

JD Edwards EnterpriseOne Tools
Performance Monitor Guide
Release 8.98 Update 4
E14715-02

March 2011

Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark licensed through X/Open Company, Ltd.

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

Contents

Preface	vii
Audience.....	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
1 Introduction to JD Edwards EnterpriseOne Performance Monitor	
1.1 JD Edwards EnterpriseOne Performance Monitor Overview.....	1-1
1.2 JD Edwards EnterpriseOne Performance Monitor Implementation.....	1-1
2 Understanding JD Edwards EnterpriseOne Performance Monitor	
2.1 JD Edwards EnterpriseOne Performance Monitor	2-1
2.1.1 JD Edwards EnterpriseOne Performance Monitor Output.....	2-1
2.2 JD Edwards EnterpriseOne Performance Monitor Elements	2-2
2.2.1 The Monitored System	2-4
2.2.1.1 JD Edwards EnterpriseOne Application Database.....	2-4
2.2.1.2 Application Server.....	2-4
2.2.1.3 Web Server.....	2-5
2.2.2 The Monitoring System.....	2-5
2.2.2.1 JD Edwards EnterpriseOne Performance Monitor Java Adapter.....	2-5
2.2.2.2 JD Edwards EnterpriseOne PPMI Servlet.....	2-5
2.2.2.3 JD Edwards EnterpriseOne Monitor Servlet	2-6
2.2.2.4 JD Edwards EnterpriseOne Data Collator Servlet.....	2-6
2.3 Implementing the Production Monitoring System	2-6
2.4 JD Edwards EnterpriseOne Performance Monitor Instrumentation.....	2-6
2.4.1 Understanding PMUs	2-7
2.4.2 Understanding Events	2-7
2.5 Performance Integration with Third-Party Systems.....	2-8
2.6 JD Edwards EnterpriseOne Performance Monitor Data.....	2-8
2.7 Scope of JD Edwards EnterpriseOne Performance Monitor	2-8
3 Setting Up JD Edwards EnterpriseOne Performance Monitor	
3.1 Configuring JD Edwards EnterpriseOne Performance Monitor.....	3-1
3.2 Enabling the Required Elements on the Monitoring System	3-1
3.2.1 Setting Up JD Edwards EnterpriseOne Performance Monitor Database	3-1

3.2.2	Setting Up the Web Server for the Monitoring System.....	3-2
3.2.3	Configuring JAS.INI file on the Monitoring Server.....	3-3
3.2.3.1	[JDENET KERNEL_DEF2]	3-4
3.2.3.2	[JDENET]	3-4
3.2.3.3	[PERFMON]	3-4
3.2.3.4	[PERFCOL]	3-6
3.3	Enabling the Required Elements on the Monitored System.....	3-6
3.3.1	Setting Up the Database of the Monitored System	3-7
3.3.2	Specifying the Monitor URL	3-7
3.3.3	Setting Up the Web Server of the Monitored System.....	3-7
3.3.4	Configuring JAS.INI file on the Monitored Server (JD Edwards EnterpriseOne JAS Server)	3-7
3.3.4.1	[PERFMON]	3-7
3.3.5	Setting Up the Application Server.....	3-8
3.3.6	Configuring JDE.INI file on the Monitored Server (JD Edwards EnterpriseOne Application Server)	3-8
3.3.6.1	[PERFMON]	3-9

4 Administering JD Edwards EnterpriseOne Performance Monitor

4.1	Understanding JD Edwards EnterpriseOne Performance Monitor Administration	4-1
4.1.1	JD Edwards EnterpriseOne Performance Monitor Administration.....	4-1
4.1.2	Forms Used for JD Edwards EnterpriseOne Performance Monitor Administration	4-2
4.2	Setting Global System Options	4-3
4.3	Setting System Definition Defaults	4-4
4.4	Modifying System Definitions	4-5
4.5	Viewing Agent Definitions.....	4-6
4.6	Setting Agent Filter Options.....	4-7
4.7	Scheduling the Reaper Program	4-8

