Properties in a Table

Access the Universal Table Browser. In Solution Explorer, select the Cross Application Development Tools (GH902) menu, Universal Table Browser.

1. On Universal Table Browser, view a table as described in the previous task.
2. Right-click a column and select Column Properties.

   If you are viewing a formatted table, the data dictionary properties are displayed in the upper-right portion of the Column Properties form. If you are viewing an unformatted table, the data dictionary properties are not displayed.

CHAPTER 19

Working with Flat File Encoding

This chapter provides an overview of flat file encoding, an example of how to set up a flat file encoding record, and discusses how to:

- Add a flat file encoding record.
- Activate a flat file encoding file.

Understanding Flat File Encoding

Because JD Edwards EnterpriseOne software uses Unicode and not all third-party software does, there is a preprocessing and postprocessing intercept of all flat files. During the intercept, the software converts the flat file into the Unicode character set or back into the original character set. You can assign the conversion character set applied to a flat file—based on the user or role, the program ID, the program version, and the environment—by adding and activating a flat file encoding record.

Using Unicode Flat File Encoding Configuration (P93081), you create records for a table that specifies what character sets are used for programs. The character sets are based on the user or role, the program ID, program version, and the environment. When the pre- or post-processing intercept occurs, the intercept program calls the table, searches it, and applies the record. The search is from more specific records to less specific records.

The primary users of Unicode Flat File Encoding Configuration are power users and system administrators. The business manager can provide the character set that is used to encode the third-party flat file.

Before setting up a flat file encoding record, you need to know the encoding of the flat file being transferred. You also need to know the user or role, program, program version, and environment that is calling the flat file.

To ensure that all files are encoded to the primary character set, set up a default flat file encoding record for the primary character set, and then add any exceptions. The system applies the more specific records before the more general records, so the default record is only used if no other records apply to the incoming flat file. If you do not add and activate a flat file encoding record, the default record is UCS2, UTF16_BE/UTF16_LS, which is a Unicode character set.

This table displays the character sets, from user defined code H95/FE, that are currently supported:

Code	Description	.ini Setting
BIG5	Chinese, Traditional	TC_BIG5
CP1250	WIN-Latin2, Central Europe	EE_CP1250
CP1251	WIN-Cyrillic	RS_CP1251
CP1252	WIN-Latin 1, Western European	WE_ISO88591

Code	Description	.ini Setting
CP1253	WIN-Greek	GR_CP1253
CP1254	WIN-Latin5, Turkish	TK_CP1254
CP1256	Win-Arabic	AR_CP1256
GB2312	Chinese, Simplified	SC_GB
IBM-1123	EBCDIC-Cyrillic	RS-EBCDIC
IBM-420	EBCDIC-Arabic	AR_EBCDIC
IBM-933	EBCDIC-Korean	KO_EBCDIC
IBM-935	EBCDIC-Simplified Chinese	SC_EBCDIC
IBM-937	EBCDIC-Traditional Chinese	TC_EBCDIC
IBM-939	EBCDIC-Japanese	JA_EBCDIC
IBM-37	EBCDIC-Latin 1 or Western Euro	US_EBCDIC
KSC-5601	Korean	KO_KSC
SHIFT_JIS	WIN-Japanese	JA_SJIS
UCS2	UTF16_BE/UTF16_LE	
UTF-16LE	UTF16_LE	
UTF-16BE	UTF16_BE	
UTF8	UTF8	
IBM-858	# PC Latin 1 with Euro	

Example: Setting Up Flat File Encoding Records

The example company primarily uses the flat file encoding character set CP1252, WIN-Latin 1, Western European. However, the Sales Order Entry program (P42101) uses the UTF8, Unicode character set, except when user JL5534221 runs version JDE0001 in the PDEVCLA environment; then the program uses the CP1250, WIN-Latin2, Central Europe character set. The same program occasionally, but not currently, uses the character set CP1254, WIN-Latin5, Turkish for environment PDEVASD2.

This table presents the information for the encoding records that the example company needs in the flat file encoding table:

User/Role	Application Name	Application Version Name	Environment	Encoding Name	Status
All	All	All	All	CP1252	Active
All	P42101	All	All	UTF8	Active
JL5534221	P42101	JDE0001	PDEVCLA	CP1250	Active
All	P42101	All	PDEVASD2	CP1254	Inactive

These steps provide instructions on how to set up flat file encoding records for the previous example:

Access the Work With Flat File Encoding form. In Solution Explorer, select the System Administration Tools, System Administration Tools, User Management, User Management Advanced and Technical Operations, Unicode Flat File Encoding Configuration.

1. On the Work With Flat File Encoding form, click Add.
2. On the Flat File Encoding Revisions form, complete these fields, and then click OK:
   - User / Role

     **PUBLIC* includes all users and roles.
   - Environment

     **ALL* includes all environments.
   - Program ID

     **DEFAULT* includes all programs.
   - Version

     **DEFAULT* includes all program versions.
   - Encoding Name

     Enter the following value: *CP1252.*
3. Repeat the previous step to add the following records to the table:

User/Role	Environment	Program ID	Version	Encoding Name
*PUBLIC	*ALL	*DEFAULT	*DEFAULT	CP1252
*PUBLIC	*ALL	P42101	*DEFAULT	UTF8
JL5534221	PDEVCLA	P42101	JDE0001	CP1250
*PUBLIC	PDEVASD2	P42101	*DEFAULT	CP1254

4. Click Cancel to return to Work With Flat File Encoding.
5. Click Find to display all of the flat file encoding records.
6. Select the first record and from the Row menu, select Change Status to activate the recode.
7. Repeat the previous step to activate the currently active records.

   This table displays the final configuration:

User/Role	Environment	Program ID	Version	Encoding Name	Status
*PUBLIC	*ALL	*DEFAULT	*DEFAULT	CP1252	AV
*PUBLIC	*ALL	P42101	*DEFAULT	UTF8	AV
JL5534221	PDEVCLA	P42101	JDE0001	CP1250	AV
*PUBLIC	PDEVASD2	P42101	*DEFAULT	CP1254	NA

Work with Flat File Encoding Records

This section discusses how to:

- Add a flat file encoding record.
- Activate a flat file encoding record.

Forms Used to Work with Flat File Encoding Records

Form Name	FormID	Navigation	Usage
Work With Flat File Encoding	W93081A	System Administration Tools (GH9011), Unicode Flat File Encoding Configuration	Access the form to add a flat file encoding record. Locate the defined flat file encoding files.
Flat File Encoding Revisions	W93081B	On the Work With Flat File Encoding form, click Add.	Enter the information for a flat file encoding record.

Adding a Flat File Encoding Record

To define the character set that is applied to a flat file during the pre- or post-processing intercept, add a flat file encoding record. You can apply a flat file encoding file based on the user, the user role, the program ID, the program version, and the environment.

After you add the flat file encoding record, you must activate it.

Access the Work With Flat File Encoding form.

1. Click Add.
2. On the Flat File Encoding Revisions form, complete these fields, and then click OK:
   - User / Role

     The default user/role is **PUBLIC,* which includes all users. By specifying a user or role, you can limit flat file encoding to only programs running under that user or role.
   - Environment

     The default environment is **ALL,* which applies the character set encoding to all environments. By specifying an environment, you can limit the flat file encoding to only programs running under that environment.

- Program ID

  The program ID identifies the batch or interactive application to which to apply the flat file encoding. The default value, **DEFAULT,* applies flat file encoding to all programs.

- Version

  A version is a set of user defined specifications that determines how a batch or interactive application runs. A program version identifies the batch or interactive application version to which to apply the flat file encoding. The default version, **DEFAULT,* applies the flat file encoding file to all versions.

- Encoding Name

  The encoding name identifies the character set used by the incoming or outgoing flat file. You must specify an encoding name.

Activating a Flat File Encoding Record

After adding a flat file encoding record, you must activate it before it will be applied to incoming and outgoing flat files.

Access the Work With Flat File Encoding form.

1. Click Find to display the defined flat file encoding files.
2. Select the flat file encoding file to activate or deactivate.
3. From the Row menu, select Change Status.

   The status of the flat file encoding becomes active (*AV*) or inactive (*NA*).

CHAPTER 20

Understanding Naming Conventions

This chapter discusses:

- Path codes
- Data sources
- Package names
- Server names
- Workstation names

Path Codes

The naming conventions for a path code are as follows:

- Limited to 10 characters.
- Letters must be uppercase only.

Data Sources

The naming conventions for a data source are as follows:

- Limited to 30 characters.
- Case-sensitive.

Specific naming convention exceptions for the Client Access data source are as follows:

- Limited to 32 characters.
- Must begin with an alphabetic character.
- You cannot use the following characters:
- { }
- [ ]
- ( )
- ?
- *

- =
- !
- @
- ;

Note. You must type the data source name before you can use the Client Access ODBC driver to access iSeries data.

Data Source Description

Limited to 80 characters.

Package Names

The naming conventions for a package are as follows:

- Limited to 10 characters.
- Uppercase only.
- You cannot use the following characters:
- /
- \
- :
- *
- ?
-
- <
- >
- |

Server Names

The naming conventions for a server depend on the specific platform. For example, an HP9000 and an iSeries allow you to enter different characters when you define the server name. JD Edwards EnterpriseOne limits the number of characters that you can use to name a server to 15, regardless of the platform.

Workstation Names

Use the following naming conventions for a workstation:

- Limited to 15 characters.
- Each workstation requires a unique name.

  When you add a workstation to a Windows Server domain, you must use the name created for the computer by the network administrator.

  If the workstation name does not have a computer account in the domain, you cannot sign in to the domain or access any domain user accounts.

CHAPTER 21

Understanding the jde.ini File Settings

This chapter provides an overview of the jde.ini file (on the iSeries, it is known as the INI file) and discusses:

- jde.ini file location.
- Workstation jde.ini settings.
- iSeries server jde.ini settings.
- UNIX/Linux server jde.ini settings.
- Windows enterprise server jde.ini settings.
- Server jde.ini settings for WebSphere.

Understanding the jde.ini File

The jde.ini file is an initialization file that provides runtime settings for JD Edwards EnterpriseOne. Specific versions of the file must reside on every workstation and enterprise server in the installation.

The jde.ini is divided into sections with informational headings. Each section heading is enclosed in square brackets, such as [JDENET]. Each section contains one or more keys. The key name is on the left side of the equal (=) sign; the value of the key is on the right side.

The workstation jde.ini file can be accessed three ways:

- Access Windows Explorer, locate the jde.ini file, and double-click it to open it.

  Use Notepad to view the file.
- Click the Windows Start button, and select Run from the list of options.

  Type *jde.ini* in the Open field.
- Type *jde.ini* in the Fast Path field of Solution Explorer.

Locating the jde.ini File

You can locate the jde.ini file in various places, depending on the platform.

Other JD Edwards EnterpriseOne INI Files

Besides jde.ini file, the jas.ini and the jdbj.ini files contain important settings for configuring your JD Edwards EnterpriseOne installation.

The jas.ini file defines the startup Object Configuration Manager and other web server-specific properties for the web server to communicate back to the enterprise environment. It is also the key to the installation. If any critical settings are incorrect or left blank, the web server does not run.

The jas.ini file is automatically updated during web server installation using the Java Server Installer. However, you can modify the jas.ini settings after installation by using a text editor. The jas.ini file is functionally similar to the jde.ini file on the enterprise server. Refer to the "Parameters and Values for the jas.ini File" section in the JD Edwards EnterpriseOne Tools 8.96: HTML Web Server Installation guide

The jdbj.ini file contains configuration information for JDBj, which allows the web server to access JD Edwards EnterpriseOne databases. The parameters in this file are only used for accessing JD Edwards EnterpriseOne data and are ignored for standalone database access.

The critical parameters of the jdbj.ini file are automatically updated during the installation of the web server using the Java Server Installer. However, you can modify these settings after installation using a text editor. The jdbj.ini file is functionally similar to the jde.ini file on the JD Edwards EnterpriseOne enterprise server.

See "Parameters and Values for the jdbj.ini File" in the *JD Edwards EnterpriseOne Tools 8.96 HTML Server Installation Guide*.

Workstation jde.ini Settings

This section discusses the settings found in the client-side workstation jde.ini file. Information is organized by section—for example, [DEBUG]. Sections are alphabetized, but settings found within sections are listed in the order in which they are found in the software.

The jde.ini file is located in the default Windows directory of the workstation. This directory might have a variety of names, depending on the type of operating system being used. If you are using Windows 2000, the default directory might be called Win2000.

[ACTIVE DIRECTORY]

The ACTIVE DIRECTORY settings are:

Setting	Value	Purpose
JdenetSCP	Variable	The value is the name of the Service Connection Point object in Active Directory. The SCP enables the workstation to connect to a server that has JD Edwards EnterpriseOne running on it. Typically, the name is the name of the JD Edwards EnterpriseOne service running on the server, such as: JDEdwards_ERP_811_SP1. JdenetSCP is the connection port parameter.
SecurityServerSCP	Variable	Same as previous. SecurityServerSCP is the security server parameter.

Setting	Value	Purpose
LockManagerSCP	Variable	Same as previous. LockManagerSCP is the Lock Managerkadol parameter
UnifiedLogonServerSCP	Variable	Same as previous. UnifiedLogOnServer SCP is the unified logon server parameter.

[DB SYSTEM SETTINGS]

The settings in this section contain information about the default environment and path code. A directory must reside on the workstation that has the same name as the default path code shown in its jde.ini file. The name of the server can also be found in this section:

Setting	Value	Purpose
Version=	43	A version number to prevent mismatch of the jde.ini file with the running version of JD Edwards EnterpriseOne.
Default User=	JDE	The user account name for the database bootstrap tables.
Default Env=	ADEVCLA	The default environment on the workstation or the enterprise server.
Default PathCode=	PROD	The name of a subdirectory under \811 that the software uses to find specifications to display sign-in information before an environment is selected.
Base Datasource=	System - 811	The data source representing the database from which logon information is retrieved.
Object Owner=	Object/owner	The owner of the system database tables.
Server=	Server name	The server on which the database resides
Database=		The name of the Oracle connect string or the ODBC datasource name for iSeries.
Load Library=		The PSFT driver that is used to access the database that stores the system tables. This value is set dynamically by JD Edwards EnterpriseOne runtime.
Decimal Shift=	N (default) Y	A flag to indicate if decimal shifting is used for numeric data.
Julian Dates=	N (default) Y	A flag to indicate if dates are stored in Julian or database-specific format.

Setting	Value	Purpose
Use Owner=	N (default) Y	A flag to indicate that table names are to be qualified by owner.
Secured=	N (default) Y	Indicates whether the database is a secured database that requires a user and password login.
Type=	A (default) O I L W M S	A single character denoting the type of database holding the system tables. These characters can be O (Oracle), A (MS Access), I (Client Access, iSeries), L (SQL Server OLEDB), W (DB2 UDB for Windows/Unix), S (SQL Server), M MSDE/OLEDB OR N (MSDE/ODBC).
DatabaseName2=		ODBC name for the SQL Server database or iSeries library database name
DatabaseInstance=		Name of the SQL Server database instance if using multiple instances. Leave this setting blank if using a single instance.
ServerPort=		The port number of the SQL Server database port
UnicodeFlag=	N (default) Y	Indicates whether Unicode is used on the datasource. Set this to Y if using Unicode.
LOBFlag=	Y	For Oracle and iSeries. Indicates that LOBs are used in the datasource instead of BLOBs. This value should always be Y.
Default Pwd=		The default password.
Default Journal=	OW_JRNL	iSeries only. The name of the default journal. Journaling is required on the iSeries for rollback recovery. The two components to journaling are: • The journal • The journal receiver Both before and after images of a database transaction can be recorded by journaling. Journaling can be set to any character string of 10 characters or fewer.
Default Journal LIBRARY=	Journal library	iSeries only. The library name where the journal is stored, which can be set to any valid library name. The library name changes for each release.

Setting	Value	Purpose
Default Journal Receiver	OW_JRNL000	iSeries only. The name of the journal receiver, which can be set to any character string of 10 characters or fewer.
Default Journal Receiver LIBRARY=	Journal library	iSeries only. The library name where the journal receiver is stored, which can be set to any valid library name. The library name changes for each release.
Size of Journal Receiver=		iSeries only.
ThousandsSeparator=	,	Sets the default character for ThousandsSeparator; the default can be set to any character except a number. This value should match the ThousandsSeparator that is specified by the client operating system. **Note.** The INI file does not support the use of a space. If a space or non-blocking space must be specified, use the strings SPACE or NB_SPACE instead.
DecimalSeparator=	.	Sets the default character for DecimalSeparator; the default can be set to any character except a number. This value should match the DecimalSeparator that is specified by the client operating system. **Note.** The INI file does not support the use of a space. If a space or non-blocking space must be specified, use the strings SPACE or NB_SPACE instead.

[DB SYSTEM SETTINGS - SECONDARY]

This section is used for workstations only. The settings are used for a secondary data source to start JD Edwards EnterpriseOne if the primary data source is unavailable. These settings should be the same as the values in the F98611 table for the secondary data source:

Setting	Typical Value	Purpose
Base Datasource=	Access32	The data source representing the database from which logon information is retrieved.
Object Owner=		The database owner of the system tables.
Server=	Server name	The server on which the database that stores the system tables resides.
Database=	Access32	The name of the database that stores the system tables.
Load Library=	JDBODBC.DLL (default)	The JDE driver that is used to access the database holding the system tables.

Setting	Typical Value	Purpose
Decimal Shift =	N (default) Y	A flag to indicate if decimal shifting is used for numeric data.
Julian Dates=	N (default) Y	A flag to indicate if dates are stored in Julian or database-specific formats.
Use Owner=	N (default) Y	A flag to indicate that table names are to be qualified by owner.
Secured=	N (default) Y	A flag to indicate whether database is securing, requiring user and password login.
Type=	A (default) O S I	A single character denoting the type of database that stores the system tables. These characters can be: O (Oracle), A (MS Access), S (SQL Server), or I (Client Access, iSeries).
Library List=		iSeries only. Database server that stores the system tables.
Library=		iSeries only. Database library that stores the system tables.

[DEBUG]

The settings in this section determine the location of the jde.log and jdedebug.log. The settings are also used to turn the jdedebug.log on and off:

Setting	Typical Value	Purpose
TAMMultiUserOn=	0	
Output=	None	Controls the status of the jdedebug file. Valid values are: • NONE. No trace information is written to jdedebug.log. • FILE. Database and runtime trace information are written to the file that is specified by the DebugFile= parameter in the [DEBUG] section. • EXCFILE. Runtime trace information is written to the file that is specified by the DebugFile= parameter in the [DEBUG] section. • BOTH. Trace information is written to both jde.log and jdedebug.log.

Setting	Typical Value	Purpose
ServerLog=	0 (default) 1	Valid values are: • 0. Disables the workstation requesting the business function JDE.LOG and JDEDEBUG.LOG entries from the server. • 1. Enables workstation requesting business function JDE.LOG and JDEDEBUG.LOG entries from the server.
LEVEL=	BSFN,EVENTS	Controls the debug level. You can specify any combination of allowable values using commas as delimiters. The default setting is LEVEL=BSFN,EVENTS. Valid values are: • EVENTS. Traces the starting and stopping of events. • BSFN. Traces when business functions are entered and when they return. • SF_x. Traces when system functions execute. The x variable is any allowable system function value. Valid values are: • SF_GRID • SF_PARENT_CHILD • SF_GENERAL • SF_MESSAGING • SF_WORKFLOW • SF_WORKFLOW_ADMIN • SF_MEDIA_OBJ • SF_CONTROL For example, LEVEL=SF_CONTROL. In addition, you can specify multiple system functions by separating them with commas. For example, LEVEL=SF_GRID,SF_CONTROL.
DebugFile=	c:\jdedebug.log	The location and name of the jdedebug.log file.
JobFile=	c:\jde.log	The location and name of the jde.log file.