5 Monitoring System Performance

5.1	System JD Edwards EnterpriseOne Performance Monitoring.....	5-1
5.1.1	Monitoring System Performance.....	5-1
5.1.2	Forms Used to Modify System Performance	5-2
5.2	Viewing Open PMUs.....	5-2
5.3	Monitoring Individual User Performance.....	5-3
5.4	Viewing Open PMU Trees.....	5-4

6 Evaluating JD Edwards EnterpriseOne Performance Monitor Meta-Data

6.1	Understanding JD Edwards EnterpriseOne Performance Monitor Meta-Data Definitions	6-1
6.1.1	Metrics	6-1
6.1.2	Events	6-1
6.1.3	Contexts.....	6-2
6.1.4	PMUs	6-2
6.1.5	Forms Used to Evaluate JD Edwards EnterpriseOne Performance Monitor Meta-Data.....	6-3

6.2	Revising Metric Definitions.....	6-4
6.3	Revising Event Definitions	6-6
6.3.1	Context Tab.....	6-6
6.3.2	Metric Tab	6-6
6.4	Revising Context Definitions	6-6
6.5	Revising PMU Definitions	6-7

7 Analyzing Historical Performance Data

7.1	Historical Performance Data	7-1
7.1.1	Historical Performance Data.....	7-1
7.1.2	Forms Used to Analyze Historical Performance Data	7-1
7.2	Viewing Completed PMUs.....	7-2
7.2.1	Entering Search Criteria.....	7-2
7.2.2	Working with Completed PMUs.....	7-2
7.2.2.1	Completed PMUs	7-3
7.2.2.2	Context	7-4
7.2.2.3	Metrics.....	7-4
7.2.2.4	Additional Data	7-4
7.2.2.5	Agent Details.....	7-4
7.2.2.6	Identification	7-4
7.2.3	Completed PMU Status	7-4

A Understanding JD Edwards EnterpriseOne Performance Monitor Architecture

A.1	JD Edwards EnterpriseOne Performance Monitor Architecture Components	A-1
A.1.1	JD Edwards EnterpriseOne Application Server.....	A-1
A.1.1.1	AppServer C-Agent.....	A-1
A.1.1.2	Archive Batch Process.....	A-1
A.1.1.3	Reaper Batch Process	A-2
A.1.2	JD Edwards EnterpriseOne JAVA Application Server	A-2
A.1.2.1	JAS Runtime Engine.....	A-2
A.1.3	JD Edwards EnterpriseOne Monitoring Server.....	A-2
A.1.3.1	PerfMon Java Adapter	A-2
A.1.3.2	Monitor Servlet	A-2
A.1.3.3	Data Collator Servlet.....	A-3
A.1.3.4	PPMI Servlet.....	A-3
A.1.4	Data Flow	A-3
A.1.4.1	Sending Performance Data to Monitor Server	A-3
A.1.4.2	Saving performance Data in database	A-3
A.1.4.3	Sending configuration changes to the Agents.....	A-4
A.1.4.4	Archiving data in database	A-4
A.1.4.5	Reaping data in database.....	A-4

B PMU Definition Reference

B.1	PMU 100 C Call Object.....	B-1
B.1.1	Filter Level	B-1
B.1.2	Contexts.....	B-1

B.1.3	Metrics	B-2
B.1.4	Additional Label	B-2
B.2	PMU 101 Java Call Object	B-2
B.2.1	Filter Level	B-2
B.2.2	Contexts.....	B-2
B.2.3	Metrics	B-3
B.2.4	Additional Label	B-3
B.3	PMU 200 JDB/JDBj DML APIs	B-3
B.3.1	Filter Level	B-3
B.3.2	Contexts.....	B-3
B.3.3	Metrics	B-3
B.3.4	Additional Label	B-4
B.4	PMU 201 SQL DML APIs.....	B-4
B.4.1	Filter Level	B-4
B.4.2	Contexts.....	B-4
B.4.3	Metrics	B-4
B.4.4	Additional Label	B-5
B.5	PMU 300 Execute Form.....	B-5
B.5.1	Filter Level	B-5
B.5.2	Contexts.....	B-5
B.5.3	Metrics	B-5
B.5.4	Additional Label	B-5
B.6	PMU 301 Event Rules	B-5
B.6.1	Filter Level	B-6
B.6.2	Contexts.....	B-6
B.6.3	Metrics	B-6
B.6.4	Additional Label	B-6
B.7	PMU 302 Fetch Process	B-6
B.7.1	Filter Level	B-6
B.7.2	Contexts.....	B-6
B.7.3	Metrics	B-6
B.7.4	Additional Label	B-6
B.8	PMU 303 OK Process	B-7
B.8.1	Filter Level	B-7
B.8.2	Contexts.....	B-7
B.8.3	Metrics	B-7
B.8.4	Additional Label	B-7
B.9	PMU 304 Asynch OK Process	B-7
B.9.1	Filter Level	B-7
B.9.2	Contexts.....	B-7
B.9.3	Metrics	B-8
B.9.4	Additional Label	B-8

Glossary

Index

Preface

Welcome to the JD Edwards EnterpriseOne Tools Performance Monitor Guide.

Audience

This guide is intended for system administrators who are responsible for monitoring the performance of JD Edwards EnterpriseOne.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at
<http://www.oracle.com/us/corporate/accessibility/index.html>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/support/contact.html> or visit <http://www.oracle.com/accessibility/support.html> if you are hearing impaired.

Related Documents

You can access related documents from the JD Edwards EnterpriseOne Release Documentation Overview pages on My Oracle Support. Access the main documentation overview page by searching for the document ID, which is 876932.1, or by using this link:

<https://support.oracle.com/CSP/main/article?cmd=show&type=NOT&id=876932.1>

To navigate to this page from the My Oracle Support home page, click the Knowledge tab, and then click the Tools and Training menu, JD Edwards EnterpriseOne, Welcome Center, Release Information Overview.

This guide contains references to server configuration settings that JD Edwards EnterpriseOne stores in configuration files (such as jde.ini, jas.ini, jdbj.ini, jdelog.properties, and so on). Beginning with the JD Edwards EnterpriseOne Tools Release 8.97, it is highly recommended that you only access and manage these settings for the supported server types using the Server Manager program. See the Server Manager Guide on My Oracle Support.

Conventions

The following text conventions are used in this document:

Convention	Meaning
Bold	Indicates field values.
<i>Italics</i>	Indicates emphasis and JD Edwards EnterpriseOne or other book-length publication titles.
Monospace	Indicates a JD Edwards EnterpriseOne program, other code example, or URL.

Introduction to JD Edwards EnterpriseOne Performance Monitor

This chapter contains the following topics:

- [Section 1.1, "JD Edwards EnterpriseOne Performance Monitor Overview"](#)
- [Section 1.2, "JD Edwards EnterpriseOne Performance Monitor Implementation"](#)

1.1 JD Edwards EnterpriseOne Performance Monitor Overview

Oracle's JD Edwards EnterpriseOne Performance Monitor enables you to view real-time and historical performance data of your JD Edwards systems. JD Edwards EnterpriseOne Performance Monitor provides the information that you need to solve immediate performance issues and analyze trends in system performance.

By default, JD Edwards EnterpriseOne Performance Monitor is disabled on a new JD Edwards EnterpriseOne installation. To enable JD Edwards EnterpriseOne Performance Monitor, follow the configuration instructions in this guide.

When you have a JD Edwards application environment running and you have set up JD Edwards EnterpriseOne Performance Monitor, you can begin to capture and analyze JD Edwards EnterpriseOne performance data.

1.2 JD Edwards EnterpriseOne Performance Monitor Implementation

This section provides an overview of the steps that are required to implement JD Edwards EnterpriseOne Performance Monitor.

In the planning phase of your implementation, take advantage of all JD Edwards sources of information, including the installation guides and troubleshooting information.

Follow these steps to implement JD Edwards EnterpriseOne Performance Monitor:

1. Install EnterpriseOne.

See JD Edwards EnterpriseOne installation documentation for your platform.

2. Install one or more EnterpriseOne applications.

See JD Edwards EnterpriseOne installation documentation for your applications.