[EVEREST]

The EVEREST setting is:

Setting	Typical Value	Purpose
ShowAlias=	0 (default for PROD packages) 1 (default for APPL packages)	This setting disables (0) or enables (1) the ability to right-click a data dictionary item and display its alias.

[INSTALL]

The settings in this section contain directory paths and general installation information:

Setting	Typical Value	Purpose
DefaultSystem=	System	The name of the subdirectory under \811 that contains the foundation code and tools.
ClientPath=	EnterpriseOne Client Install	The name of the directory on the deployment server that contains the Workstation Installation program and other files that are used during deployment.
PackagePath=	Package	The name of the subdirectory on the deployment server under a path code that contains the packages that were built for that path code.
DataPath=	Data	The name of the subdirectory on the deployment server under the path code that contains the Access database that is delivered for all packages for that path code.
HOSTS=	Hosts	The name of the directory on the deployment server that contains all of the types of host files. Used in the host configuration generate application.
HP9000=	hp9000	The name of the directory on the deployment server that contains HP9000 files. Used in the host configuration generate application.
RS6000=	rs6000	The name of the directory on the deployment server that contains RS/6000 files. Used in the host configuration generate application.
AS400=	as400	The name of the directory on the deployment server that contains iSeries files. Used in the host configuration generate application.
SUN=	Sun	The name of the directory on the deployment server that contains Sun files. Used in the host configuration generate application.
LocalCodeSet=	WE_ISO88591	A setting that is used to determine alternate language usage. See the appropriate JD Edwards EnterpriseOne Tools 8.12 Upgrade Guide for other language values.

Setting	Typical Value	Purpose
ActiveConsole	0 1	If this setting is 0, the package build does not add the entry to the package.inf file. If this setting is 1, an ActivEra Console shortcut is added to the package build .inf file. When the package is installed to a workstation, the shortcut is added to the desktop.
WebAdmin=	0 1	A setting of 1 gives the user administrative rights to the Java & HTML Generator so that the administrator can generate any Java serialized object publicly. A setting of 0 means that the user can only generate personal forms and menus using the Java & HTML Generator.

[JDE_CG]

The JDE_CG section settings are:

Setting	Typical Value	Purpose
STDLIBDIR=	$(COMP)\VC98\lib	The path to the lib directory that is used by the MSVC compiler. This value is updated by a workstation installation that is based on the user's deployment preferences.
TPLNAME=	EXEFORM2	
ERRNAME=	CGERR	
TARGET=	Debug (default) Release	Used by the code generator and global build program to determine the type of build. Customer should only build under release, as conflicts with the release build of the tools occur if they build under debug.
INCLUDES=	$(COMP)\VC98\include;$ (SYSTEM)\include; $(SYSTEM)\cg; $(APP)\include; $(SYSTEM)\includev	The path to the include (header files) directory that is used by the MSVC compiler. This value is updated by a workstation installation, based on the user's deployment preferences.
LIBS=	$(COMP)\VC98\lib; $ (SYSTEM)\lib32; $(APP)\lib32; $(SYSTEM)\libv32	The path to the library directory that is used by the MSVC compiler and JD Edwards EnterpriseOne Foundation. This value is updated by a workstation installation, based on the user's deployment preferences.
MAKEDIR=	$(COMP)\VC98\bin; $(COMP) \Common\MSDev98\Bin	The path to the make directory that is used by the MSVC compiler. This value is updated by a workstation installation, based on the user's deployment preferences.
USER=	User name	The user ID of the person who performed the workstation installation.

[JDEMAIL]

The JDEMAIL settings are:

Setting	Typical Value	Purpose
ClientType=	Windows HTML	Defines whether the application shortcut that is attached to an external email message contains a Windows application shortcut or a URL for an HTML application shortcut. The default value is Windows.
mailProfile=	Default Profile	The name of the profile to be used for external mail systems that are accessed through JD Edwards EnterpriseOne Work Center. Examples of external mail servers include Microsoft Exchange Server and Lotus Domino Mail Server.
mailServer=	owsmtp.jdedwards.com	The domain name of the SMTP server to be accessed for sending server mail messages.

[JDENET]

The JDENET section settings are:

Setting	Typical Value	Purpose
serviceNameListen=	6005	Specifies the communications service port on the TCP/IP network. The software uses this port address to listen for requests on the network.
serviceNameConnect=	6005	Specifies the communications service port on the TCP/IP network. The software uses this port address to connect to the network. OCM determines on which server a business function runs. If you run multiple instances of JD Edwards EnterpriseOne on the same server, each instance runs on a different port. The serviceNameConnect parameter value determines the JD Edwards EnterpriseOne instance that handles the business function request.
maxLenInlineData=	1024	For internal use only.
maxLenFixedData=	4096	For internal use only.
maxFixedDataPackets=	1024	For internal use only.
netTrace=	0	Turns netTrace on or off. The default setting of 0 means that netTrace is off. A setting of 1 - 4 enables JDENET debug to log messages. You can increase the level of detail by increasing this number, and you can use these log messages for debugging.
kernelDelay=	0	For internal use only.

[JDENET_KERNEL_DEFx]

The JDENET_KERNEL_DEFx settings are:

Setting	Typical Value	Purpose
bOneUserOnly=	0	Parameter value of 1 allows the client workstation to get its own kernel process on the server. For the setting to work, a corresponding parameter, bAllowOneUserOnly, with a value of 1, must be added to the [JDE_KERNEL_DEFx] section of the server jde.ini file. Specify the kernel process that the user will have on the server by adding the number of the kernel definition section: [JDENET_KERNEL_DEF6] bOneUserOnly=1 This setting allows a client workstation to have its own CallObject kernel process on the server.

[LOCK MANAGER]

The LOCK MANAGER settings enable transaction processing.

Note. Enable transaction processing on the server before you enable it on the workstation. If you try to set up the workstation jde.ini file before you set up the server jde.ini, you could be requesting a service on the server that is not yet available, which generates an error.

The LOCK MANAGER settings are:

Setting	Typical Value	Purpose
Server=	Server name	This setting indicates the lock manager server to be used to process records. For example, a server name might be `intelnta`. The value for this setting is the name of the server that is acting as the lock manager.
RequestedService=	NONE	This setting indicates the type of service that the client requests from the server. Valid values are: • TS Time stamp service. • NONE No service is requested.

[MAILMERGE]

The MAILMERGE setting is:

Setting	Purpose
FileLocation=	The location on the workstation of the mailmerge file.

[NETWORK QUEUE SETTINGS]

The settings in this section contain the name of the queue that is used when running batch jobs on the server. The settings also show the workstation's UBE priority, and whether to hold the jobs in a spool file or immediately send them to a printer:

Setting	Typical Value	Purpose
UBEQueue=	QBATCH	The batch name that the client submits for the UBE or package installation to the server.
UBEPriority=	5	The priority that is set when the UBE is submitted. For workstations, valid values are 1 to 5, where 1 is the highest priority setting. The priority setting is relative to other UBE jobs that are submitted by the software.
PrintImmediate=	FALSE (default) TRUE	Valid values (which are case sensitive) are: • FALSE (default) Holds the report output in a queue until you select Print from the Row menu on the Submit Job - Submitted Job Search form on the Microsoft Windows client. • TRUE Sends the report output to the printer immediately after processing. JD Edwards EnterpriseOne servers hold the UBE spool files submitted from a workstation unless the jde.ini file on the workstation has the PrintImmediate=TRUE setting. On the iSeries, the spool file is created with the HOLD(*YES) attribute as a default. If the setting PrintImmediate=TRUE is set in the jde.ini file on the workstation, upon submission of the UBE to the JD Edwards EnterpriseOne server, the spool file is released once it is placed on the appropriate outqueue and closed.
SaveOutput=	TRUE (default) FALSE	A setting that lists whether the user wants to save the log files that are generated by the UBE.
InstallSpecs=	Y	A setting that lists whether the user wants to install specifications when submitting UBEs.
JDENETTimeout=	60	The timeout value, listed in seconds, for clients to attempt to connect to the server.

[OBJECT LIBRARIAN]

The OBJECT LIBRARIAN section settings are:

Setting	Value	Purpose
OLTLogMode=	YES (default) NO APPEND	This setting specifies if and how the Object Management Workbench Transaction log (OLT.log) is generated. Valid values are: • YES. The OLT.log is generated for each transaction. If the log exists before a transaction, its contents are overwritten. • NO. No OLT.log is generated during Object Management Workbench object transactions. • APPEND. The information for a transaction is appended to the OLT.log. When the size of the log reaches the maximum size allowed (2 MB), the user is prompted to rename the existing file. If the user chooses not to rename it, the existing contents of the log are overwritten by the information that is generated by the new transaction.
OLTLogContents=	GENERAL (default) DETAIL	Specifies if detail information about specification records will be generated in the OLT.log. Valid values are: • GENERAL. No detail information about specification records will be generated. • DETAIL. Detail information will be generated.

[OFFLINE DB SYSTEM SETTINGS]

The settings in this section are used only for running JD Edwards EnterpriseOne in detached mode. If you have not installed the workstation with the detached mode option, this section does not appear in the workstation jde.ini file. The following settings are the same as in the section [DB SYSTEM SETTINGS] listed previously in this chapter, although the values are different:

Setting	Value	Purpose
Version=	43	A version number to prevent mismatch of the jde.ini file with the running version of JD Edwards EnterpriseOne.
Default User=	JDE	The user account name for the database bootstrap tables.

Setting	Value	Purpose
Default Env=	DEMO811	The default environment on the workstation or the enterprise server.
Default PathCode=	APPL_PGF	The name of a subdirectory under \811 that the software uses to find specifications to display sign-in information before an environment is selected.
Base Datasource=	System Local	Data source representing the database from which logon information is retrieved.
Object Owner=		The owner of the system database tables.
Server=		The server on which the database resides.
Database=	System Local	The name of the Oracle connect string or the ODBC datasource name for iSeries.
Load Library=		The PSFT driver that is used to access the database that stores the system tables. This value is set dynamically by JD Edwards EnterpriseOne runtime.
Decimal Shift=	N	A flag to indicate if decimal shifting is used for numeric data.
Julian Dates=	N	A flag to indicate if dates are stored in Julian or database-specific format.
Use Owner=	N	A flag to indicate that tables names are qualified by owner.
Secured=	N	Indicates whether this database is secured, requiring a user and password login.
Type=	A (default) O I L W M S	A single character denoting the type of database holding the system tables. These characters can be O (Oracle), A (MS Access), I (Client Access, iSeries), L (SQL Server OLEDB), W (DB2 UDB for Windows/Unix), S (SQL Server), M MSDE/OLEDB OR N (MSDE/ODBC).
DatabaseName2=		ODBC name for the SQL Server database or iSeries library database name.

Setting	Value	Purpose
DatabaseInstance=		Name of the SQL Server database instance if using multiple instances. Leave this setting blank if using a single instance.
ServerPort=		The port number of the SQL Server database port.
UnicodeFlag=	N (default) Y	Indicates whether Unicode is used on the datasource. Set this to Y if using Unicode.
LOBFlag=	Y	For Oracle and iSeries. Indicates that LOBs are used in the datasource instead of BLOBs. This value should always be Y.
LibraryList=		iSeries only. Database server that stores the system tables.
Default Pwd=		The default password.

[SECURITY]

The SECURITY settings are:

Setting	Typical Value	Purpose
SecurityServer=	Server name	
RowSecurity=	DEFAULT	
DataSource=	ORACLE PVC	
DefaultEnvironment=	Environment name	This setting defines a valid environment in which the path code defines F98OWSEC.
UnifiedLogon=	0 (default) 1	This setting specifies whether the unified logon feature is on or off. When off, JD Edwards EnterpriseOne uses the standard logon functionality. Enter 0 (or leave blank) to set unified logon to off, or 1 to set it to on.
UnifiedLogonServer=	server name	This setting specifies where the unified logon server resides. If no server is present, the default is the JD Edwards EnterpriseOne security server.
ShowUnifiedLogon=	0 1 (default)	This setting determines whether the JD Edwards EnterpriseOne environment selection form appears when the unified logon feature is used. Valid values are: • 0: The environment selection form is not displayed. • 1: The environment selection form is displayed.

[SVR]

The settings in this section contain environment and subdirectory information:

Setting	Typical Value	Purpose
EnvType=	1	Used by JDEKRNL.
EnvironmentName=		
SpecPath=	spec	This line and all of the following lines in this section specify the path names that enable other JD Edwards EnterpriseOne source programs to locate files. For instance, if spec is changed to specifications, changing SpecPath would allow this change immediately. This value is not updated by any program or process. The only reason to change this value is aesthetic. This line is the subdirectory under the path code that is used to store the set of specification files on the workstation.
SourcePath=	source	On the client workstation, the subdirectory under the path code that is used to store the business function source files.
ObjectPath=	obj	On the client workstation, the subdirectory under the path code that is used to store the business function object files.
HeaderPath=	include	On the client workstation, the subdirectory under the path code and system directory that is used to store the business function and system header files.
HeaderVPath=	includev	On the client workstation, the subdirectory under the system directory that is used to store the foundation code header files.
BinPath=	bin32	On the client workstation, the subdirectory under the path code and system directory that is used to store the set of business functions, and application and foundation code dlls.
LibPath=	lib32	On the client workstation, the subdirectory under the path code and system directory that is used to store the business function and system lib files.
LibVPath=	libv32	On the client workstation, the subdirectory under the path code and system directory that is used to store the third-party libraries.
MakePath=	make	On the client workstation, the subdirectory under the path code that is used to store the set of business function make files. This value is not updated by any program or process. We recommend that you do not change the name of this directory.

Setting	Typical Value	Purpose
WorkPath=	work	On the client workstation, the subdirectory under the path code that is used to store the set of application temp files that are created during a build. This value is not updated by any program or process. We recommend that you not change the name of this directory.
CodeGeneratorPath=	cg	On the client workstation, the subdirectory under the system directory that is used to store the templates for interactive application form types. These templates are used at runtime and are created during a build of applications.
ResourcePath=	res	On the client workstation, the subdirectory under the path code that is used to store the set of bitmaps.
IconPath=	res\icons	On the client workstation, the subdirectory under the path code that is used to store the set of icons.
FontPath=	res\font	On the client workstation, the subdirectory under the path code that is used to store the set of fonts.
HelpPath=	helps	The path to the location that stores the client-accessible set of help files, if any. This path can point to a server and is specified in User Profiles.
TreeBmpPath=	res\treebmps	On the client workstation, the subdirectory under the path code that is used to store the tree bit map files.
ModelPath=	models	On the client workstation, the subdirectory under the path code that is used to store the models files.
LocalePath=	locale	The base directory for the National Language Support (NLS) conversion tables.
Iconvpath=	locale\Iconv	The directory for the NLS conversion map.

[UBE]

The settings in this section determine whether the jdedebug.log is on or off. This setting also determines the level of debugging:

Setting	Typical Value	Purpose
UBEException	0 1 (default)	Disable (0) or enable (1) exception handling. 0 allows the system to exit gracefully on an error. 1 brings up a debug log.
UBEChkMem	0 1 (default)	Disable (0) or enable (1) calls to jdeCheckMemory, which allows memory traces on business functions.

Setting	Typical Value	Purpose
UBESubsystemLimit	3	Used to specify the number of subsystem jobs per report version.
UBESaveLogFile	0 (default) 1	Delete (0) or save (1) UBE log file (delete only works when DebugLevel=0).
UBEDBOutputLocation		Used to define the location for database output or mapping.
UBETabOpt=	1 0	Not delivered in the default jde.ini file. The system automatically performs row data selection optimization. However, use this setting if you want to disable row data selection optimization. The values are enabled (1) or disabled (0). If you want to re-enable optimization, you can change the value back to 1 or simply delete the UBETabOpt setting.
UBEDebugLevel=	0 (default) 1- 6	Used to specify what level of debugging information will be provided when using UBE debug logging. The highest level of logging information is 6. Valid values are: • 0: Error messages only • 1: Informative messages • 2: Section-level messages • 3: Object-level messages • 4: Event rules messages • 5: SQL statements • 6: UBE function messages
UBEShowPDFLink	0 (default) 1	Type 1 to show a box around PDF links.
UBEPrintDataItems	0 (default) 1	Used to specify whether to print the associated data item description in the .pdf file as metadata for third-party vendors. Valid values are: • 0: No, do not print. • 1: Yes, do print.
UBESSDebug	0 (default) 1	Disable (0) or enable (1) printing the subsystem key.
UBEVCDebug	0 (default) 1	Auto-attach VC when starting UBE in process (NT only).

Setting	Typical Value	Purpose
UBEThread	0 1 (default)	Run UBE as a process (0) or a thread (1).
WebServer	0 (default) 1	This setting specifies whether the system enables the UBE feature from the web server and identifies the JD Edwards EnterpriseOne kernel as a web kernel to meet the special needs of the web. If you leave this value blank, the calls from the business functions or the error message handling from the kernel do not work properly. Valid values are: • 0: Disabled • 1: Enabled

[WORKFLOW]

The WORKFLOW settings are:

Setting	Value	Purpose
Asynchronous Workflow=	FALSE	Used to activate or deactivate asynchronous workflow. The default value is FALSE.

iSeries Server jde.ini Settings

This section describes the settings found in the JD Edwards EnterpriseOne iSeries server INI. Information is organized by section—for example, [DEBUG]. Sections are alphabetized, but settings within sections are listed in the order in which they are found in the software.

[AS400]

The AS400 settings are:

Setting	Value	Purpose
CRTMOD=	CRTMOD MODULE(%s /%s) SRCFILE(%s/%s) SRCMBR(%s) OUTPUT(*PRINT) DBGVIEW(*NONE) OPTIMIZE(40)	The string used by the package install to compile business functions. Note that CRTMOD and CRTMOD2 are concatenated and used by the software to compile business functions.
CRTMOD2=	DEFINE(JDENV_ AS400MUTEX PRODUCTION_ VERSION NO_SIGNALS) TGTRLS(V4R3M0)	The concatenated string that is used by the package install for declaring additional definitions for compiling business functions.

Setting	Value	Purpose
CRTSRVPGM=	CRTSRVPGM SRVPGM(%s/%s) MODULE(%s/*ALL) BNDSRVPGM(JDELIB JDEKRNL OWVER) EXPORT(*ALL) OPTION(*DUPPROC *DUPVAR *UNRSLVREF) ALWLIBUPD(*YES) TGTRLS(V4R3M0)	The string that is used by the package install for binding business function modules to create the JD Edwards EnterpriseOne service programs (*SRVPGM).
CRTDBPGM1=	CRTPGM PGM(%s/%s) MODULE(DBDRVAG DBDRV_AC DBDRV_CC DBDRV_CN	The concatenation of CRTDBPGM* settings is used to create the database programs JDB_*. These database programs are automatically created by JD Edwards EnterpriseOne at startup. The SENTINEL job creates them at startup time, and then monitors and creates additional programs as needed during runtime. The status of the programs and their usage are maintained in the user space JDEPGMCTL in the CONTROL library.
CRTDBPGM2=	DBDRV_CH DBDRV_CP DBDRV_RQ DBDRVSQL DBMONCTL DBDRVDLI	See purpose for CRTDBPGM1.
CRTDBPGM3=	DBSQL DBSQL_A DBSQL_D DBSQL_I DBSQL_M DBSQL_S DBSQL_U DBSQL_X	See purpose for CRTDBPGM1.
CRTDBPGM4=	BNDSRVPGM(JDEKRNL JDELIB JDEIPC) ACTGRP(%s) OPTION(*DUPPROC	See purpose for CRTDBPGM1.
CRTDBPGM5=	*DUPVAR) ALWLIBUPD (*YES) AUT(*ALL) TGTRLS(V4R3M0)	See purpose for CRTDBPGM1.
PrintUBEJoblog=	FALSE (default) TRUE	If true, indicates that the software always writes the iSeries JOBLOG for the batch application (UBE) to a spool file.
PrintUBEJoblogOn Error=	FALSE (default) TRUE	If true, indicates that the software writes the iSeries JOBLOG for the batch application (UBE) to a spool file if an error occurs—for example, if a UBE fails.

[BSFN BUILD]

The BSFN BUILD settings are:

Setting	Value	Purpose
Build Area=	/psft811/packages	The location on the server where the package will be built.
Optimization Flags=	(40)	Machine dependent. These compile flags are used when building business functions in Release mode. You should not change these flags.
DebugFlags=	*ALL	Machine dependent. These compile flags are used when building business functions in Debug mode. You should not change these flags.
InliningFlags=	Y (default, yes) N (no)	Valid values are: • Yes: Turns on inlining on the iSeries. • No: Turns inlining off. This entry is blank for non-iSeries servers.
DefineFlags=	JDENV_AS$))MUTEX PRODUCTION_VERSION JDBDB2400 AS400V3R6	
CompilerFlags=	*EXPMAC *NOSHOWINC	This setting determines whether to compile listings when building a server package. Valid values are: • *PRINT: Listings are compiled. • *NONE: Listings are not compiled.
CompileOutput=	*PRINT blank	Machine dependent. Valid compiler flags. The spill flag sets the stack space when business functions are compiled. The default value of 1024 is adequate to compile the delivered business functions. While values other than 1024 might be valid on various host platforms, this value is the only one validated by JD Edwards EnterpriseOne.
OSReleaseLevel=	V4R3M0	The release level to which you are compiling. You should not change this setting.
LinkFlags=	*DUPROC *DUPVAR * UNRSLVREF	Machine dependent. These flags are used when linking business functions. You should not change these flags.
LinkLibraries=	JDELIB JDEKRNL JDENET JDEIPC OWVER	Libraries to which business functions are linked (Windows and iSeries servers only).

Setting	Value	Purpose
SimultaneousBuilds=	0 (default) any integer	Indicates the number of DLLs that can be built at a time. 0 (zero) means that all DLLs are built simultaneously.
QName=	as400 batch jobq name	The job queue name to which all package builds will be submitted. If left blank, QName uses the default JOBQ as specified in the user profile that is doing the submitting.

[DB SYSTEM SETTINGS]

The settings in this section contain information about the default environment and path code:

Setting	Value	Purpose
Version=	43	A version number to prevent mismatch of the jde.ini file with the running version of JD Edwards EnterpriseOne.
Default User=	PSFT	The user account name for the database bootstrap tables.
Default Pwd=	PSFT	The user account password for the database bootstrap tables.
Default Env=	811APP	The default data source on the workstation or the enterprise server.
Default PathCode=	811APP	The subdirectory under \$PKG under which the business function code is stored.
Base Datasource=	DB2	The data source representing the database from which logon information is retrieved.
Object Owner=		The owner of the system database tables.
Server=	server name	The server on which the database resides.
Database=	database name	The name of the database where the system tables reside.
Load Library=	DBDR (default)	The PSFT driver that is used to access the database that stores the system tables. This driver depends on the database to be used and the type of system running JD Edwards EnterpriseOne.
Decimal Shift=	Y (default) N	A flag to indicate if decimal shifting is used for numeric data.
Julian Dates=	Y (default) N	A flag to indicate if dates are stored in Julian or database-specific format.

Setting	Value	Purpose
Use Owner=	Y N (default)	A flag to indicate that tables names are to be qualified by owner.
Secured=	Y (default) N	Indicates whether this is a secured database requiring a user and password login.
Type=	I	A single character denoting the type of database holding the system tables. These characters can be O (Oracle), A (MS Access), I (Client Access, iSeries), L (SQL Server OLEDB), W (DB2 UDB for Windows/Unix), S (SQL Server), M MSDE/OLEDB OR N (MSDE/ODBC).
Library=	database library	iSeries only. The database library that stores the system tables.
DatabaseName2=		ODBC name for the SQL Server database or iSeries library database name.
DatabaseInstance=		Name of the SQL Server database instance if using multiple instances. Leave this setting blank if using a single instance.
ServerPort=		The port number of the SQL Server database port.
UnicodeFlag=	N (default) Y	Indicates whether Unicode is used on the datasource. Set this to Y if using Unicode.
LOBFlag=	Y	For Oracle and iSeries. Indicates that LOBs are used in the datasource instead of BLOBs. This value should always be Y.
DatabaseProgamMax=	-1 (default)	iSeries only. The maximum number of database connection programs to allow. The value -1 means no limit.
DatabaseProgramInitial=	10 (default)	iSeries only. The number of database connection programs to start initially when JD Edwards EnterpriseOne is started.
DatabaseProgramThreshold=	3 (default)	iSeries only. The threshold for starting new database connection programs. If the number of database connection programs not in use drops below this limit, start new ones.
DatabaseProgramAdditional=	10 (default)	iSeries only. The number of new database connection programs to start when the threshold number is reached.

Setting	Value	Purpose
DatabaseProgramCheckIntervalSeconds=	0 (default)	iSeries only. The length, in seconds, before the software idles after the database connection programs are created.
Default Journal=	OW_JRNL	iSeries only. The name of the default journal. Journaling is required on the iSeries for rollback recovery. The two components to journaling are: • The journal • The journal receiver Both before and after images of a database transaction can be recorded by journaling. This value can be set to any character string that is 10 characters or fewer.
Default Journal LIBRARY=	journal library	iSeries only. The library name where the journal is stored. This name can be set to any valid library name. The library name changes for each release.
Default Journal Receiver	OW_JRNL000	iSeries only. The name of the journal receiver. This can be set to any character string that is 10 characters or fewer.
Default Journal Receiver LIBRARY=	journal library	iSeries only. The library name where the journal receiver is stored. This can be set to any valid library name. The library name changes for each release.
Size of Journal Receiver=	6000	iSeries only. This setting specifies a storage space threshold value (in KB) for the journal receiver. If the threshold value is exceeded during journaling, one of the following occurs: • The message CPF7099 is sent to the journal message queue if the journal has the MBGRCV(*USER) attribute. • The system attempts to create and attach a new receiver if the journal has the MBGRCV(*SYSTEM) attribute. When the old receiver is detached, the message CPF7020 is sent to the journal message queue. If the attempt fails due to lock conflicts, the system sends the message CPI70E5 and then tries again every ten minutes until the change journal operation is successful. When the system cannot determine if the journal has the MBGRCV(*SYSTEM) attribute, or if the attempt to create and attach a new journal receiver fails because of something other than a lock conflict, the message CPI70E3 is sent.

[DEBUG]

The settings in this section determine the location of the jde.log and jdedebug.log. The settings are also used to turn the jdedebug.log on and off.

Setting	Typical Value	Purpose
Output=	FILE	Controls the status of the jdedebug log file. Valid values are: • NONE: No trace information is written to jdedebug.log. • FILE: Database and runtime trace information is written to the file that is specified by the DebugFile= parameter in the [DEBUG] section.
Trace=	TRUE	Writes additional trace information to the log files to aid in debugging.
DebugFile=	psft811/ jdedebug	Location of the jdedebug log. The default value is jdedebug. No processes update this value. The names of the resulting files are path/jdedebug_######.log, where ###### represents the iSeries job number that is associated with the job that created the file. **Note.** The software does not create the path to these files. The path must exist prior to the logging process. The path resides in the Integrated File System (IFS) on the iSeries. You can use the iSeries WRKLNK command to see a list of directories and files and navigate between the IFS directories. JD Edwards EnterpriseOne contains a command called DSPSTMF that enables you to view these log files. In addition, you can set up Client Access to more easily view some of the smaller log files. See *JD Edwards EnterpriseOne Tools 8.96 Server and Workstation Administration Guide*, "Administering the iSeries Server," Using iSeries Integrated File System Logging Support.
JobFile=	psft811/jde.log	Location of the jde log. By default, this value is set to the jde.log. No processes update this value. Examine the log files jde.log, and jdedebug for information that can be used to assist in problem analysis and resolution. The names of the resulting files is path/jde_######.log where ###### is the iSeries job number that is associated with the job that created the file. See *JD Edwards EnterpriseOne Tools 8.96 Server and Workstation Administration Guide*, "Administering the iSeries Server," Using iSeries Integrated File System Logging Support.
JDETSFile=	/psft811/JDETS.LOG	Specifies the location of the lock manager trace file on the iSeries.

Setting	Typical Value	Purpose
ClientLog=	1 (default) 0	Valid values are: • 1: enables servicing CALLOBJ server trace to workstation. • 0: disables servicing CALLOBJ server trace to workstation.
LogErrors=	1 (default) 0	The action for error messages. Valid values are: • 0 or FALSE: Indicates that no error messages will be written to JDE.LOG. • 1 or TRUE: Indicates that error messages will be written to JDE.LOG.
CMTrace	0 1	This setting disables (0) or enables (1) cache logging. This setting works in conjunction with the CMTraceFilter. In addition, if output logging is disabled and cache level logging is enabled, the system will continue to format and prepare cache level logging information, but will not display it in the log file. This can cause a degradation in system performance.
CMTraceFilter=	ALL	This setting determines if cache logging messages will appear in the JDEDEBUG.LOG file.
ThreadTraceLevel=	0–3	This setting determines the thread logging level for the JDEDEBUG.LOG file. The higher the value, the more information the system logs. Valid values are: • 0 (Default) Disables thread trace logging. • 1 • 2 • 3 **Note.** Use level 3 logging with caution since excess logging can result in severe degradation of system performance. In addition, if output logging is disabled and thread level logging is enabled, the system will continue to format and prepare thread level logging information, but will not display it in the log file. This can cause a degradation in system performance.

Setting	Typical Value	Purpose
KeepLogs=	1	Valid values are: • 1: Indicates that the logs will be saved after printing. • 0: Indicates that the logs will not be saved.
RunBatchDelay=	0	Specifies the time that runbatch waits upon startup, in seconds. This setting enables developers to start debugging the job or process.
TAMTraceLevel=	0 (default)	Specifies the level of TAM tracing, where 0 is off and 9 provides the greatest amount of tracing detail.

[ENTERPRISE TIMEZONE ADJUSTMENT]

The setting in this section contains information for properly adjusting the PeopleSoft Enterprise Portal time for single sign-on.

Setting	Purpose
EntNode=	Enables you to enter the difference, in minutes, between Greenwich Mean Time (GMT) and PeopleSoft Enterprise Portal Node time when setting up single sign-on between PeopleSoft Enterprise Portal and JD Edwards EnterpriseOne. You should change this setting whenever daylight savings time changes to reflect the difference between GMT time and the PeopleSoft Enterprise Portal Node time.

[INSTALL]

The settings in this section contain directory paths and general installation information:

Setting	Typical Value	Purpose
DefaultSystem=	E811SYS	The name of the JD Edwards EnterpriseOne System library. This value must be unique for each JD Edwards EnterpriseOne instance.
ClientPath=	811APP	The name of a valid path code on the deployment server that contains the workstation installation program and other files that are used during deployment.
811=		Should be left blank on the iSeries.
LocalCodeSet=	US_EBCDIC	A setting that is used to determine alternate language usage. See the appropriate JD Edwards EnterpriseOne Tools 8.12 Upgrade Guide for other language values.

Setting	Typical Value	Purpose
WebAdmin=	1	This setting specifies whether the system generates all of the Java objects for the default user. This setting includes overriding Java objects that were previously generated. If you leave this value blank, the system generates all the Java objects for the current user.
EnvCreation=	1 (default) to 5	This setting determines the number of environments that can be processed (loaded) at the same time.

[INTEROPERABILITY]

The INTEROPERABILITY settings are:

Setting	Typical Value	Purpose
LEVEL	EVENTS,DATA	The system writes specified interoperability event data to the debug log file. You can specify one or more of the allowable logging settings. The interoperability documentation contains a list of possible values and their definitions. See *JD Edwards EnterpriseOne Tools 8.96 Interoperability Guide*, "Getting Started with JD Edwards EnterpriseOne Tools Interoperability".
RegisteredEvents=	RTSOOUT	Names of EventTypes. An event is JD Edwards EnterpriseOne business transaction running on an enterprise server. To enable real-time generation of events, you must register each event that you want to generate in realtime.
FilteredEvents=	*ALL	The value of this parameter defines the events that you want to create in realtime. A value of *ALL generates all registered events. *NONE disables event generation. You can also enter a subset of registered events.
SaveEVNDoc	0 1	This setting determines the logging level for events. You can disable (0) or enable (1) event logging. If this setting is enabled, you must make sure that logging is enabled as well. See the Output and LogErrors settings in the [DEBUG] section to enable logging. **Note.** If output logging is disabled and event level logging is enabled, the system will continue to format and prepare event level logging information, but will not display it in the log file. This can cause a degradation in system performance.

[JDEIPC]

The JDEIPC settings are:

Setting	Typical Value	Purpose
maxNumberOfResources=	1000	The total number of IPC resources that are available.
startIPCKeyValue	2101	On NT, this value is used to uniquely name the IPC Shared memory. On all other systems, this value is the value of the IPC ID, which JDEIPC used for its shared memory. This value, plus the maxNumberOfResources, defines the range of IPC IDs that JD Edwards EnterpriseOne will use on the system. System administrators should ensure that this range of IDs is not used by any other software. Although JDEIPC will not use an existing ID in its range, this situation might not be true of other software.
avgResourceNameLength	15	Oracle internal. Increase this value if you get an IPC error String table full.
maxMsgqEntries=	1024	
mazMsgqBytes=	65536	
ipcTrace=	0	Controls the level of interprocess communications (IPC) messages written to the jdedebug.log. Valid values are: • 0 (default): Writes no messages to the debug log. • 1: Writes only general trace messages. • 2: Writes IPC handle state trace messages. • 3: Writes both general and IPC handle state trace messages.

[JDEITDRV]

The JDEITDRV settings are:

Setting	Typical Value	Purpose
DrvCount=	3	The number of event drivers that is used for processing messages from event generators, either Z file or real-time.
Drv1=	Z:ZDRV	The directory location of the Z file event driver.
Drv2=	RT:RTDRV	The directory location of the real-time event driver.
Drv3=	JDENET:JDETRDRV	The directory location of the JDENET driver.

[JDEMAIL]

The JDEMAIL settings are:

Setting	Typical Value	Purpose
mailServer=	owsmtp.jdedwards.com	The domain name of the SMTP server that is accessed for sending server mail messages.

[JDENET]

The JDENET settings are:

Setting	Typical Value	Purpose
serviceNameListen=	jde_server	Specifies the communications service port on the TCP/IP network. JD Edwards EnterpriseOne uses this port address to listen for requests on the network.
serviceNameConnect=	jde_server	Specifies the communications service port on the TCP/IP network. JD Edwards EnterpriseOne uses this port address to connect to the network.

Setting	Typical Value	Purpose
maxNetProcesses=	Depends on the maximum number of concurrent JD Edwards EnterpriseOne users that the system is expected to handle, as well as the processing and memory power of the server. If you only need one process, then maxNetProcesses=1. However, if you need two or more processes, note that the first process is used exclusively to handle new connections and distribute them evenly among the other processes. In this case, you must add an extra process to act as a broker. To determine the number of processes that you need, you must first determine the maximum number of connections that you need and the number of connections a single process can handle. The formula for the number of connections allowed for each process is: (maxNetConnections/ (maxNetProcesses - 1)) + 2 Each process requires two extra connections for listening (one for TCP and another for UDP), and an extra process is dedicated to handling incoming connections.	Defines the maximum number of JDENET_N processes that can be running. You can increase the value for a server that is expecting heavy JDENET message flow.
maxNetConnections=	Depends on the maximum number of concurrent JD Edwards EnterpriseOne users that the system is expected to handle, as well as the processing and memory power of the server.	The total number of connections that all JDENET_N processes can handle. This value is platform-specific. You can increase the value for a server that is expecting to handle larger number of workstations at the same time.
netShutdownInterval=	15	

Setting	Typical Value	Purpose
maxKernelProcesses=	Depends on several factors: Total of the individual kernel type maximums. The value should be at least that large, but it can be increased as needed. The number of one-user-only kernels you want to allow. Any number above the individual kernel maximum total will be allocated to one-user-only kernels. The room that you want to allow for dynamic increase of kernel processes from the Server Administration Workbench.	The maximum number of JDENET_K processes that can be running. The value should be greater than all of the values added together in maxNumberOfProcesses for all the dedicated servers.
maxKernelRanges=	30	The number of dedicated servers and types.
kernelDelay=	0	For internal use only.
maxLenInlineData=	1024	For internal use only.
maxLenFixedData=	4096	For internal use only.
maxFixedDataPackets=	1024	For internal use only.
netTrace=	0	For internal use only.
krnlCoreDump=	0	For internal use only.
newProcessThreshold Connects=	0	
MaxIPCQueueMsgs	12	For internal use only.
InternalQueueTimeout	30	For internal use only.

[JDENET_KERNEL_DEFx]

This section defines internal dedicated server processes for JDENET. The sections are numbered JDENET_KERNEL_DEF1 to JDENET_KERNEL_DEF30. The settings in these sections should not be changed except where noted:

Setting	Value	Purpose
bAllowOneUserOnly=	1	Parameter value of 1 (default) means that one-user-only kernel processes are allowed on client workstations. Add the setting only for CallObject kernel processes: [JDENET_KERNEL_DEF6] bAllowOneUserOnly=1 Setting must be added with adding a [JDENET_KERNEL_DEFx] bOneUserOnly=1 section to the client workstation jde.ini file.

Setting	Value	Purpose
krnlName	DEF1: JDENET RESERVED KERNEL DEF2: UBE KERNEL DEF3: REPLICATION KERNEL DEF4: SECURITY KERNEL DEF5: LOCK MANAGER KERNEL DEF6: CALL OBJECT KERNEL DEF7: JDBNET KERNEL DEF8: PACKAGE INSTALL KERNEL DEF9: SAW KERNEL DEF10: SCHEDULER KERNEL DEF11: PACKAGE BUILD KERNEL DEF12: UBE SUBSYSTEM KERNEL DEF 13: WORKFLOW KERNEL DEF 16: XML LIST KERNEL DEF 19: EVENT NOTIFICATION KERNEL DEF 20: INTEROPERABILITY EVENT OBSERVER KERNEL DEF 22: XML DISPATCH KERNEL DEF 23: XTX KERNEL DEF 24: XML SERVICE KERNEL DEF30: METADATA KERNEL	DEF1: Used for internal purposes and testing. DEF2: Processes JD Edwards EnterpriseOne batch process requests. DEF3: Processes data replication requests. DEF4: Processes security server requests. DEF5: Processes transaction manager and lock manager requests. DEF6: Processes requests for remote master business functions (MBF). DEF7: Processes JDBNet server-to-server requests. DEF8: Processes package installation request. DEF9: Processes SAW application requests. DEF10: Processes Scheduler application requests. DEF11: Processes package build requests. DEF12: Processes UBE subsystem requests. DEF13: Processes workflow requests. DEF16: Processes and returns request for data in XML document format. DEF19: Processes real-time events and XML documents generated by the Interoperability Event Observer, as well as Z file events. Publishes all JD Edwards EnterpriseOne events to subscribers. DEF20: Processes information from business functions calling real-time APIs and uses that information to create an XML or a Z file that is publishable to subscribers by the Event Notification Kernel. DEF22: Kernel for routing XML messages to their proper kernel. DEF23: Kernel for transforming XML messages from one type to another. DEF24: Processes inbound XAPI messages. DEF30: Processes XML spec access requests.