Understanding JD Edwards EnterpriseOne Performance Monitor

This chapter contains the following topics:

- [Section 2.1, "JD Edwards EnterpriseOne Performance Monitor"](#)
- [Section 2.2, "JD Edwards EnterpriseOne Performance Monitor Elements"](#)
- [Section 2.3, "Implementing the Production Monitoring System"](#)
- [Section 2.4, "JD Edwards EnterpriseOne Performance Monitor Instrumentation"](#)
- [Section 2.5, "Performance Integration with Third-Party Systems"](#)
- [Section 2.6, "JD Edwards EnterpriseOne Performance Monitor Data"](#)
- [Section 2.7, "Scope of JD Edwards EnterpriseOne Performance Monitor"](#)

2.1 JD Edwards EnterpriseOne Performance Monitor

JD Edwards EnterpriseOne Performance Monitor helps system administrators monitor JD Edwards EnterpriseOne performance data and, if wanted, share the data with third-party monitoring tools. JD Edwards EnterpriseOne Performance Monitor is a diagnostic utility for monitoring the performance of the main elements of your JD Edwards EnterpriseOne system, such as web servers, application servers, and process scheduler servers. You can monitor real-time performance and analyze historical data.

JD Edwards EnterpriseOne Performance Monitor reports durations and key metrics of JD Edwards EnterpriseOne runtime execution, such as SQL statements and business functions.

Note: JD Edwards EnterpriseOne Performance Monitor enables you to isolate and identify the causes of performance issues; however, it does not provide automatic solutions to performance issues.

2.1.1 JD Edwards EnterpriseOne Performance Monitor Output

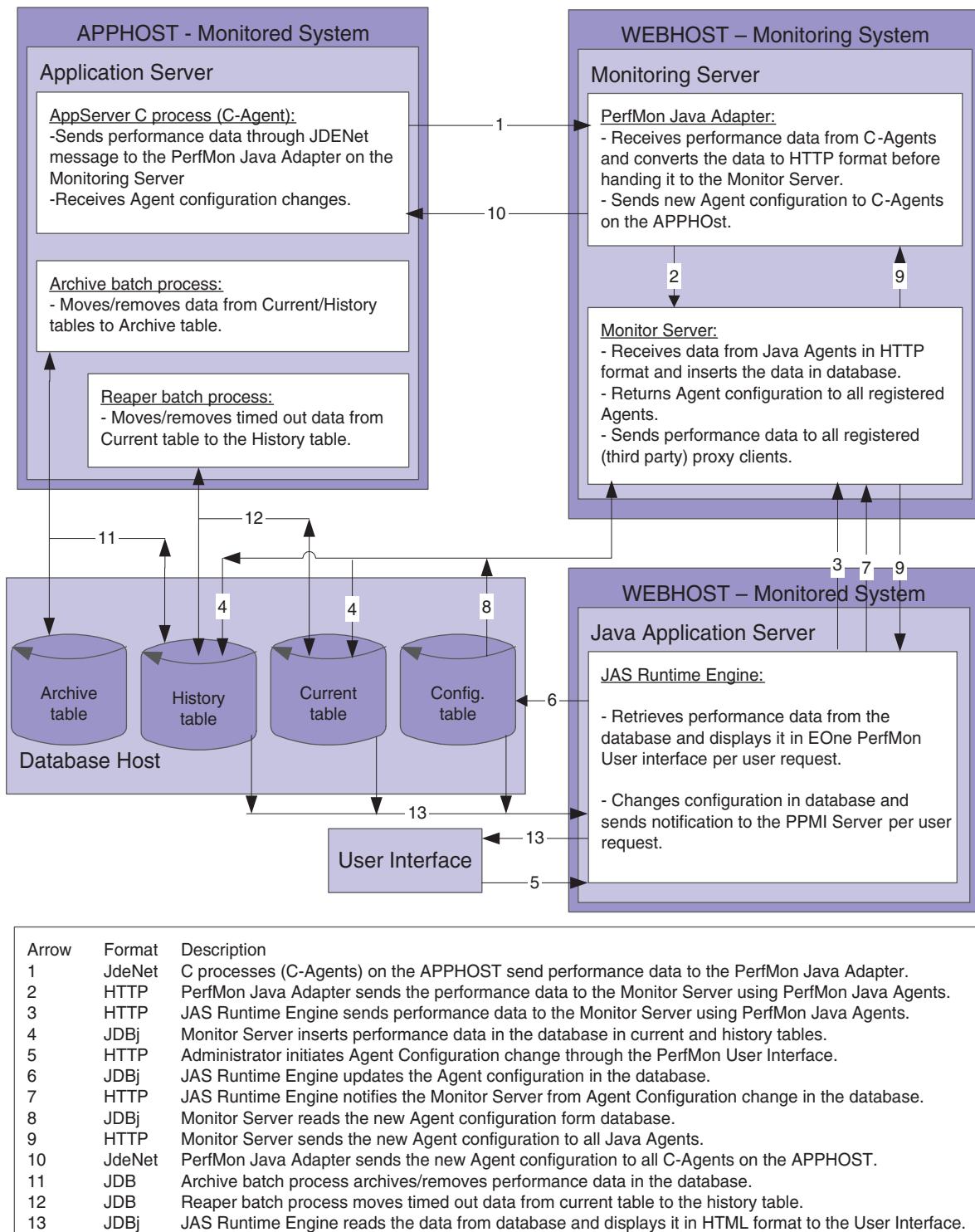
The metrics provided by JD Edwards EnterpriseOne Performance Monitor enable system administrators to:

- Monitor real-time system performance.
- Identify poorly performing tiers, hosts, domains, servers, application code, and SQL statements in an EnterpriseOne environment.
- Identify performance trends.

- Address and isolate performance bottlenecks.

2.2 JD Edwards EnterpriseOne Performance Monitor Elements

This diagram illustrates the main elements of JD Edwards EnterpriseOne Performance Monitor:

Figure 2-1 JD Edwards EnterpriseOne Performance Monitor Elements

For a detail explanation of the diagram, see [Appendix A, "Understanding JD Edwards EnterpriseOne Performance Monitor Architecture"](#)

A monitoring system can monitor one or more JD Edwards EnterpriseOne application systems. However, the monitored system can be monitored by just one monitoring system.

You can configure a self-monitoring system for development and testing.

2.2.1 The Monitored System

The monitored system comprises these main elements:

- JD Edwards EnterpriseOne application database.
- JD Edwards EnterpriseOne application server.
- JD Edwards EnterpriseOne web server.

Note: An application server is also known as a *business logic server* in some materials. However, in this guide, only the term *application server* appears.

JD Edwards EnterpriseOne Performance Monitor agents reside on each main element of the monitored system. Agents run on the servers to gather and report performance information. An agent is a small piece of code operating from within an existing process. It collects performance data and uses a dedicated thread to send the data to the monitoring system. Agent threads report data to the monitoring system's monitor servlet, which in turn reports performance data to the PSPPMSRV server processes in the monitoring system's application server.

Two types of agents exist:

- *Process instrumentation agents* that report data about the specific server activities.
- *Domain resource agents* that report resource usage statistics about the process and server, such as CPU usage.

2.2.1.1 JD Edwards EnterpriseOne Application Database

This database is installed with the installation of the JD Edwards EnterpriseOne system. This database has all the tables to store performance data for EnterpriseOne. The default installation installs both JD Edwards EnterpriseOne application tables and JD Edwards EnterpriseOne Performance Monitor tables within the same database. JD Edwards EnterpriseOne users can use the Object Configuration Management (OCM) settings to use a separate database for Performance Data. Separating performance data and production data into separate databases should enhance the performance of the JD Edwards EnterpriseOne Servers.

2.2.1.2 Application Server

The JD Edwards EnterpriseOne application server, also known as the JD Edwards EnterpriseOne Enterprise Server, contains the CallObject Kernel processes used to run business functions on the application server. Each CallObject Kernel process, or each CallObject thread, in the case of multi-threaded CallObjects, uses a C-Agent to collect and send the performance data to the JD Edwards EnterpriseOne Performance Monitor Java Adapter on the monitoring server.