Setting	Value	Purpose
dispatchDLLName=	DEF1: JDENET DEF2: JDEKRNL DEF3: JDEKRNL DEF4: JDEKRNL DEF5: JDEKRNL DEF6: JDEKRNL DEF7: JDEKRNL DEF8: JDEKRNL DEF9: JDESAW DEF10: JDEKRNL DEF11: JDEKRNL DEF12: JDEKERNL DEF13: JDEKRNL DEF16: XMLLIST DEF19: JDEIE DEF20: JDEIEO DEF22: XMLDISPATCH DEF23: XJSKERNEL DEF24: XMLSERVICE DEF30: MDSERIALIZ	Identifies the name of the JDENET service program.

Setting	Value	Purpose
dispatchDLLFunction=	DEF1: JDENET_Dispatch Message DEF2: JDEK_ DispatchUBEMessage DEF3: DispatchRepMessage DEF4: JDEK_ DispatchSecurity DEF5: TM_ DispatchTransactionManager DEF6: JDEK_ DispatchCallObject Message DEF7: JDEK_ DispatchJDBNETMessage DEF8: JDEK_ DispatchPkgInstallMessage DEF9: JDEK_ DispatchSAWMessage DEF10: JDEK_ DispatchScheduler DEF11: JDEK_ DispatchPkgBuildMessage DEF12: JDEK_ DispatchUBESBSMessage DEF13: JDEK_WFServerProcess DEF16: JDEK_XMLListDispatch DEF19: JDEK_DispatchITMessage DEF20: JDEK_DispatchIEOMessage DEF22: XMLDispatch DEF23: JDEK_ DISPATCHXTSMessage DEF24: XMLServiceDispatch DEF30: MetadataDispatch	The name of the JDENET function for handling JDENET messages. The dispatchDLLName and dispatchDLLFunction entries are platform-specific.

Setting	Value	Purpose
maxNumberOfProcesses=	Depends on the number of concurrent users and kernel types. For example, CallObject kernels should be configured to start five to ten concurrent users per kernel.	The maximum number of kernel processes that can be run on this server for each kernel type. The user can modify this setting to tune performance. The default value is 1 for all JDENET_KERNEL_DEF sections.
numberOfAutoStartProcesses=	Variable	The number of kernel processes that will automatically start for each kernel type. If this number is 0, then no processes start automatically for that kernel type. This number must be less than the maximum number of processes for that kernel type. The user can modify this setting to tune performance. The default value is 0 for all JDENET_KERNEL_DEF sections. The decision on assigning a value to this parameter should be based on when the user wants the overhead of starting a kernel process to occur: either when JD Edwards EnterpriseOne services start or when the first message for kernel type is received.

[LOCK MANAGER]

The LOCK MANAGER section enables transaction processing and includes the following settings:

Setting	Typical Value	Purpose
Server	server name	This setting specifies the name of the lock manager server to be used to process records—for example, a server name might be `intelna`.
AvailableService=	NONE	This setting indicates the service that the lock manager server is offering. It is also used to indicate whether the lock manager server is on or off. Valid values are: • TS Time stamp service. • NONE No service is available. **Note.** This setting only applies to servers.
RequestedService=	NONE	This setting indicates the type of service that the workstation requests from the server. The service that is currently being provided by servers is time stamping (TS) only. Valid values are: • TS • NONE

[LREngine]

The LREngine settings are:

Setting	Typical Value	Purpose
System=	/E811SYS_X (you must use the integrated file system (IFS)).	The directory location of the List-Retrieval Engine, a database that is used to manage access to XML repository files.

[NETWORK QUEUE SETTINGS]

The settings in this section contain the name of the queue that is used when running batch jobs on the server. The settings also show the workstation's UBE priority, and whether to hold the jobs in a spool file or immediately send them to a printer:

Setting	Typical Value	Purpose
DefaultPrinterOUTQ=	QGPL/ONEWORLD_A	The default printer to which batch applications are routed.
UBEQueue	QBATCH	The batch name that the client submits for the UBE or package installation to the server.
JDENETTimeout=	60	The timeout value, listed in seconds, for clients to attempt to connect to the server. A server can act as a client when it uses JDBNET, submits UBEs to another server, calls a business function on another server, uses a Lock Manager on another server, or when it makes security server requests to another server.

[SAMPLE_EVENT]

The SAMPLE_EVENT settings are:

Setting	Typical Value	Purpose
DS1= DS2= DS3=	D4202150B D4202150C D34A1050C	Defines the data structure for each real-time event registered in the [INTEROPERABILITY] section. Replace [SAMPLE_EVENT] with an event name, such as RTSOOUT, and then enter the values that define the data structure of the event.

[SECURITY]

The SECURITY settings are:

Setting	Typical Value	Purpose
DataSource=		

Setting	Typical Value	Purpose
User=	JDESVR	This value must be a valid JD Edwards EnterpriseOne user ID and database user ID. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.
Password=	JDESVR	This value must be the correct password for the JD Edwards EnterpriseOne user ID specified in the User setting of the [SECURITY] section. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.
DefaultEnvironment=	P810ASD1	Defines a valid environment in which the path code defines the F98OWSEC table.
SecurityServer=	security server name	
ServerPswdFile=	TRUE	The setting of this parameter determines whether JD Edwards EnterpriseOne uses special password handling for batch reports running on the server. Set the value to TRUE to instruct the software to enable special handling of passwords. Set the value to FALSE to disable special handling. When the software runs a batch report on the server, it runs the report by using a string of line commands and parameters that includes the user password. Under some operating systems, the status of a job can be queried and the parameters that were used to start the process can be viewed. As a security measure, you can enable special handling by JD Edwards EnterpriseOne. When enabled, the software does not include the user password in the parameter list for a batch process. Instead, it includes the name of a file that contains the user password. The software instructs the operating system to destroy this file as soon as the batch report reads the password.
History=	0	
SecurityMode=	0 (default) 1 2	This setting controls whether JD Edwards EnterpriseOne uses the standard logon procedure, unified logon, or both. Valid values are: • 0: Accepts only the standard logon. • 1: Accepts only the unified logon. • 2: Accepts both.

Setting	Typical Value	Purpose
AllowedUsers=	group or user names	This setting for the unified logon feature enables you to specify users or groups who are allowed to use JD Edwards EnterpriseOne. If no users or groups are specified, all of the users who have logged on to the proper domains are authenticated by the unified logon server.
secTrace	0–2	This setting determines the security trace logging level for the JDEDEBUG.LOG file. Valid values are: • 0 (Default) Disables security trace logging. • 1 • 2 If output logging is disabled and security trace level logging is enabled, the system will continue to format and prepare security trace level logging information, but will not display it in the log file. This also can cause a degradation in system performance.

[SVR]

The settings in this section contain environment and subdirectory information:

Setting	Typical Value	Purpose
SpecPath=		This line and all of the following lines in this section specify the path names that enable source programs to locate files. This value is not updated by any program or process. The only reason to change this value is aesthetic.
PackedSpecPath=	/811APP	

[TCENGINE]

The TCENGINE settings are:

Setting	Typical Value	Purpose
TraceLevel=	0 (default)	The level of table conversion logging to perform. Valid values are 0-9, where 9 generates the most logging, and 0 generates no logging.

Setting	Typical Value	Purpose
StopAfterRow=	0 (default)	The number of rows to process during table conversion. This setting is useful for debugging. The value 0 indicates that the table conversion processes all rows. Enter a number to indicate the number of rows after which to stop proceeding.
ForceRowByRow=	0 (default) 1	Valid values are: • 0: Allows inserts from selects. • 1: Forces table conversions to convert one row at a time, regardless of whether an insert could be used.

[UBE]

The settings in this section determine whether the jdedebug.log is on or off. This setting also determines the level of debugging:

Setting	Typical Value	Purpose
UBEException	0 1 (default)	Disable (0) or enable (1) exception handling. 0 allows the system to exit gracefully on an error. 1 brings up a debug log.
UBEChkMem	0 1 (default)	Disable (0) or enable (1) calls to jdeCheckMemory, which allows memory traces on business functions.
UBESubsystemLimit	3	Used to specify the number of subsystem jobs per report version.
UBESaveLogFile	0 (default) 1	Delete (0) or save (1) UBE log file (delete only works when DebugLevel=0).
UBEDBOutputLocation		Used to define the location for database output or mapping.
UBETabOpt=	1 0	Not delivered in the default jde.ini file. The system automatically performs row data selection optimization. However, use this setting if you want to disable row data selection optimization. The values are enabled (1) or disabled (0). If you want to re-enable optimization, you can change the value back to 1 or simply delete the UBETabOpt setting.

Setting	Typical Value	Purpose
UBEDebugLevel=	0 (default) 1- 6	Used to specify what level of debugging information will be provided when using UBE debug logging. The highest level of logging information is 6. Valid values are: • 0: Error messages only • 1: Informative messages • 2: Section-level messages • 3: Object-level messages • 4: Event rules messages • 5: SQL statements • 6: UBE function messages
UBEPDFCompression	0 1 (default)	Used to enable (1) or disable (0) the generation of compressed PDF files for viewing on the JD Edwards EnterpriseOne web client.
UBEPDFLinearization	0 (default) 1	Used to enable (1) or disable (0) the linearization of PDF files when generated for the JD Edwards EnterpriseOne web client. When enabled, this feature provides efficient incremental access to the PDF in a network environment.
UBEShowPDFLink	0 (default) 1	Type 1 to show a box around PDF links.
UBEPrintDataItems	0 (default) 1	Used to specify whether to print the associated data item description in the .pdf file as metadata for third-party vendors. Valid values are: • 0: No, do not print. • 1: Yes, do print.
UBESSDebug	0 (default) 1	Disable (0) or enable (1) printing the subsystem key.
UBEVCDebug	0 (default) 1	Auto-attach VC when starting UBE in process (NT only).

Setting	Typical Value	Purpose
UBEThread	0 1 (default)	Run UBE as a process (0) or a thread (1).
WebServer	0 (default) 1	This setting specifies whether the system enables the UBE feature from the web server and identifies the JD Edwards EnterpriseOne kernel as a web kernel to meet the special needs of the web. If you leave this value blank, the calls from the business functions or the error message handling from the kernel do not work properly. Valid values are: • 0: Disabled • 1: Enabled

[WORKFLOW]

The WORKFLOW settings are:

Setting	Value	Purpose
Asynchronous Workflow=	FALSE	Used to activate and deactivate asynchronous workflow. The default value is FALSE.

[WORLD ENVIRONMENT MAP]

The WORLD ENVIRONMENT MAP settings are:

Setting	Value	Purpose
OneWorldEnvironmentName= (for example, APPLJDEDC2)	WorldEnvironment Name (for example, QA811COMP)	The string used by the special business function code to set up JD Edwards World library lists from within JD Edwards EnterpriseOne. The library lists call JD Edwards World software from JD Edwards EnterpriseOne. The functions associated with these settings might not be used by application developers.

[XML Dispatch]

The XML Dispatch setting is:

Setting	Typical Value	Purpose
PollIntervalMillis=	3000	The number of milliseconds that the XML Dispatch kernel sleeps during inactivity when it is waiting on responses from other XML kernels (such as the XML Call Object kernel). The lower this value is, the more CPU cycles the XML Dispatch kernels use when waiting for responses from other XML kernels.

[XTS]

The XTS setting is:

Setting	Typical Value	Purpose
ResponseTimeout=	600	The number of seconds that the XML Dispatch kernel waits for a response from other XML kernels (such as XML Call Object kernel) before giving up on the response.

UNIX/Linux Server jde.ini Settings

This section describes the settings found in the UNIX and Linux server jde.ini file. In these environments, there is an environment variable called JDE_BASE that defines the directory in which the jde.ini is found. Some settings might differ between server platforms. Information is organized by section, such as [DEBUG]. Sections are alphabetized, but settings found within sections are generally listed in the order in which they are found in the software.

[BSFN BUILD]

The BSFN BUILD settings are:

Setting	Typical Value	Purpose
Build Area=	/usr/jdedwards/811/packages	The location on the server where the package will be built.
Optimization Flags=	+01 (default for HP-UX) -02 (default for AIX) -0 (default for Solaris)	Machine dependent. These compile flags are used when building business functions in Release mode. You should not change these flags.
DebugFlags=	-g -y -D_DEBUG -DJDEDEBUG (default for HP-UX) -g -qfulpath -qdbextra -D_DEBUG -DJDEDEBUG (default for AIX) -g -D_DEBUG -DJDEDEBUG (default for Solaris or Linux)	Machine dependent. These compile flags are used when building business functions in Debug mode. You should not change these flags.
InliningFlags=	blank (default)	This entry is blank for non-iSeries servers.

Setting	Typical Value	Purpose
DefineFlags=	-DKERNEL -DPRODUCTION_ VERSION -DNATURAL_ ALIGNMENT -D_HPUX_SOURCE (default for HP) -D_SUN-SOURCE (default for Solaris) -D_GNU_SOURCE (default for Linux)	These definitions are passed to the compiler. The first three are common on all platforms. The remaining definitions are platform-specific.
CompilerFlags=	-Aa +w1 +z -c (default for HP-UX) -qalign=natural -qflag=I:I -c (default for AIX) -qspill=1024 (AIX only) -xCC -Xa -misalign -KPIC -c (default for Solaris) -fPIC -Wall -c (default for Linux)	Machine dependent. Valid compiler flags. The spill flag sets the stack space when business functions are compiled. This flag is not set by default, but may be needed for some older AIX compilers.
OSReleaseLevel=	+DAportable (for HP-UX only)	The release level to which you are compiling. You should not change these flags.

Setting	Typical Value	Purpose
LinkFlags=	-b -z -B symbolic -ljdesaw -L/usr/jdedwards/811/system/lib (default for HP-UX) -bM:SRE -bexpall -brtl -lc -lm -bnoentry -L. -L/usr//jdedwards/811 /system/lib -ljdelib -lcallobj -lerror -lgentext -ljdb -ljde_erk -ljdecache -ljdeddapi -ljdeknet -ljderepl -ljdeschr -ljdesec -ljdespec -ljdetam -llanguage -lmisc -lpackage -lport -Iqueeknl -lruntime -lsrc -ltransmon -lube -lworkflow -Ijdesaw -ljdenet -lowver -ljdeunicode -lv_verify -bloadmap:loadmap -ljdesaw (default for AIX) -dy-G -L/usr/jdedwards/811 /system/lib -Ijdesaw (default for Solaris) -shared -L/usr/jdedwards/811 /system/lib - ljdesaw (default for Linux)	Machine dependent. These flags are used when linking business functions, including linking them to the jdesaw system that is the shared library. In general, you should not have to change these flags. The -B symbolic setting tells the HP linker to always resolve symbols (calls to functions) in the same library from where they are referenced, if possible. This action prevents a call from one library going to another library of the same name in a different path code.
LinkLibraries=	blank (default)	Libraries to which business functions are linked (Windows and iSeries servers only).
SimultaneousBuilds=	0 (unlimited) (default) any integer (number of simultaneous builds)	Indicates the number of business functions libraries that can be built simultaneously. A value of 0 means that all libraries can be built simultaneously.

[CLUSTER]

The CLUSTER settings are:

Setting	Typical Value	Purpose
Primary Node=	host name	When clustering is used with JD Edwards EnterpriseOne, this setting specifies either a primary server where the software will run or a floating IP address name. This setting can also be used to specify a "virtual" host name that overrides the actual host name. This can be useful for some load balancing applications. This setting is delivered commented out.

[DB SYSTEM SETTINGS]

The settings in this section contain information about the default environment and path code used during the bootstrap process. A directory must reside on the server that has the same name as the default path code shown in its jde.ini file:

Setting	Value	Purpose
Version=	43	A version number to prevent a mismatch of the jde.ini file with the running version of JD Edwards EnterpriseOne.
Default User=	PSFT	The user account name for the database bootstrap tables.
Default Pwd=		The user account password for the database bootstrap tables. This is usually left blank.
Default Role=	*ALL	The role that is used for the bootstrap initialization (usually *ALL).
Default Env=	PD9	The default data source on the enterprise server.
Default PathCode=	PD9	The business function directory that is used for initialization.
Base Datasource=	host - B9 Server Map	The data source representing the database from which sign-in information is retrieved.
Object Owner=	SVM9	The owner of server map database tables.
Server=	server name	The server on which the database resides.
Database=	jde9	The database connect string (Oracle) or database alias name (DB2 UDB) where the server map tables reside.
Load Library=	(blank)	The JD Edwards EnterpriseOne driver library that is used to access the database that stores the server map tables. Beginning in JD Edwards EnterpriseOne 8.9, this value is automatically determined.
Decimal Shift=	Y (default) N	A flag to indicate if decimal shifting is used for numeric data.
Julian Dates=	Y (default) N	A flag to indicate if dates are stored in Julian or database-specific format.
Use Owner=	Y (default) N	A flag to indicate that table names are qualified by owner.

Setting	Value	Purpose
Secured=	Y (default) N	Indicates whether this database is a secured, requiring a user and password login.
Type=	O W	A single character denoting the type of database holding the server map tables. This character can be O (Oracle) or W (DB2 UDB for Windows/Unix). Other values may be valid for accessing databases on other platforms.
JDBNetUse=	N (default) Y	Indicates whether JDBNet is being used to access the bootstrap tables.
UnicodeFlag=	N (default) Y	Indicates whether the database is enabled for Unicode data.
LOBFlag=	N (default) Y	Indicates whether the server map bootstrap tables contain BLOB data.

[DEBUG]

The settings in this section determine the location of the jde.log and jdedebug.log files. The settings are also used to turn the jdedebug.log on and off:

Setting	Typical Value	Purpose
Output=	FILE	Controls the status of the jdedebug file. Valid values are: • NONE: No trace information is written to jdedebug.log. • FILE: Database and runtime trace information is written to the file that is specified by the DebugFile= parameter in the [DEBUG] section.
ClientLog=	1 (default) 0	Valid values are: • 1: Enables servicing of business functions JDE.LOG and JDEDEBUG.LOG entries from the server to the workstation. • 0: Disables this service.
DebugFile=	/u01/jdedwards/811/log /jdedebug.log	The location and name of the jdedebug.log file.
JobFile=	/u01/jdedwards/811/log /jde.log	The location and name of the jde.log file.

Setting	Typical Value	Purpose
LogErrors=	1 (default) 0	The action for error messages. Valid values are: • 0 or FALSE: Indicates that no error messages will be written to JDE.LOG. • 1 or TRUE: Indicates that error messages will be written to JDE.LOG.
CMTrace	0 1	This setting disables (0) or enables (1) cache logging. This setting works in conjunction with the CMTraceFilter. In addition, if output logging is disabled and cache level logging is enabled, the system will continue to format and prepare cache level logging information, but will not display it in the log file. This can cause a degradation in system performance.
CMTraceFilter=	ALL	This setting determines if cache logging messages will appear in the JDEDEBUG.LOG file.
ThreadTraceLevel=	0–3	This setting determines the thread logging level for the JDEDEBUG.LOG file. The higher the value, the more information the system logs. Valid values are: • 0 (Default) Disables thread trace logging. • 1 • 2 • 3 **Note.** Use level 3 logging with caution since excess logging can result in severe degradation of system performance. In addition, if output logging is disabled and thread level logging is enabled, the system will continue to format and prepare thread level logging information, but will not display it in the log file. This can cause a degradation in system performance.
JDETSFile=	/u01/jdedwards/811/log /JDETS.log	Specifies the location of the lock manager trace file.
RepTrace=	1	Enables replication of log messages.
TAMTraceLevel=	0	Set the level of logging for spec (or TAM) file operations.

[ENTERPRISE TIMEZONE ADJUSTMENT]

The setting in this section contains information for properly adjusting the PeopleSoft Enterprise Portal time for single sign-on.

Setting	Purpose
EntNode=	Enables you to enter the difference, in minutes, between Greenwich Mean Time (GMT) and PeopleSoft Enterprise Portal Node time when setting up single sign-on between PeopleSoft Enterprise Portal and JD Edwards EnterpriseOne. You should change this setting whenever daylight savings time changes to reflect the difference between GMT time and the PeopleSoft Enterprise Portal Node time.

[INSTALL]

The settings in this section contain directory paths and general installation information:

Setting	Typical Value	Purpose
DefaultSystem=	system	The name of the subdirectory under /811 that contains the foundation code and tools.
ClientPath=	client	The name of the directory on the deployment server that contains the Workstation install program and other files that are used during deployment.
PackagePath=	package	The name of the subdirectory on the deployment server under a path code that contains the packages that were built for that path code.
DataPath=	data	The name of the subdirectory on the deployment server under the path code that contains the Access database that is delivered for all packages for that path code.
811=	/usr//jdedwards/811	Base path of the JD Edwards EnterpriseOne installation.
Double_Byte=	0	
LocalCodeSet=	WE_ISO88591	A setting that is used to determine alternate language usage. See the appropriate JD Edwards EnterpriseOne Tools 8.12 Upgrade Guide for other language values.

[INTEROPERABILITY]

The INTEROPERABILITY settings are:

Setting	Typical Value	Purpose
LEVEL	EVENTS,DATA	The system writes specified interoperability event data to the debug log file. You can specify one or more of the allowable logging settings. The interoperability documentation contains a list of possible values and their definitions. See *JD Edwards EnterpriseOne Tools 8.96 Interoperability Guide*, "Getting Started with JD Edwards EnterpriseOne Tools Interoperability".
RegisteredEvents=	RTSOOUT	Names of EventTypes. An event is JD Edwards EnterpriseOne business transaction running on JD Edwards EnterpriseOne enterprise server. To enable real-time generation of events, you must register each event that you want to generate in realtime.
FilteredEvents=	*ALL	The value of this parameter defines the events that you want to create in realtime. A value of *ALL generates all registered events. *NONE disables event generation. You can also enter a subset of registered events.
SaveEVNDoc	0 1	This setting determines the logging level for events. You can disable (0) or enable (1) event logging. If this setting is enabled, you must make sure that logging is enabled as well. See the Output and LogErrors settings in the [DEBUG] section to enable logging. **Note.** If output logging is disabled and event level logging is enabled, the system will continue to format and prepare event level logging information, but will not display it in the log file. This can cause a degradation in system performance.

[JDEIPC]

The JDEIPC settings are:

Setting	Typical Value	Purpose
ipcTrace=	0	Set to 1 to enable IPC logging messages. **Important!** This setting can cause the log files to grow very fast.
maxNumberOfSemaphores=	200	This setting defines the size of the semaphore array that JD Edwards EnterpriseOne creates. Changing this setting may require you to update kernel settings on Solaris or Linux platforms.

Setting	Typical Value	Purpose
MaxNumberOfResources	1000	Not delivered in the default jde.ini file. This setting should only be added or changed if the JD Edwards EnterpriseOne system is reporting IPC resource allocation errors.
startIPCKeyValue	7999	Delivered commented out. On UNIX, this setting is the value of the IPC ID that JD Edwards EnterpriseOne uses for its shared memory. This value, plus the maxNumberOfResources, defines the range of IPC IDs that JD Edwards EnterpriseOne will use on the system. The default range is from 5000 to 5999. You should change this value only if there is a conflict with other software running on the server.

[JDEITDRV]

The JDEITDRV settings are:

Setting	Typical Value	Purpose
DrvCount=	3	The number of event drivers that are used for processing messages from event generators, either Z file or real-time.
Drv1=	Z:libzdrv.so (Sun and AIX) Z;libzdrv.sl (HP9000)	The directory location of the Z file event driver.
Drv2=	RT:librtdrv.so (Sun and AIX) RT:librtdrv.sl (HP9000)	The directory location of the real-time event driver.
Drv3=	JDENET:libjdetrdrv.so (Sun and AIX) JDENET:libjdetrdrv.sl (HP9000)	The directory location of the JDENET driver.

[JDEMAIL]

The JDEMAIL setting is:

Setting	Typical Value	Purpose
mailServer=	mail.jdedwards.com	The name of the SMTP server to access for sending server mail messages.

[JDENET]

The JDENET settings are:

Setting	Typical Value	Purpose
serviceNameListen=	6012	The port number or service name that is used by JD Edwards EnterpriseOne to communicate with clients and other servers.
serviceNameConnect=	6012	The port number or service name that is used by JD Edwards EnterpriseOne to communicate with clients and other servers.
maxNetProcesses=	5	Defines the maximum number of jdenet_n processes that can be running. You can increase the value for a server that is expecting a large number of concurrent users. If you need two or more processes, note that the first process is used exclusively to handle new connections and distribute them evenly among the other processes. In this case, you must add an extra process to act as a broker. Thus, the number of connections each jdenet_n process will handle is defined by the formula: (maxNetConnections) / (maxNetProcesses -1) If you increase the maxNetConnections setting, you should also increase the maxNetProcesses setting accordingly. In general, allow at least one jdenet_n process per 250 concurrent users.
maxNetConnections=	1000	The total number of connections that all jdenet_n process can handle. This value is site-specific. You can increase the value for a server that is expecting to handle larger number of concurrent users.
maxKernelProcesses=	50	The maximum number of jdenet_k processes that can be running. The value should be greater than the sum of all maxNumberOfProcesses settings for the dedicated servers.
maxKernelRanges=	30	The number of dedicated server types.
netTrace=	0	Enables JDENET log messages. This setting should be turned off (set to 0) unless specifically needed for debugging, as it can impact performance of the JD Edwards EnterpriseOne system even when debug logging is not enabled.

Setting	Typical Value	Purpose
HandleKrnlSignals=	1	Not delivered in default jde.ini file. Used for debugging purposes. Turns on and off the handling of signals that are delivered to the process. Kernel processes read the setting on startup. A value of 1 turns on handling, which means that the kernel process handles the signal, performs some cleanup tasks, and exits. A value of 0 turns off signal handling. With a value of 0, when signals are delivered to a process, the process writes out a core file. The core file contains data that developers can use to determine the cause and location of the signal. Use the value of 0 for debugging purposes. Once JD Edwards EnterpriseOne service has started, only processes started after you make a change to this setting are affected.
netCoreDump=	0	For internal use only. Not delivered with the RS/6000.
netTemporaryDir=	temporary file directory	Not delivered in default jde.ini file. Use this setting to change the directory to use for JD Edwards EnterpriseOne temporary files. Without this setting, most temporary files are created in the system default directory (usually /tmp). Some parts of the JD Edwards EnterpriseOne system may use the Unix environment variable TMPDIR to determine the location for temporary files.

[JDENET_KERNEL_DEFx]

This section defines internal dedicated server processes for JD Edwards EnterpriseOne. The sections are numbered JDENET_KERNEL_DEF1 to JDENET_KERNEL_DEF30. The settings in these sections should not be changed except as noted:

Setting	Value	Purpose
krnlName	DEF1: JDENET RESERVED KERNEL DEF2: UBE KERNEL DEF3: REPLICATION KERNEL DEF4: SECURITY KERNEL DEF5: LOCK MANAGER KERNEL DEF6: CALL OBJECT KERNEL DEF7: JDBNET KERNEL DEF9: SAW KERNEL DEF10: SCHEDULER KERNEL DEF11: PACKAGE BUILD KERNEL DEF12: UBE SUBSYSTEM KERNEL DEF 13: WORKFLOW KERNEL DEF 14: QUEUE KERNEL DEF 15: XML TRANS KERNEL DEF 16: XML LIST KERNEL DEF 19: EVENT NOTIFICATION KERNEL DEF 20: INTEROPERABILITY EVENT OBSERVER KERNEL DEF 22: XML DISPATCH KERNEL DEF 23: XTX KERNEL DEF 24: XML SERVICE KERNEL DEF30: METADATA KERNEL	• Value should not be changed. • DEF1: Used for internal purposes and testing. • DEF2: Processes batch process requests. • DEF3: Processes data replication requests. • DEF4: Processes security server requests. • DEF5: Processes transaction manager and lock manager requests. • DEF6: Processes requests for remote master business functions (MBF). • DEF7: Processes JDBNet server-to-server requests. • DEF9: Processes SAW application requests. • DEF10: Processes Scheduler application requests. • DEF11: Processes package build requests. • DEF12: Processes UBE subsystem requests. • DEF13: Processes workflow requests. • DEF14: Processes job queue messages. • DEF15: Processes XML transactions. • DEF16: Processes and returns request for data in XML document format. • DEF19: Processes real-time events and XML documents that are generated by the Interoperability Event Observer, as well as Z file events. Publishes all JD Edwards EnterpriseOne events to subscribers. • DEF20: Processes information from business functions calling real-time APIs and uses that information to create an XML or a Z file that is publishable to subscribers by the Event Notification Kernel. • DEF22: Kernel for routing XML messages to their proper kernel. • DEF23: Kernel for transforming XML messages from one type to another. • DEF24: Processes inbound XAPI messages. • DEF30: Processes XML spec access request messages.

Setting	Value	Purpose
dispatchDLLName=	DEF1: libjdenet.{ext} DEF2: libjdeknet.{ext} DEF3: libjderepl.{ext} DEF4: libjdeknet.{ext} DEF5: libtransmon.{ext} DEF6: libxmlcallobj.{ext} DEF7: libjdeknet.{ext} DEF9: libjdesaw.{ext} DEF10: libjdeschr.{ext} DEF11: libjdeknet.{ext} DEF12: jdekern.{ext} DEF13: libworkflow.{ext} DEF14: libqueueknl.{ext} DEF15: libxmltransactions.{ext} DEF16: libxmllist.{ext} DEF19: libjdeie.{ext} DEF20: libjdeieo.{ext} DEF22: libxmldispatch.{ext} DEF23: libxtskrnl.{ext} DEF24: libxmlservice.{ext} DEF30: libmdserializer.{ext}	Values should not be changed. Identifies the name of the library loaded by the given kernel type. The library extension {ext} is .sl for HP-UX, and .so for the other Unix platforms.

Setting	Value	Purpose
dispatchDLLFunction=	DEF1: JDENET_Dispatch Message DEF2: JDEK_ DispatchUBEMessage DEF3: DispatchRepMessage DEF4: JDEK_ DispatchSecurity DEF5: TM_ DispatchTransactionManager DEF6: JDEK_ DispatchCallObject Message DEF7: JDEK_ DispatchJDBNETMessage DEF9: JDEK_ DispatchSAWMessage DEF10: JDEK_ DispatchScheduler DEF11: JDEK_ DispatchPkgBuildMessage DEF12: JDEK_ DispatchUBESBSMessage DEF13: JDEK_ DispatchWFServer Process DEF 14: DispatchQueueMessage DEF16:XMLListDispatch DEF19: JDEK_ DispatchITMessage DEF20: JDEK_ DispatchIEOMessage DEF22: XMLDispatch DEF23: JDEK_ dispatchXTSMessage DEF24: XMLServiceDispatch DEF30: Metadata Dispatch	The name of the main function for handling messages of the give type. The dispatchDLLName and dispatchDLLFunction entries are platform-specific.

Setting	Value	Purpose
maxNumberOfProcesses=	1	The maximum number of kernel processes that can be run on this server for each kernel type. Some settings may need to be changed based on user count.
numberOfAutoStartProcesses=	0	The number of kernel processes that automatically start for each kernel type. If this number is 0, then no processes start automatically for that kernel type. This number must be less than the maximum number of processes for that kernel type. The user can modify this setting to tune performance. The default value is 0 for all JDENET_KERNEL_DEF sections.

[LOCK MANAGER]

The LOCK MANAGER section enables transaction processing and includes the following settings:

Setting	Typical Value	Purpose
Server=	server name	This setting specifies the name of the lock manager server to be used to process records—for example, a server name might be `intelna`.
AvailableService=	TS	This setting indicates the service that the lock manager server is offering. It is also used to indicate whether the lock manager server is on or off. Valid values are: • TS Time stamp service. • NONE No service is available. **Note.** This setting only applies to servers.
RequestedService=	TS	This setting indicates the type of service that the workstation requests from the server. The service that is currently being provided by servers is time stamping (TS) only. Valid values are: • TS • NONE

[LREngine]

The LREngine setting is:

Setting	Typical Value	Purpose
System=	/owdisk2/811/bdev/output	The directory location of the List-Retrieval Engine, a database that is used to manage access to XML repository files.

[NETWORK QUEUE SETTINGS]

The settings in this section contain the name of the queue that is used when running batch jobs on the server. The settings also show the workstation's UBE priority, and whether to hold the jobs in a spool file or immediately send them to a printer:

Setting	Typical Value	Purpose
OutputDirectory=	directory name	Not delivered in the default jde.ini file. This is the directory where you want to create the PrintQueue directory. The default value is the install location found in the [INSTALL] section of the jde.ini.
JDENETTimeout=	60	The timeout value, listed in seconds, for clients to attempt to connect to the server. A server can act as a client when it uses JDBNET, submits UBEs to another server, calls a business function on another server, uses a Lock Manager on another server, or when it makes security server requests to another server.
QKOnldle	300	If the queue kernel is idle for this amount of time, it does a check of each of its queues and performs any necessary cleanup activities. The time is in seconds.

[SAMPLE_EVENT]

The SAMPLE_EVENT settings are:

Setting	Typical Value	Purpose
DS1= DS2= DS3=	D4202150B D4202150C D34A1050C	Defines the data structure for each real-time event registered in the [INTEROPERABILITY] section. Replace [SAMPLE_EVENT] with an event name, such as RTSOOUT, and then enter the values that define the data structure of the event.

[SECURITY]

The SECURITY settings are:

Setting	Typical Value	Purpose
User=	PSFT	This value must be a valid JD Edwards EnterpriseOne user ID and database user ID. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.
Password=	PSFT	This value must be the correct password for the JD Edwards EnterpriseOne user ID specified in the User setting of the [SECURITY] section. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.
Default Role=	*ALL	The role that is used for the security data source (usually *ALL).
DefaultEnvironment=	PD9	Defines a valid environment in which the path code defines table F98OWSEC.
DataSource=	System B9	The name of the data source that contains the security tables.
SecurityServer=	server name	The name of the server that provides security services. Usually this corresponds to the current host.
ServerPswdFile=	TRUE	The setting of this parameter determines whether the software uses special password handling for batch reports running on the server. Set the value to TRUE to instruct the software to enable special handling of passwords. Set the value to FALSE to disable special handling. (Typically, this is only done for debugging purposes.)
History=	0	
secTrace	0–2	This setting determines the security trace logging level for the JDEDEBUG.LOG file. Valid values are: • 0 (Default) Disables security trace logging. • 1 • 2 If output logging is disabled and security trace level logging is enabled, the system will continue to format and prepare security trace level logging information, but will not display it in the log file. This also can cause a degradation in system performance.

[SERVER ENVIRONMENT MAP]

The SERVER ENVIRONMENT MAP settings are:

Setting	Typical Value	Purpose
environmentName=	environment name	Allows one environment name to be translated to another name and be treated by the server accordingly.

[SVR]

The settings in this section contain environment and subdirectory information:

Setting	Typical Value	Purpose
EnvType=	1	Used by JDEKRNL.
EnvironmentName=	PD9	Default JD Edwards EnterpriseOne startup environment.
SpecPath=	spec	This line and all of the following in this section specify the path names so that other JD Edwards EnterpriseOne source programs know where to look for files. For instance, if spec were ever to be changed to specifications, changing SpecPath would allow changes to be made quickly. This value is not updated by any program or process. The only reason to change this setting is aesthetic. This setting is the subdirectory under the path code user to store the replicated set of specification files on the workstation.
SourcePath=	source	NA
ObjectPath=	obj	NA
HeaderPath=	include	NA
HeaderVPath=	includev	NA
BinPath=	bin32	NA
LibPath=	lib32	NA
LibVPath=	libv32	NA
MakePath=	make	NA
WorkPath=	work	NA
CodeGeneratorPath=	cg	NA
ResourcePath=	res	NA

Setting	Typical Value	Purpose
HelpPath=	helps	NA
NextIDPath=	nextid	NA
LibraryListName=	PD9	Default JD Edwards EnterpriseOne startup environment.

[WORKFLOW]

The WORKFLOW settings are:

Setting	Value	Purpose
Asynchronous Workflow=	FALSE	Used to activate and deactivate asynchronous workflow. The default value is FALSE.

[XML Dispatch]

The XML Dispatch setting is:

Setting	Typical Value	Purpose
PollIntervalMillis=	3000	The number of milliseconds that the XML Dispatch kernel sleeps during inactivity when it is waiting on responses from other XML kernels (such as the XML Call Object kernel). The lower this value is, the more CPU cycles the XML Dispatch kernels use when waiting for responses from other XML kernels.

[XTS]

The XTS setting is:

Setting	Typical Value	Purpose
ResponseTimeout=	600	The number of seconds that the XML Dispatch kernel waits for a response from other XML kernels (such as XML Call Object kernel) before giving up on the response.

Windows Enterprise Server jde.ini Settings

This section describes the settings found in the Windows enterprise server jde.ini file. Information is organized by section, such as [DEBUG]. Sections are alphabetized, but settings found within sections are listed in the order in which they appear in the software. For cases when defaults for Intel and Compaq AlphaServer processors differ, the two values are labeled.

[ACTIVE DIRECTORY]

The setting in this section is used when Active Directory is installed:

Setting	Value	Purpose
SCPToPublish	Variable. Typically, use a version of JD Edwards EnterpriseOne running on the server—for example, PeopleSoft_ERP_811_SP1.	Identifies the Service Connection Point (SCP) object in Active Directory. When a user signs in to JD Edwards EnterpriseOne, the software searches Active Directory for an SCP object with a service name that matches the parameter value in the [ACTIVE DIRECTORY] section of the workstation jde.ini file. The software chooses an SCP object with a status of *running* and retrieves the server name and port number, thus enabling the workstation to make a connection to the server. If you move JD Edwards EnterpriseOne service from one server to another or change the service port number, no changes to the workstation jde.ini file are needed, as long as the name of the SCP object in Active Directory and the parameter values of the [ACTIVE DIRECTORY] section of the workstation jde.ini file match.