The C-Agent is a small piece of code that operates within an existing CallObject process or thread. The C-Agent uses the JD Edwards EnterpriseOne JDENet communication protocol to send the performance data to the monitoring server. C-Agents report data to the PerfMon Java Adapter, which in turn sends the data to the

Monitor servlet. The monitoring system's monitor servlet then sends performance data to the monitoring system's Data Collator Servlet.

Note: JD Edwards EnterpriseOne Performance Monitor does not monitor the performance of the RDBMS or the server on which it runs.

2.2.1.3 Web Server

The JD Edwards EnterpriseOne web server, also known as the JD Edwards EnterpriseOne JAS (Java application server), contains Java components of the JD Edwards EnterpriseOne web server. The JD Edwards EnterpriseOne Java components have been instrumented to send performance data to the monitoring server. Each component of the JAS Server uses a Java Agent to collect and send the performance data to the monitor servlet on the monitoring server.

A Java Agent is a small piece of code operating from within an existing JVM process. It collects data and uses a dedicated thread to send the data to the monitoring system. Java Agents report data to the Monitor servlet which in turn sends XML formatted performance data to the Monitoring System's Data Collator servlet.

2.2.2 The Monitoring System

The monitoring system includes these elements:

- JD Edwards EnterpriseOne Performance Monitor Java Adapter.
- JD Edwards EnterpriseOne Monitor Servlet.
- JD Edwards EnterpriseOne Performance Monitor Integration (PPMI) Servlet.
- JD Edwards EnterpriseOne Data Collator Servlet.

2.2.2.1 JD Edwards EnterpriseOne Performance Monitor Java Adapter

JD Edwards EnterpriseOne Performance Monitor Java Adapter is the bridge between the JD Edwards EnterpriseOne application server and the monitoring server. C-Agents on the application server (CallObject Kernel processes/threads) collect and send the performance data to the Java Adapter using the JDENet communication protocol. For each C-Agent on the application server, a Java Agent is created within the JD Edwards EnterpriseOne PerfMon Java Adapter process. Based on the type of the message sent by the C-Agent, the Java Agent calls into the appropriate Java Agent API. From this point on, the performance data is processed the same way as if a Java Agent running on the JD Edwards EnterpriseOne web server sent the data. This architecture allows the same components on the monitoring server (Monitor, PPMI, and Data Collator servlets) to manage Agents and process performance data regardless of the origin of the data: application server or web server.

See [Implementing the Production Monitoring System](#).

2.2.2.2 JD Edwards EnterpriseOne PPMI Servlet

The PPMI servlet registers and manages PPMI clients. The Data Collator is an integrated PPMI client for JD Edwards EnterpriseOne monitoring server. Any third-party solution can register to the PPMI servlet and receive JD Edwards EnterpriseOne performance data.

The PPMI servlet also receives the Agent configuration notifications sent by the JD Edwards EnterpriseOne web server. Administrators can change the Agent configuration for one or all the registered Agents and notify all the Agents of the

changes. The JD Edwards EnterpriseOne web server sends an Agent notification message to the PPMI servlet, which in turn, notifies the Monitor servlet. The Monitor servlet reads the new Agent configuration for the appropriate Agents from the database and sends the configuration back to Agents. Note that each time an Agent sends the performance data to the Monitor Servlet the Agent configuration is sent back to the Agent as the response. The PPMI servlet also adds a new record in the database for each Agent registered with the monitoring system.

2.2.2.3 JD Edwards EnterpriseOne Monitor Servlet

The Monitor servlet receives the performance data from Agents and sends it to the Data Collators. For each message sent by an Agent, the Monitor servlet sends back the Agent configuration. This way, the Agents are always up-to-date of any configuration changes. Note that administrators can change the Agent configurations from the JD Edwards EnterpriseOne PerfMon User Interface.