[BSFN BUILD]

The BSFN BUILD settings are:

Setting	Value	Purpose
DoCompression=	0	Used to compress server packages for redeployment to other servers of the same platform type. This setting saves you from having to build a package on each server. Valid values are: • 0: Do not use compression. • 1: Use compression.
BuildArea=	Z: \jdedwards\811\ddp \packages	The location on the server where the Package Name directory will be created and the package built.
DebugFlags=	/Gz /Od /Zi /MDd /Yd /W4 /GX /Gy /D_DEBUG	Machine dependent. These compile flags are used when building business functions in debug mode. You should not change these flags.
OptimizationFlags=	/Gz /O2 /MD /W4 /GX /Gy	Machine dependent. These compile flags are used when building business functions in release or optimize mode. You should not change these flags.
OSReleaseLevel=	5.0	The Windows server release level to which you are compiling. You should not change this flag.
DefineFlags=	/D WIN32 /D _WINDOWS /D IAMASERVER /D KERNEL	Machine dependent. These compile flags are used when linking business functions. You should not change these flags.

Setting	Value	Purpose
CompilerFlags=	/nologo /c	Machine dependent. These compile flags are used when linking business functions. You should not change these flags.
LinkFlags=	/DLL /DEBUG /SUBSYSTEM:windows /FORCE:MULTIPLE /FORCE:UNRESOLVED /INCREMENTAL:YES /VERBOSE /MAP	Machine dependent. These flags are used when linking business functions. You should not change these flags.
LinkLibraries=	jdekrnl.lib, jdel.lib, jdenet.lib, jdeipc.lib	Libraries to which business functions are linked.
SimultaneousBuilds=	0	Indicates the number of processes that are started for the business function build. 0 means to run as many build processes as possible.

[BSFN Builder]

The settings in this section are for Oracle internal use only:

Setting	Value	Purpose
User=	JDE	User ID used to run BSFNBuilder.exe.
Pwd=	JDE	User password used to run BSFNBuilder.exe.
PathCode=	appl_pgf	Pathcode under which BSFNs are built.
Build Area=	z: \jdedwards\811\ddp	The path to the parent directory of the pathcode for the business functions that you are building. This setting is typically the same as the base installation directory.
DBSFNFlags=	/Gz /Od /Zi /MDd /Yd /W4 /GX /Gy /Fp$(PRECOMPHDR) /D WIN32 /D _DEBUG /D _WINDOWS /D IAMASERVER /D KERNEL /nologo /c	Machine-dependent compiler flags that are used to create debug builds.
RBSFNFlags=	/Gz /O2 /MD /W4 /GX /Gy /Fp$(PRECOMPHDR) /D WIN32 /D NDEBUG /D _WINDOWS /D IAMASERVER /D KERNEL /nologo /c	Machine-dependent compiler flags that are used to create release builds.

Setting	Value	Purpose
DLinkFlags=	/DLL /DEBUG /SUBSYSTEM:windows /out:$(DLLTARGET) /PDB:$(PDB) /IMPLIB:$(LIBRARY) /FORCE:MULTIPLE /FORCE:UNRESOLVED /INCREMENTAL:YES /VERBOSE /MAP	Machine-dependent link flags for debug builds.
RLinkFlags=	/DLL /DEBUG /SUBSYSTEM:windows /out:$(DLLTARGET) /PDB:$(PDB) /IMPLIB:$(LIBRARY) /FORCE:MULTIPLE /FORCE:UNRESOLVED /VERBOSE /MAP:$ (MAPTARGET) /OPT:REF	Machine-dependent link flags for release builds.
KeepMake=	0	The status of make files after the build. Valid values are: • 0: The default setting; do not keep. • 1: Keep.
BFDir=	bsfnerr	Subdirectory under path code that will contain build error logs. The default value is bsfnerr.

[DB SYSTEM SETTINGS]

The settings in this section contain information about the default environment and path code. A directory must reside on the enterprise server that has the same name as the default path code shown in its jde.ini file:

Setting	Value	Purpose
Version=	43	A version number to prevent a mismatch of the jde.ini file with a running version of JD Edwards EnterpriseOne. The only valid value is 43.
Default User=		The JD Edwards EnterpriseOne user ID that is used to access the bootstrap tables, F986101 and F98611.
Default Pwd=		The user password that is used to access the bootstrap tables.
Default Env=	P811HPO1	The environment that is used in situations where an environment is not specified.
Default PathCode=	PROD	The default path code. The specification files for the bootstrap tables are then read from the spec subdirectory of this pathcode folder.

Setting	Value	Purpose
Server=	hp9000a	The server where the bootstrap tables are located. This value is ignored except when jdbnet is used.
Type=	O A S I	The database type where the bootstrap tables reside. Valid values are: • O (Oracle) • A (MS Access) • S (SQL Server) • I (Client Access)

[DEBUG]

The settings in this section determine the location of the jde.log and jdedebug.log. The settings are also used to turn logging on and off:

Setting	Typical Value	Purpose
DebugFile=	z: \jdedwards\811 \ddp\log \jdedebug.log	The path and name of the log file that are used to write debug tracing information. The process ID is added before the period in this file name.
JobFile=	z: \jdedwards\811\ddp\log \jde.log	The path and name of the log file that are used to write job error and warning information. The process ID is added before the period in this file name.
LogErrors=	1 (default) 0	The action for error messages. Valid values are: • 0 or FALSE: Indicates that no error messages will be written to JDE.LOG. • 1 or TRUE: Indicates that error messages will be written to JDE.LOG.
CMTrace	0 1	This setting disables (0) or enables (1) cache logging. This setting works in conjunction with the CMTraceFilter. In addition, if output logging is disabled and cache level logging is enabled, the system will continue to format and prepare cache level logging information, but will not display it in the log file. This can cause a degradation in system performance.
CMTraceFilter=	ALL	This setting determines if cache logging messages will appear in the JDEDEBUG.LOG file.

Setting	Typical Value	Purpose
ThreadTraceLevel=	0–3	This setting determines the thread logging level for the JDEDEBUG.LOG file. The higher the value, the more information the system logs. Valid values are: • 0 (Default) Disables thread trace logging. • 1 • 2 • 3 **Note.** Use level 3 logging with caution since excess logging can result in severe degradation of system performance. In addition, if output logging is disabled and thread level logging is enabled, the system will continue to format and prepare thread level logging information, but will not display it in the log file. This can cause a degradation in system performance.
Output=	FILE	Controls how tracing information is written. Valid values are: • NONE: The default setting; no trace information is written to DebugFile. • FILE: Database and runtime information is written to DebugFile. • AUX: Tracing information is written to the program debugger output window. • BOTH: Tracing information is written to both DebugFile and the program debugger output window.
JDETSFile=	z: \jdedwards\811 \ddp\log\jdets.log	The path and name of the log file that is used to write lock manager tracing information.
KeepLogs=	1	Keeps logs for UBEs in the Print Queue directory. Valid values are: • 1: Keeps the logs created when UBEs are run. • 2: Delete the UBE logs when the UBE is finished processing. Regardless of this setting, logs are kept if an error occurs when processing the UBE.

Setting	Typical Value	Purpose
TAMTraceLevel=	0	Controls the amount of TAM information that is logged to the jdedebug.log. Valid values are: • 0-10, with higher numbers increasing the amount of information being logged. • 0: Default setting; no information output.
TAMTrace=	0	Controls TAM file trace information. Valid values are: • 0: Do not write TAM trace information to the debug file. • 1: Write TAM trace information to the debug file.
ClientLog=	0	Sends log information to the client and merges it with the client's jde.log and jdedebug.log files. Valid values are: • 0: Do not send log information to the client. • 1: Send log information to the client.
QKLog=	0	Controls JDE Queue Kernel tracing information. Valid values are: • 0: Do not write Queue Kernel message information to the debug file. • 1: Write Queue Kernel message information to the debug file.
TraceRowSecurityFetch=	FALSE	Controls row level security tracing. Valid values are: • FALSE (default): Deactivate tracing. • TRUE: Activate tracing.
WTSLogs=	FALSE	Creates logs in the User Profile directory for TSE installations. Valid values are: • FALSE (default): Set the log file paths by the JobFile and DebugFile. • TRUE: Write the log files to c:\WTSRV\Profiles\%Userid%\Windows.
jdelibFatal=	FALSE	Determines whether message boxes are supported. Valid values are: • FALSE (default): Message boxes are not supported. • TRUE: Message boxes are supported.

Setting	Typical Value	Purpose
TAMMultiUserOn=	0	Determines whether multiuser access to TAM files is allowed. Valid values are: • 0 (default). • -1: Do not allow multiuser access. • 1: Allow multiuser access.
TAMErrorMsgBox=	0	Controls whether TAM error messages open a message box. Valid values are: • 0 (default): Do not write fatal error messages to a message box. • 1: Write fatal error messages to a message box.

[ENTERPRISE TIMEZONE ADJUSTMENT]

The setting in this section contains information for properly adjusting the PeopleSoft Enterprise Portal time for single sign-on.

Setting	Purpose
EntNode=	Enables you to enter the difference, in minutes, between Greenwich Mean Time (GMT) and PeopleSoft Enterprise Portal Node time when setting up single sign-on between PeopleSoft Enterprise Portal and JD Edwards EnterpriseOne. You should change this setting whenever daylight savings time changes to reflect the difference between GMT time and the PeopleSoft Enterprise Portal Node time.

[INSTALL]

The settings in this section contain directory paths and general installation information:

Setting	Typical Value	Purpose
B9=	\jdedwards\811\ddp	The path to the JD Edwards EnterpriseOne base installation directory.
LocalCodeSet=	WE_ISO88591	Determines the character code set used by the software. Valid values are: • WE_ISO88591 (1252) - English • JA_SJIS (932) - Japanese • TC_BIG5 (950) - Traditional Chinese • SC_GB (936) - Simplified Chinese • KO_KSC (949) - Korean
StartServicePrefix	psft811	Uniquely identifies JD Edwards EnterpriseOne services to a single installation. The prefix tags JD Edwards EnterpriseOne services when running parallel releases on a single server. The default value is PSFT811.

Setting	Typical Value	Purpose
DefaultSystem	system	The name of the system directory. The default value is system. Do not change this value.
Double_Byte	0	Indicates if this installation is a double-byte installation. Valid values are: • 0 (default): No, not a double-byte installation. • 1: Yes, a double-byte installation.
POSTSCRIPT_ONLY	0	Used with double-byte to force postscript only. Valid values are: • 0 (default): Do not force. • 1: Force.
	1	Used to retrieve the code page for the current process. Valid values are: • 1 (default): Use the 1252 English code set. • <0: Use 1252 (English). • =0: Use the code page found in [INSTALL] LocalCodeSet in the jde.ini file. • >0: Use the code page already in effect.

[INTEROPERABILITY]

The INTEROPERABILITY settings are:

Setting	Typical Value	Purpose
LEVEL	EVENTS,DATA	The system writes specified interoperability event data to the debug log file. You can specify one or more of the allowable logging settings. The interoperability documentation contains a list of possible values and their definitions. See *JD Edwards EnterpriseOne Tools 8.96 Interoperability Guide*, "Getting Started with JD Edwards EnterpriseOne Tools Interoperability".
RegisteredEvents=	RTSOOUT	Names of EventTypes. An event is a JD Edwards EnterpriseOne business transaction running on a enterprise server. To enable real-time generation of events, you must register each event that you want to generate in realtime.

Setting	Typical Value	Purpose
FilteredEvents=	*ALL	The value of this parameter defines the events that you want to create in realtime. A value of *ALL generates all registered events. *NONE disables event generation. You can also enter a subset of registered events.
SaveEVNDoc	0 1	This setting determines the logging level for events. You can disable (0) or enable (1) event logging. If this setting is enabled, you must make sure that logging is enabled as well. See the Output and LogErrors settings in the [DEBUG] section to enable logging. **Note.** If output logging is disabled and event level logging is enabled, the system will continue to format and prepare event level logging information, but will not display it in the log file. This can cause a degradation in system performance.

[JDE_CG]

The JDE_CG settings are:

Setting	Typical Value	Purpose
TARGET=	RELEASE	The type of build that is used to compile objects. Valid values are: • RELEASE (default): Build using release mode. • DEBUG: Build using debug mode.
INCLUDES=	c:\msdev\devstudio\vc\ include	The path to Microsoft Visual C++, system, and JD Edwards EnterpriseOne pathcode include (header) files.
LIBS=	c:\msdev\devstudio\vc\lib	The path to Microsoft Visual C++, JD Edwards EnterpriseOne system, and JD Edwards EnterpriseOne pathcode library files.
MAKEDIR=	c:\msdev\devstudio\vc \bin	The path to the directories of Microsoft Visual C++ programs.
STDLIBDIR=	c:\msdev\devstudio\vc \lib	The path to directories of Microsoft Visual C++ libraries.
ServerPackage Sleep=	60	The wait time, in seconds, between status checks of server package builds. The default value is 60.

[JDEIPC]

The JDEIPC settings are:

Setting	Typical Value	Purpose
ipcTrace=	0	Controls the number of interprocess communications (IPC) written to the jdedebug.log. Valid values are: • 0 (default): Write no messages to the log. • 1: Write only general trace messages to the log. • 2: Write IPC handle state trace messages to the log. • 3: Write both general and IPC handle state trace messages to the log.
startIPCKeyValue=	7001	An integer offset that is used to separate globally shared memory when running multiple instances of JD Edwards EnterpriseOne. The values of these keys for each instance must differ by at least the value of maxNumberOfResources. The default value is 5000.
maxNumberOfResources=	1000	The maximum number of IPC resources that the JD Edwards EnterpriseOne instance will use. When this value is reached, no more IPC resources can be created. The default value is 1000.
maxNumberOfSemaphores=	100	The maximum number of semaphore resources that JD Edwards EnterpriseOne will use. When this value is reached, no more semaphore resources can be created. On Windows NT, two semaphore resources are used to implement each message queue. The default value is 100.
maxMsgqMsgBytes=	2048	The maximum number of bytes in a message to be put on a message queue. The default value is 2048 (2K).
maxMsgqEntries=	1024	The maximum number of messages that can be on a message queue at one time. The default value is 1024.
maxMsgqBytes=	65536	The maximum number of bytes that can be on a message queue at one time. The default value is 65536 (64K).

[JDEITDRV]

The JDEITDRV settings are:

Setting	Typical Value	Purpose
DrvCount=	3	The number of event drivers that is used for processing messages from event generators, either Z file or real-time.

Setting	Typical Value	Purpose
Drv1=	Z:zdrv.dll	The directory location of the Z file event driver.
Drv2=	RT:rtdrv.dll	The directory location of the real-time event driver.
Drv3=	JDENET:jdetrdrv.dll	The directory location of the JDENET driver.

[JDEMAIL]

The JDEMAIL settings are:

Setting	Typical Value	Purpose
ClientType=	Windows HTML	Defines whether the application shortcut that is attached to an external email message will contain a Windows application shortcut or a URL for an HTML application shortcut. The default value is Windows.
mailServer=	owsmtp.jdedwards.com	The domain name of the SMTP server to access for sending server mail messages.
RuleN=	HANDLER: DATA	The SMTP email configuration rules that are taken from the F90005 table. Any empty or invalid entry is considered at the end of the list. N is a positive integer starting at 1.

[JDENET]

The JDENET settings are:

Setting	Typical Value	Purpose
serviceNameListen=	6003 jde_server	The TCP/IP port number that is used for receiving communications packets. If this value is an integer, that number is used as the port. If this value is a character string, it will be translated using the file c:\winnt\system32\drivers\etc\services into a port number and transport protocol. The default value is 6003.
serviceNameConnect=	6004 jde_server	The TCP/IP port number that is used for sending communications packets. If this value is an integer, that number is used as the port. If this value is a character string, it will be translated using the file c:\winnt\system32\drivers\etc\services into a port number and transport protocol. The default value is jde_server.

Setting	Typical Value	Purpose
maxNetProcesses=	1 If you only need one process, then maxNetProcesses=1. However, if you need two or more processes, note that the first process is used exclusively to handle new connections and distribute them evenly among the other processes. In this case, you must add an extra process to act as a broker. To determine the number of processes that you need, you must first determine the maximum number of connections that you need, and the number of connections that a single process can handle. The formula for the number of connections allowed for each process is: (maxNetConnections/ (maxNet Processes - 1)) + 2 Each process requires two extra connections for listening (one for TCP and another for UDP), and an extra process is dedicated to handling incoming connections.	Defines the maximum number of JDENET_N processes that can be running. You can increase the value for a server that is expecting heavy JDENET message flow.
maxNetConnections=	100	The maximum number of connections for all jdesnet and jdenet_n processes that are running. The default value is 100.
maxKernelProcesses=	50	The maximum number of JDENET_K processes that can be running. The value should be greater than all of the values added together in maxNumberOfProcesses for all the dedicated servers.
maxKernelRanges=	30	The number of dedicated server types
NetHostName=		The IP address to use if multiple network cards are used on the server.
netTrace=	1	Enables JDENET log messages.

Setting	Typical Value	Purpose
ServiceControlRefresh	1	The rate in seconds at which the Jdesctrl program refreshes its status of the services. Jdesctrl can be used instead of the Services applet to start, stop, pause, and continue JD Edwards EnterpriseOne net and queue services on Windows NT enterprise servers. The default value is 1.
EnablePredefinedPorts=	0	Allows JD Edwards EnterpriseOne net to use a predefined range of TCP/IP ports. This setting is required to permit the JD Edwards EnterpriseOne Java server outside a firewall. This port range starts at the port number that is specified by serviceNameListen and ends at the port that is calculated by the equation serviceNameListen = maxNetProcesses - 1. The default value, 0, means do not use a predefined range of ports. Set the value at 1 and restart the server if you set the server up behind a firewall.
PreConnectHosts=	0	The number of enterprise servers that will be initialized. This initialization allows the enterprise servers that are listed in the keys PreConnectHost1, PreConnectHost2, and so on, to load their bootstrap tables, thereby improving response time when task requests are actually sent to the servers.
PreConnectHostN	EntServer1	The name of enterprise servers that will be initialized. N is a positive integer, starting with 1.
NetTemporaryDir=	Variable	Allows the Server Administration Workbench (SAW) to create, transfer, and remove temporary log files larger than 5 MB. The variable should be the name of the temporary director that SAW uses to accomplish these tasks.
MaxIPCQueueMsgs	12	For internal use only.
InternalQueueTimeout	30	For internal use only.

[JDENET_KERNEL_DEFx]

This section defines the internal dedicated server processes for JD Edwards EnterpriseOne. The sections are numbered JDENET_KERNEL_DEF1 to JDENET_KERNEL_DEF30. The settings in these sections should not be changed except as noted:

Setting	Value	Purpose
krnlName	DEF1: JDENET RESERVED KERNEL DEF2: UBE KERNEL DEF3: REPLICATION KERNEL DEF4: SECURITY KERNEL DEF5: LOCK MANAGER KERNEL DEF6: CALL OBJECT KERNEL DEF7: JDBNET KERNEL DEF9: SAW KERNEL DEF10: SCHEDULER KERNEL DEF11: PACKAGE BUILD KERNEL DEF12: UBE SUBSYSTEM KERNEL DEF 13: WORKFLOW KERNEL DEF 14: QUEUE KERNEL DEF 15: XML TRANS KERNEL DEF 16: XML LIST KERNEL DEF 19: EVENT NOTIFICATION KERNEL DEF 20: INTEROPERABILITY EVENT OBSERVER KERNEL DEF 22: XML DISPATCH KERNEL DEF 23: XTX KERNEL DEF 24: XML SERVICE KERNEL DEF30: METADATA KERNEL	• Value should not be changed. • DEF1: Used for internal purposes and testing. • DEF2: Processes batch process requests. • DEF3: Processes data replication requests. • DEF4: Processes security server requests. • DEF5: Processes transaction manager and lock manager requests. • DEF6: Processes requests for remote master business functions (MBF). • DEF7: Processes JDBNet server-to-server requests. • DEF9: Processes SAW application requests. • DEF10: Processes Scheduler application requests. • DEF11: Processes package build requests. • DEF12: Processes UBE subsystem requests. • DEF13: Processes workflow requests. • DEF14: Processes job queue messages. • DEF15: Processes XML transactions. • DEF16: Processes and returns request for data in XML document format. • DEF19: Processes real-time events and XML documents that are generated by the Interoperability Event Observer, as well as Z file events. Publishes all JD Edwards EnterpriseOne events to subscribers. • DEF20: Processes information from business functions calling real-time APIs and uses that information to create an XML or a Z file that is publishable to subscribers by the Event Notification Kernel. • DEF22: Kernel for routing XML messages to their proper kernel. • DEF23: Kernel for transforming XML messages from one type to another. • DEF24: Processes inbound XAPI messages. • DEF30: Processes XML spec access request messages.

Setting	Value	Purpose
dispatchDLLName=	DEF1: libjdenet.{ext} DEF2: libjdeknet.{ext} DEF3: libjderepl.{ext} DEF4: libjdeknet.{ext} DEF5: libtransmon.{ext} DEF6: libxmlcallobj.{ext} DEF7: libjdeknet.{ext} DEF9: libjdesaw.{ext} DEF10: libjdeschr.{ext} DEF11: libjdeknet.{ext} DEF12: jdekern.{ext} DEF13: libworkflow.{ext} DEF14: libqueueknl.{ext} DEF15: libxmltransactions.{ext} DEF16: libxmllist.{ext} DEF19: libjdeie.{ext} DEF20: libjdeieo.{ext} DEF22: libxmldispatch.{ext} DEF23: libxtskrnl.{ext} DEF24: libxmlservice.{ext} DEF30: mdserializer.dll.{ext}	Values should not be changed. Identifies the name of the library loaded by the given kernel type.

Setting	Value	Purpose
dispatchDLLFunction=	DEF1: JDENET_Dispatch Message DEF2: JDEK_ DispatchUBEMessage DEF3: DispatchRepMessage DEF4: JDEK_ DispatchSecurity DEF5: TM_ DispatchTransactionManager DEF6: JDEK_ DispatchCallObject Message DEF7: JDEK_ DispatchJDBNETMessage DEF9: JDEK_ DispatchSAWMessage DEF10: JDEK_ DispatchScheduler DEF11: JDEK_ DispatchPkgBuildMessage DEF12: JDEK_ DispatchUBESBSMessage DEF13: JDEK_ DispatchWFServer Process DEF 14: DispatchQueueMessage DEF16:XMLListDispatch DEF19: JDEK_ DispatchITMessage DEF20: JDEK_ DispatchIEOMessage DEF22: XMLDispatch DEF23: JDEK_ dispatchXTSMessage DEF24: XMLServiceDispatch DEF30:_ MetadataDispatch@28	The name of the main function for handling messages of the give type. The dispatchDLLName and dispatchDLLFunction entries are platform-specific.

Setting	Value	Purpose
maxNumberOfProcesses=	1	The maximum number of kernel processes that can be run on this server for each kernel type. Some settings may need to be changed based on user count.
numberOfAutoStartProcesses=	0	The number of kernel processes that automatically start for each kernel type. If this number is 0, then no processes start automatically for that kernel type. This number must be less than the maximum number of processes for that kernel type. The user can modify this setting to tune performance. The default value is 0 for all JDENET_KERNEL_DEF sections.

[LOCK MANAGER]

The LOCK MANAGER section enables transaction processing and includes the following settings:

Setting	Typical Value	Purpose
Server=	server	This setting specifies the name of the lock manager server to be used to process records—for example, a server name might be `intelna`.
RequestedService=	TS	This setting indicates the type of service that the workstation requests from the server. The service that is currently being provided by servers is time stamping (TS) only. Valid value are: • NONE (default) No service is available. • TS Time stamp service.
AvailableService=	NONE	This setting indicates the service that the lock manager server is offering. It is also used to indicate whether the lock manager server is on or off. Valid values are: • NONE (default) • TS **Note.** This setting only applies to servers.
LogServices=	0	Controls lock manager tracing information. Valid values are: • 0 (default): Do not write messages to the file specified in [DEBUG] JDETSFile. • 1: Write messages to this file.

[LREngine]

The LREngine settings are:

Setting	Typical Value	Purpose
System=	P:\Builds\BDEV_WF\output	The directory location of the List-Retrieval Engine, a database that is used to manage access to XML repository files.

[MQSI]

The settings in this section are for the header information on the message that is required for Commerce Integrator. If the adapter is being used without Net Commerce/Commerce Integrator, the create header is *No*; the following jde.ini settings are blank, except for the OWHostName:

Setting	Typical Value	Purpose
QMGRName=	JDE_QMGR	MQ Series queue manager.
QInboundName=	INBOUND.Q	MQ Series inbound message queue name. This queue is used to place incoming MQ Series messages.
QErrorName=	DEFRES.Q	MQ Series default response queue name. This queue is used if a success and failure destination is not provided in the incoming message.
QOutboundName=	OUTBOUND.Q	MQ Series outbound queue name. This queue is used to place outbound MQ Series messages.
TimeoutWaitInterval=	15	Timeout wait interval for the kernel processing.
MaxBufferLength=	10240	The maximum buffer length of an MQ Series message.
CreateHeader=	YES	Create special header information in MQ Series message for Commerce Integrator.
AppGroup=	NNJDE	Used with create header.
JDEOrderStatusCode=	JDESOOUT	Used with create header. Transaction type for JD Edwards EnterpriseOne sales order status.
JDECustomerCode=	JDEAB	Used with create header. Transaction type for JD Edwards EnterpriseOne customer add and update.
JDEItemPriceCode=	JDEPRICE	Used with create header. Transaction type for JD Edwards EnterpriseOne price update.
JDEItemQtyCode=	JDEIL	Used with create header. Transaction type for JD Edwards EnterpriseOne product quantity update.
NCOrderStatusCode=	JDE.IC.F4201Z1	Used with create header. Net commerce order status code.

Setting	Typical Value	Purpose
NCCustomerCode=	JDE.IC.F0101Z2	Used with create header. Net commerce customer add and update code.
NCProductPriceCode=	JDE.IC.F4106NC	Used with create header. Net commerce product quantity update code.
NCProductQtyCode=	JDE.IC.F41021Z1	Used with create header. Net commerce product quantity update code.
OWHostName=		JD Edwards EnterpriseOne host name. Used to create outbound net message. The OWHostName creates the net message to trigger the Outbound Adapter. This setting is the name of the server on which JD Edwards EnterpriseOne is installed.

[NETWORK QUEUE SETTINGS]

The settings in this section contain information for starting batch queues:

Setting	Typical Value	Purpose
QEnv=	P811HPO1	The environment for starting batch queues.
QUser=	JDE	The JD Edwards EnterpriseOne user ID for starting batch queues.
QPassword=	JDE	The JD Edwards EnterpriseOne user password for starting batch queues.
QName=	QBATCH	The default queue name if not specified in UBEQueueN, PKGQueueN, or SPCQueueN.
QueueDelay=	30	The time, in seconds, between which the batch queues search for jobs in table F986110. The default value is 5.
JDENETTimeout=	60	The timeout value, listed in seconds, for clients to attempt to connect to the server. A server can act as a client when it uses JDBNET, submits UBEs to another server, calls a business function on another server, uses a Lock Manager on another server, or when it makes security server requests to another server.
UBEQueues=	1	The total number of batch queues that are devoted to handling UBE requests. Set the value at 2 if you launch a subsystem UBE to run on the server. This value allows the subsystem UBE to run in one queue while normal UBEs can run in a separate queue. This setting is necessary because the subsystem UBE goes into a permanent processing mode and consumes all other UBEs in the queue.

Setting	Typical Value	Purpose
UBEQueueN=	QBATCH	The names of the UBE batch queues. *N* ranges from 1 to the value of UBEQueues.
SpecInstallQueues=	1	The total number of batch queues that are devoted to handling spec file installation requests. The default value is 1.
SpcQueues=	QBATCH	The names of the specification installation queues. *N* ranges from 1 to the value of SpcQueues.
KillImmediate=	1	The action of the shutdown process. Valid values are: • 0: Allows batch queue processes to finish their current task after receiving a shutdown request. • 1 (default): Stop queue processes immediately upon receiving a shutdown request.
OutputDirectory=	z: \jdedwards/811 \ddp	The parent directory for the PrintQueue directory where job files (in .pdf format) are located.

[SAMPLE_EVENT]

The SAMPLE_EVENT settings are:

Setting	Typical Value	Purpose
DS1= DS2= DS3=	D4202150B D4202150C D34A1050C	Defines the data structure for each real-time event that is registered in the [INTEROPERABILITY] section. Replace [SAMPLE_EVENT] with an event name, such as RTSOOUT, and then enter the values that define the data structure of the event.

[SECURITY]

The SECURITY settings are:

Setting	Typical Value	Purpose
SecurityServer=	server1	This value must be a valid JD Edwards EnterpriseOne user ID and database user ID. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.

Setting	Typical Value	Purpose
User=	JDE	This value must be the correct password for the JD Edwards EnterpriseOne user ID specified in the User setting of the [SECURITY] section. Accordingly, this means that the user ID and password pair in the [SECURITY] section must be a valid JD Edwards EnterpriseOne user ID and password and database user ID and password.
Password=	JDE	The database account password that is used to access security table F98OWSEC.
DefaultEnvironment=	JDEOPT32	The default environment in which the security kernel runs.
DataSource=	System - 811	The data source where security table F98OWSEC can be found. The default value is System - 811.
SecurityMode=	0, 1, or 2	Controls whether the software accepts a standard logon, unified logon, or both. Valid values are: • 0 (default): Accept only the standard logon. • 1: Accept only the unified logon. • 2: Accept both.
AllowedUsers=	EnterpriseOne _users, Bowens	A comma-delimited list of user accounts, groups of user accounts, or both, that are permitted to sign in to JD Edwards EnterpriseOne using unified logon. This setting allows the users to bypass the JD Edwards EnterpriseOne client sign-in screen.
NumServers=	1	The total number of servers running security services that can validate a connection. The security server request is sent to each security server in turn until one answers the request or no more security servers are listed. The default value is 1.

Setting	Typical Value	Purpose
History=	0	Turns on the sign-on security history logging. This information is stored in table F9312.
secTrace	0–2	This setting determines the security trace logging level for the JDEDEBUG.LOG file. Valid values are: • 0 (Default) Disables security trace logging. • 1 • 2 If output logging is disabled and security trace level logging is enabled, the system will continue to format and prepare security trace level logging information, but will not display it in the log file. This also can cause a degradation in system performance.

[SERVER ENVIRONMENT MAP]

The SERVER ENVIRONMENT MAP settings are:

Setting	Typical Value	Purpose
ENV1=	ENV2	Maps one environment name to another. Wherever ENV1 is to be used on the enterprise server, it is replaced by ENV2. Multiple environment mappings can be specified.

[SVR]

The settings in this section specify path names so that other programs can find source, headers, specifications, and other information:

Setting	Typical Value	Purpose
SpecPath=	spec	The path to TAM files. Do not change.
SourcePath=	source	The path to business function source files. Do not change.
ObjectPath=	obj	The path to business function object files. Do not change.
HeaderPath=	include	The path to business function header files. Do not change.
HeaderVPath=	includev	The path to third-party vendor header files. Do not change.

Setting	Typical Value	Purpose
BinPath=	bin32	The path to system and business function executables and DLLs. Do not change.
LibPath=	lib32	The path to system and business function library files. Do not change.
LibVPath=	libv32	The path to third-party vendor library files. Do not change.
MakePath=	make	The path to business function make files. Do not change.
WorkPath=	work	The path to work files. Do not change.
FontPath=	res\fonts	The path to font files. This setting can be used in creating batch reports.
SysFontPath=	winnt\fonts	The path to Windows NT system font files. This setting can be used in creating batch reports.

Server Jde.ini Settings for WebSphere

This section details the settings that are found in the enterprise server jde.ini file as needed to run WebSphere third-party software. The information in this section supplements the information for the platform-specific sections. Information is organized by section, such as [JDENET]. Sections are alphabetized, but settings found within sections are generally listed in the order they are found in the software.

[JDENET]

The JDENET settings are:

Setting	Typical Value	Purpose
maxKernelRanges=	13	The maximum number of kernel types and ranges that will be used. Verify that this number is updated to the next number.

[JDENET_KERNEL_DEF13]

This section defines JDENET internal dedicated server processes.

Setting	Typical Value	Purpose
krnlName=	MQSI Kernel	
beginningMsgTypeRange=	5513	The beginning message of the range for each kernel type.

Setting	Typical Value	Purpose
endingMsgTypeRange=	6001	The ending message of the range for each kernel type.
dispatchDLLName=	mqsadapt.dll	This setting determines the .DLL that is used for kernel processes.
dispatchDLLFunction=	_JDEK_DispatchMQ SeriesProcess@28	The name of the kernel function for handling kernel request messages.
maxNumberOfProcesses=	1	The maximum number of kernel processes that can be run on this server for each kernel type.
numberOfAutoStartProcesses=	1	The number of kernel processes that automatically start for each kernel type. Verify that this value is 1.

[MQSI]

The settings in this section are for the header information on the message that is required for Commerce Integrator. If the adapter is being used without Net Commerce/Commerce Integrator, the create header is No and the following jde.ini settings would be blank, except for the OWHostName:

Setting	Typical Value	Purpose
QMGRName=	JDE_QMGR	MQ Series queue manager.
QInboundName=	INBOUND.Q	MQ Series inbound message queue name. This queue is used to place incoming MQ Series messages.
QErrorName=	DEFRES.Q	MQ Series default response queue name. This queue is used if a success and failure destination is not provided in the incoming message.
QOutboundName=	OUTBOUND.Q	MQ Series outbound queue name. This queue is used to place outbound MQ Series messages.
TimeoutWaitInterval=	15	Timeout wait interval for the kernel processing.
MaxBufferLength=	10240	The maximum buffer length of an MQ Series message.
CreateHeader=	YES	Create special header information in MQ Series message for Commerce Integrator.
AppGroup=	NNJDE	Used with create header.
JDEOrderStatusCode=	JDESOOUT	Used with create header. Transaction type for sales order status.
JDECustomerCode=	JDEAB	Used with create header. Transaction type for customer add and update.

Setting	Typical Value	Purpose
JDEItemPriceCode=	JDEPRICE	Used with create header. Transaction type for price update.
JDEItemQtyCode=	JDEIL	Used with create header. Transaction type for product quantity update.
NCOrderStatusCode=	JDE.IC.F4201Z1	Used with create header. Net commerce order status code.
NCCustomerCode=	JDE.IC.F0101Z2	Used with create header. Net commerce customer add and update code.
NCProductPriceCode=	JDE.IC.F4106NC	Used with create header. Net commerce product quantity update code.
NCProductQtyCode=	JDE.IC.F41021Z1	Used with create header. Net commerce product quantity update code.
OWHostName=		JD Edwards EnterpriseOne host name. Used to create outbound net message. The OWHostName creates the net message to trigger the Outbound Adapter. This setting is the name of the server on which JD Edwards EnterpriseOne is installed.

Glossary of JD Edwards EnterpriseOne Terms

activity A scheduling entity in JD Edwards EnterpriseOne tools that represents a designated amount of time on a calendar.

activity rule The criteria by which an object progresses from one given point to the next in a flow.

add mode A condition of a form that enables users to input data.

Advanced Planning Agent (APAg) A JD Edwards EnterpriseOne tool that can be used to extract, transform, and load enterprise data. APAg supports access to data sources in the form of rational databases, flat file format, and other data or message encoding, such as XML.

application server A server in a local area network that contains applications shared by network clients.

as if processing A process that enables you to view currency amounts as if they were entered in a currency different from the domestic and foreign currency of the transaction.

alternate currency A currency that is different from the domestic currency (when dealing with a domestic-only transaction) or the domestic and foreign currency of a transaction.

In JD Edwards EnterpriseOne Financial Management, alternate currency processing enables you to enter receipts and payments in a currency other than the one in which they were issued.

as of processing A process that is run as of a specific point in time to summarize transactions up to that date. For example, you can run various JD Edwards EnterpriseOne reports as of a specific date to determine balances and amounts of accounts, units, and so on as of that date.

back-to-back process A process in JD Edwards EnterpriseOne Supply Management that contains the same keys that are used in another process.

batch processing A process of transferring records from a third-party system to JD Edwards EnterpriseOne.

In JD Edwards EnterpriseOne Financial Management, batch processing enables you to transfer invoices and vouchers that are entered in a system other than JD Edwards EnterpriseOne to JD Edwards EnterpriseOne Accounts Receivable and JD Edwards EnterpriseOne Accounts Payable, respectively. In addition, you can transfer address book information, including customer and supplier records, to JD Edwards EnterpriseOne.

batch server A server that is designated for running batch processing requests. A batch server typically does not contain a database nor does it run interactive applications.

batch-of-one immediate A transaction method that enables a client application to perform work on a client workstation, then submit the work all at once to a server application for further processing. As a batch process is running on the server, the client application can continue performing other tasks.

See also direct connect and store-and-forward.

business function A named set of user-created, reusable business rules and logs that can be called through event rules. Business functions can run a transaction or a subset of a transaction (check inventory, issue work orders, and so on). Business functions also contain the application programming interfaces (APIs) that enable them to be called from a form, a database trigger, or a non-JD Edwards EnterpriseOne application. Business functions can be combined with other business functions, forms, event rules,

	and other components to make up an application. Business functions can be created through event rules or third-generation languages, such as C. Examples of business functions include Credit Check and Item Availability.
business function event rule	See named event rule (NER).
business view	A means for selecting specific columns from one or more JD Edwards EnterpriseOne application tables whose data is used in an application or report. A business view does not select specific rows, nor does it contain any actual data. It is strictly a view through which you can manipulate data.
central objects merge	A process that blends a customer's modifications to the objects in a current release with objects in a new release.
central server	A server that has been designated to contain the originally installed version of the software (central objects) for deployment to client computers. In a typical JD Edwards EnterpriseOne installation, the software is loaded on to one machine—the central server. Then, copies of the software are pushed out or downloaded to various workstations attached to it. That way, if the software is altered or corrupted through its use on workstations, an original set of objects (central objects) is always available on the central server.
charts	Tables of information in JD Edwards EnterpriseOne that appear on forms in the software.
connector	Component-based interoperability model that enables third-party applications and JD Edwards EnterpriseOne to share logic and data. The JD Edwards EnterpriseOne connector architecture includes Java and COM connectors.
contra/clearing account	A general ledger account in JD Edwards EnterpriseOne Financial Management that is used by the system to offset (balance) journal entries. For example, you can use a contra/clearing account to balance the entries created by allocations in JD Edwards EnterpriseOne Financial Management.
Control Table Workbench	An application that, during the Installation Workbench processing, runs the batch applications for the planned merges that update the data dictionary, user-defined codes, menus, and user override tables.
control tables merge	A process that blends a customer's modifications to the control tables with the data that accompanies a new release.
cost assignment	The process in JD Edwards EnterpriseOne Advanced Cost Accounting of tracing or allocating resources to activities or cost objects.
cost component	In JD Edwards EnterpriseOne Manufacturing, an element of an item's cost (for example, material, labor, or overhead).
cross segment edit	A logic statement that establishes the relationship between configured item segments. Cross segment edits are used to prevent ordering of configurations that cannot be produced.
currency restatement	The process of converting amounts from one currency into another currency, generally for reporting purposes. You can use the currency restatement process, for example, when many currencies must be restated into a single currency for consolidated reporting.
database server	A server in a local area network that maintains a database and performs searches for client computers.
Data Source Workbench	An application that, during the Installation Workbench process, copies all data sources that are defined in the installation plan from the Data Source Master and Table and Data Source Sizing tables in the Planner data source to the system-release number data source. It also updates the Data Source Plan detail record to reflect completion.

date pattern A calendar that represents the beginning date for the fiscal year and the ending date for each period in that year in standard and 52-period accounting.

denominated-in currency The company currency in which financial reports are based.

deployment server A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

detail information Information that relates to individual lines in JD Edwards EnterpriseOne transactions (for example, voucher pay items and sales order detail lines).

direct connect A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one immediate and store-and-forward.

Do Not Translate (DNT) A type of data source that must exist on the iSeries because of BLOB restrictions.

dual pricing The process of providing prices for goods and services in two currencies.

edit code A code that indicates how a specific value for a report or a form should appear or be formatted. The default edit codes that pertain to reporting require particular attention because they account for a substantial amount of information.

edit mode A condition of a form that enables users to change data.

edit rule A method used for formatting and validating user entries against a predefined rule or set of rules.

Electronic Data Interchange (EDI) An interoperability model that enables paperless computer-to-computer exchange of business transactions between JD Edwards EnterpriseOne and third-party systems. Companies that use EDI must have translator software to convert data from the EDI standard format to the formats of their computer systems.

embedded event rule An event rule that is specific to a particular table or application. Examples include form-to-form calls, hiding a field based on a processing option value, and calling a business function. Contrast with the business function event rule.

Employee Work Center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user. Each user has a mailbox that contains workflow and other messages, including Active Messages.

enterprise server A server that contains the database and the logic for JD Edwards EnterpriseOne.

EnterpriseOne object A reusable piece of code that is used to build applications. Object types include tables, forms, business functions, data dictionary items, batch processes, business views, event rules, versions, data structures, and media objects.

EnterpriseOne process A software process that enables JD Edwards EnterpriseOne clients and servers to handle processing requests and run transactions. A client runs one process, and servers can have multiple instances of a process. JD Edwards EnterpriseOne processes can also be dedicated to specific tasks (for example, workflow messages and data replication) to ensure that critical processes don't have to wait if the server is particularly busy.

Environment Workbench An application that, during the Installation Workbench process, copies the environment information and Object Configuration Manager tables for each environment from the Planner data source to the system-release number data source. It also updates the Environment Plan detail record to reflect completion.

escalation monitor A batch process that monitors pending requests or activities and restarts or forwards them to the next step or user after they have been inactive for a specified amount of time.

event rule A logic statement that instructs the system to perform one or more operations based on an activity that can occur in a specific application, such as entering a form or exiting a field.

facility An entity within a business for which you want to track costs. For example, a facility might be a warehouse location, job, project, work center, or branch/plant. A facility is sometimes referred to as a "business unit."

fast path A command prompt that enables the user to move quickly among menus and applications by using specific commands.

file server A server that stores files to be accessed by other computers on the network. Unlike a disk server, which appears to the user as a remote disk drive, a file server is a sophisticated device that not only stores files, but also manages them and maintains order as network users request files and make changes to these files.

final mode The report processing mode of a processing mode of a program that updates or creates data records.

FTP server A server that responds to requests for files via file transfer protocol.

header information Information at the beginning of a table or form. Header information is used to identify or provide control information for the group of records that follows.

interface table See Z table.

integration server A server that facilitates interaction between diverse operating systems and applications across internal and external networked computer systems.

integrity test A process used to supplement a company's internal balancing procedures by locating and reporting balancing problems and data inconsistencies.

interoperability model A method for third-party systems to connect to or access JD Edwards EnterpriseOne.

in-your-face-error In JD Edwards EnterpriseOne, a form-level property which, when enabled, causes the text of application errors to appear on the form.

IServer service This internet server service resides on the web server and is used to speed up delivery of the Java class files from the database to the client.

jargon An alternative data dictionary item description that JD Edwards EnterpriseOne appears based on the product code of the current object.

Java application server A component-based server that resides in the middle-tier of a server-centric architecture. This server provides middleware services for security and state maintenance, along with data access and persistence.

JDBNET A database driver that enables heterogeneous servers to access each other's data.

JDEBASE Database Middleware A JD Edwards EnterpriseOne proprietary database middleware package that provides platform-independent APIs, along with client-to-server access.

JDECallObject An API used by business functions to invoke other business functions.

jde.ini A JD Edwards EnterpriseOne file (or member for iSeries) that provides the runtime settings required for JD Edwards EnterpriseOne initialization. Specific versions of the file or member must reside on every machine running JD Edwards EnterpriseOne. This includes workstations and servers.

JDEIPC Communications programming tools used by server code to regulate access to the same data in multiprocess environments, communicate and coordinate between processes, and create new processes.

jde.log The main diagnostic log file of JD Edwards EnterpriseOne. This file is always located in the root directory on the primary drive and contains status and error messages from the startup and operation of JD Edwards EnterpriseOne.

JDENET A JD Edwards EnterpriseOne proprietary communications middleware package. This package is a peer-to-peer, message-based, socket-based, multiprocess communications middleware solution. It handles client-to-server and server-to-server communications for all JD Edwards EnterpriseOne supported platforms.

Location Workbench An application that, during the Installation Workbench process, copies all locations that are defined in the installation plan from the Location Master table in the Planner data source to the system data source.

logic server A server in a distributed network that provides the business logic for an application program. In a typical configuration, pristine objects are replicated on to the logic server from the central server. The logic server, in conjunction with workstations, actually performs the processing required when JD Edwards EnterpriseOne software runs.

MailMerge Workbench An application that merges Microsoft Word 6.0 (or higher) word-processing documents with JD Edwards EnterpriseOne records to automatically print business documents. You can use MailMerge Workbench to print documents, such as form letters about verification of employment.

master business function (MBF) An interactive master file that serves as a central location for adding, changing, and updating information in a database. Master business functions pass information between data entry forms and the appropriate tables. These master functions provide a common set of functions that contain all of the necessary default and editing rules for related programs. MBFs contain logic that ensures the integrity of adding, updating, and deleting information from databases.

master table See published table.

matching document A document associated with an original document to complete or change a transaction. For example, in JD Edwards EnterpriseOne Financial Management, a receipt is the matching document of an invoice, and a payment is the matching document of a voucher.

media storage object Files that use one of the following naming conventions that are not organized into table format: Gxxx, xxxGT, or GTxxx.

message center A central location for sending and receiving all JD Edwards EnterpriseOne messages (system and user generated), regardless of the originating application or user.

messaging adapter An interoperability model that enables third-party systems to connect to JD Edwards EnterpriseOne to exchange information through the use of messaging queues.

messaging server A server that handles messages that are sent for use by other programs using a messaging API. Messaging servers typically employ a middleware program to perform their functions.

named event rule (NER) Encapsulated, reusable business logic created using event rules, rather that C programming. NERs are also called business function event rules. NERs can be reused in multiple places by multiple programs. This modularity lends itself to streamlining, reusability of code, and less work.

nota fiscal In Brazil, a legal document that must accompany all commercial transactions for tax purposes and that must contain information required by tax regulations.

nota fiscal factura In Brazil, a nota fiscal with invoice information.

See also *nota fiscal.*

Object Configuration Manager (OCM) In JD Edwards EnterpriseOne, the object request broker and control center for the runtime environment. OCM keeps track of the runtime locations for business functions, data, and batch applications. When one of these objects is called, OCM directs access to it using defaults and overrides for a given environment and user.

Object Librarian A repository of all versions, applications, and business functions reusable in building applications. Object Librarian provides check-out and check-in capabilities for developers, and it controls the creation, modification, and use of JD Edwards EnterpriseOne objects. Object Librarian supports multiple environments (such as production and development) and enables objects to be easily moved from one environment to another.

Object Librarian merge A process that blends any modifications to the Object Librarian in a previous release into the Object Librarian in a new release.

Open Data Access (ODA) An interoperability model that enables you to use SQL statements to extract JD Edwards EnterpriseOne data for summarization and report generation.

Output Stream Access (OSA) An interoperability model that enables you to set up an interface for JD Edwards EnterpriseOne to pass data to another software package, such as Microsoft Excel, for processing.

package JD Edwards EnterpriseOne objects are installed to workstations in packages from the deployment server. A package can be compared to a bill of material or kit that indicates the necessary objects for that workstation and where on the deployment server the installation program can find them. It is point-in-time snapshot of the central objects on the deployment server.

package build A software application that facilitates the deployment of software changes and new applications to existing users. Additionally, in JD Edwards EnterpriseOne, a package build can be a compiled version of the software. When you upgrade your version of the ERP software, for example, you are said to take a package build.

Consider the following context: "Also, do not transfer business functions into the production path code until you are ready to deploy, because a global build of business functions done during a package build will automatically include the new functions." The process of creating a package build is often referred to, as it is in this example, simply as "a package build."

package location The directory structure location for the package and its set of replicated objects. This is usually \\deployment server\release\path_code\package\package name. The subdirectories under this path are where the replicated objects for the package are placed. This is also referred to as where the package is built or stored.

Package Workbench An application that, during the Installation Workbench process, transfers the package information tables from the Planner data source to the system-release number data source. It also updates the Package Plan detail record to reflect completion.

planning family A means of grouping end items whose similarity of design and manufacture facilitates being planned in aggregate.

preference profile The ability to define default values for specified fields for a user-defined hierarchy of items, item groups, customers, and customer groups.

print server The interface between a printer and a network that enables network clients to connect to the printer and send their print jobs to it. A print server can be a computer, separate hardware device, or even hardware that resides inside of the printer itself.

pristine environment A JD Edwards EnterpriseOne environment used to test unaltered objects with JD Edwards EnterpriseOne demonstration data or for training classes. You must have this environment so that you can compare pristine objects that you modify.

processing option A data structure that enables users to supply parameters that regulate the running of a batch program or report. For example, you can use processing options to specify default values for certain fields, to determine how information appears or is printed, to specify date ranges, to supply runtime values that regulate program execution, and so on.

production environment A JD Edwards EnterpriseOne environment in which users operate EnterpriseOne software.

production-grade file server A file server that has been quality assurance tested and commercialized and that is usually provided in conjunction with user support services.

program temporary fix (PTF) A representation of changes to JD Edwards EnterpriseOne software that your organization receives on magnetic tapes or disks.

project In JD Edwards EnterpriseOne, a virtual container for objects being developed in Object Management Workbench.

promotion path The designated path for advancing objects or projects in a workflow. The following is the normal promotion cycle (path):

11>21>26>28>38>01

In this path, *11* equals new project pending review, *21* equals programming, *26* equals QA test/review, *28* equals QA test/review complete, *38* equals in production, *01* equals complete. During the normal project promotion cycle, developers check objects out of and into the development path code and then promote them to the prototype path code. The objects are then moved to the productions path code before declaring them complete.

proxy server A server that acts as a barrier between a workstation and the internet so that the enterprise can ensure security, administrative control, and caching service.

published table Also called a master table, this is the central copy to be replicated to other machines. Residing on the publisher machine, the F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

publisher The server that is responsible for the published table. The F98DRPUB table identifies all of the published tables and their associated publishers in the enterprise.

pull replication One of the JD Edwards EnterpriseOne methods for replicating data to individual workstations. Such machines are set up as pull subscribers using JD Edwards EnterpriseOne data replication tools. The only time that pull subscribers are notified of changes, updates, and deletions is when they request such information. The request is in the form of a message that is sent, usually at startup, from the pull subscriber to the server machine that stores the F98DRPCN table.

QBE An abbreviation for query by example. In JD Edwards EnterpriseOne, the QBE line is the top line on a detail area that is used for filtering data.

real-time event A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and to provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when certain transactions occur.

refresh A function used to modify JD Edwards EnterpriseOne software, or subset of it, such as a table or business data, so that it functions at a new release or cumulative update level, such as B73.2 or B73.2.1.

replication server A server that is responsible for replicating central objects to client machines.

quote order In JD Edwards Procurement and Subcontract Management, a request from a supplier for item and price information from which you can create a purchase order.

	In JD Edwards Sales Order Management, item and price information for a customer who has not yet committed to a sales order.
selection	Found on JD Edwards EnterpriseOne menus, a selection represents functions that you can access from a menu. To make a selection, type the associated number in the Selection field and press Enter.
Server Workbench	An application that, during the Installation Workbench process, copies the server configuration files from the Planner data source to the system-release number data source. It also updates the Server Plan detail record to reflect completion.
spot rate	An exchange rate entered at the transaction level. This rate overrides the exchange rate that is set up between two currencies.
Specification merge	A merge that comprises three merges: Object Librarian merge, Versions List merge, and Central Objects merge. The merges blend customer modifications with data that accompanies a new release.
specification	A complete description of a JD Edwards EnterpriseOne object. Each object has its own specification, or name, which is used to build applications.
Specification Table Merge Workbench	An application that, during the Installation Workbench process, runs the batch applications that update the specification tables.
store-and-forward	The mode of processing that enables users who are disconnected from a server to enter transactions and then later connect to the server to upload those transactions.
subscriber table	Table F98DRSUB, which is stored on the publisher server with the F98DRPUB table and identifies all of the subscriber machines for each published table.
supplemental data	Any type of information that is not maintained in a master file. Supplemental data is usually additional information about employees, applicants, requisitions, and jobs (such as an employee's job skills, degrees, or foreign languages spoken). You can track virtually any type of information that your organization needs. For example, in addition to the data in the standard master tables (the Address Book Master, Customer Master, and Supplier Master tables), you can maintain other kinds of data in separate, generic databases. These generic databases enable a standard approach to entering and maintaining supplemental data across JD Edwards EnterpriseOne systems.
table access management (TAM)	The JD Edwards EnterpriseOne component that handles the storage and retrieval of use-defined data. TAM stores information, such as data dictionary definitions; application and report specifications; event rules; table definitions; business function input parameters and library information; and data structure definitions for running applications, reports, and business functions.
Table Conversion Workbench	An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.
table conversion	An interoperability model that enables the exchange of information between JD Edwards EnterpriseOne and third-party systems using non-JD Edwards EnterpriseOne tables.
table event rules	Logic that is attached to database triggers that runs whenever the action specified by the trigger occurs against the table. Although JD Edwards EnterpriseOne enables event rules to be attached to application events, this functionality is application specific. Table event rules provide embedded logic at the table level.
terminal server	A server that enables terminals, microcomputers, and other devices to connect to a network or host computer or to devices attached to that particular computer.

three-tier processing The task of entering, reviewing and approving, and posting batches of transactions in JD Edwards EnterpriseOne.

three-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing receipt information to supplier's invoices to create vouchers. In a three-way match, you use the receipt records to create vouchers.

transaction processing (TP) monitor A monitor that controls data transfer between local and remote terminals and the applications that originated them. TP monitors also protect data integrity in the distributed environment and may include programs that validate data and format terminal screens.

transaction set An electronic business transaction (electronic data interchange standard document) made up of segments.

trigger One of several events specific to data dictionary items. You can attach logic to a data dictionary item that the system processes automatically when the event occurs.

triggering event A specific workflow event that requires special action or has defined consequences or resulting actions.

two-way voucher match In JD Edwards Procurement and Subcontract Management, the process of comparing purchase order detail lines to the suppliers' invoices to create vouchers. You do not record receipt information.

User Overrides merge Adds new user override records into a customer's user override table.

variance In JD Edwards Capital Asset Management, the difference between revenue generated by a piece of equipment and costs incurred by the equipment.

In JD Edwards EnterpriseOne Project Costing and JD Edwards EnterpriseOne Manufacturing, the difference between two methods of costing the same item (for example, the difference between the frozen standard cost and the current cost is an engineering variance). Frozen standard costs come from the Cost Components table, and the current costs are calculated using the current bill of material, routing, and overhead rates.

Version List merge The Versions List merge preserves any non-XJDE and non-ZJDE version specifications for objects that are valid in the new release, as well as their processing options data.

visual assist Forms that can be invoked from a control via a trigger to assist the user in determining what data belongs in the control.

vocabulary override An alternate description for a data dictionary item that appears on a specific JD Edwards EnterpriseOne form or report.

wchar_t An internal type of a wide character. It is used for writing portable programs for international markets.

web application server A web server that enables web applications to exchange data with the back-end systems and databases used in eBusiness transactions.

web server A server that sends information as requested by a browser, using the TCP/IP set of protocols. A web server can do more than just coordination of requests from browsers; it can do anything a normal server can do, such as house applications or data. Any computer can be turned into a web server by installing server software and connecting the machine to the internet.

Windows terminal server A multiuser server that enables terminals and minimally configured computers to display Windows applications even if they are not capable of running Windows software themselves. All client processing is performed centrally at the Windows

terminal server and only display, keystroke, and mouse commands are transmitted over the network to the client terminal device.

workbench A program that enables users to access a group of related programs from a single entry point. Typically, the programs that you access from a workbench are used to complete a large business process. For example, you use the JD Edwards EnterpriseOne Payroll Cycle Workbench (P07210) to access all of the programs that the system uses to process payroll, print payments, create payroll reports, create journal entries, and update payroll history. Examples of JD Edwards EnterpriseOne workbenches include Service Management Workbench (P90CD020), Line Scheduling Workbench (P3153), Planning Workbench (P13700), Auditor's Workbench (P09E115), and Payroll Cycle Workbench.

work day calendar In JD Edwards EnterpriseOne Manufacturing, a calendar that is used in planning functions that consecutively lists only working days so that component and work order scheduling can be done based on the actual number of work days available. A work day calendar is sometimes referred to as planning calendar, manufacturing calendar, or shop floor calendar.

workflow The automation of a business process, in whole or in part, during which documents, information, or tasks are passed from one participant to another for action, according to a set of procedural rules.

workgroup server A server that usually contains subsets of data replicated from a master network server. A workgroup server does not perform application or batch processing.

XAPI events A service that uses system calls to capture JD Edwards EnterpriseOne transactions as they occur and then calls third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested notification when the specified transactions occur to return a response.

XML CallObject An interoperability capability that enables you to call business functions.

XML Dispatch An interoperability capability that provides a single point of entry for all XML documents coming into JD Edwards EnterpriseOne for responses.

XML List An interoperability capability that enables you to request and receive JD Edwards EnterpriseOne database information in chunks.

XML Service An interoperability capability that enables you to request events from one JD Edwards EnterpriseOne system and receive a response from another JD Edwards EnterpriseOne system.

XML Transaction An interoperability capability that enables you to use a predefined transaction type to send information to or request information from JD Edwards EnterpriseOne. XML transaction uses interface table functionality.

XML Transaction Service (XTS) Transforms an XML document that is not in the JD Edwards EnterpriseOne format into an XML document that can be processed by JD Edwards EnterpriseOne. XTS then transforms the response back to the request originator XML format.

Z event A service that uses interface table functionality to capture JD Edwards EnterpriseOne transactions and provide notification to third-party software, end users, and other JD Edwards EnterpriseOne systems that have requested to be notified when certain transactions occur.

Z table A working table where non-JD Edwards EnterpriseOne information can be stored and then processed into JD Edwards EnterpriseOne. Z tables also can be used to retrieve JD Edwards EnterpriseOne data. Z tables are also known as interface tables.

Z transaction Third-party data that is properly formatted in interface tables for updating to the JD Edwards EnterpriseOne database.

Index

I

J

L

M

N

O

P

Q

R

S

T

U

V

W