2.2.2.4 JD Edwards EnterpriseOne Data Collator Servlet

The Data Collator servlet registers itself with the PPMI servlet as a PPMI client. By registering with the PPMI, the Data Collator will receive the performance data sent by the Monitor servlet. Note that the PPMI servlet notifies the Monitor servlet of all newly registered clients. The Data Collator's main function is to write the performance data to the JD Edwards EnterpriseOne PerfMon database. Two main database tables are used to contain the performance data: Current Transaction table and History Transaction table. The Data Collator writes the performance data for a running transaction to the current transaction table until the transaction ends. The Data Collator then writes the End Transaction data to the History Transaction table and removes all records related to this transaction from the Current Transaction table.

2.3 Implementing the Production Monitoring System

In a production environment, you should configure the JD Edwards EnterpriseOne monitoring system separate from the production monitored systems. Any JD Edwards EnterpriseOne system database that is delivered with JD Edwards EnterpriseOne installation contains the appropriate database definitions to store all JD Edwards EnterpriseOne Performance Monitor data. With the separate monitoring system, you can monitor a single JD Edwards EnterpriseOne system, or you can configure it to monitor multiple JD Edwards EnterpriseOne systems.

Note: JD Edwards EnterpriseOne Performance Monitor can monitor any JD Edwards EnterpriseOne system running on 8.95 or greater. Always consult JD Edwards EnterpriseOne Release Notes and guides to become aware of any exceptions or unsupported features between release levels.

2.4 JD Edwards EnterpriseOne Performance Monitor Instrumentation

This section explains the terminology that is associated with JD Edwards EnterpriseOne Performance Monitor instrumentation. Instrumentation refers to the modifications that are made to internal JD Edwards EnterpriseOne programming code that enable the system to report performance data to a monitor. Selected areas of JD Edwards EnterpriseOne runtime source code have been instrumented. When the system runs instrumented code, it generates time-stamped performance metrics that are relevant to that section of code. Agents running on the monitored system send the

performance data to JD Edwards EnterpriseOne Performance Monitor. Instrumentation generates performance measurement units (PMUs) and events.

2.4.1 Understanding PMUs

A PMU is a unit of measure that reflects the execution of a section of code. The system starts and stops a PMU at specific locations in the code, and the system may update a PMU anytime between the start and stop times. JD Edwards EnterpriseOne has defined a set of PMU types, and each type of PMU corresponds to the instrumentation at a specific code location such as a SQL Execute in the application server or a JDBJ request in the web server.

Each PMU includes:

- PMU Type.
- Instance identifier (a unique identifier for a specific PMU instance).
- Start time.
- Stop time.
- Status.
- Metrics (such as number of SQL fetches or buffer size used in a Jolt response).

Open PMUs are those that are currently being processed. These are PMUs for which a "stop" transaction has not yet been received. Completed PMUs are those for which a "stop" transaction has been received. PMUs can assume a parent-child relationship. Child PMUs start within a parent PMU. Child PMUs are linked to their parent by the parent instance ID.

You can view PMUs within a tree structure that reveals the hierarchy of parent-child PMUs and indicates the processing times at each level.

2.4.2 Understanding Events

Events are not implemented for the 8.95 release of JD Edwards EnterpriseOne Performance Monitor.

Events are notifications containing performance metrics that are different from PMUs in that they are not hierarchical and they do not have durations. Events relate to resource usage, such as CPU usage or memory allocation. JD Edwards EnterpriseOne has defined a set of event types, and each type of event is reported at a specific location in the instrumented code.

Each event has:

- Type.
- Instance identifier (a unique identifier for a specific event instance).
- Timestamp.
- Severity.
- Metrics (such as CPU usage and memory allocation).

See [Revising Event Definitions](#).

2.5 Performance Integration with Third-Party Systems

Using web service technology, performance data that is generated by JD Edwards EnterpriseOne Performance Monitor can be shared with third-party performance-monitoring tools by way of JD Edwards EnterpriseOne Performance Monitor Integration (PPMI) API. The PPMI API uses the Simple Object Access Protocol (SOAP).

Note: The PPMI API protocol is not documented in guides. It is used only in products that are developed by JD Edwards EnterpriseOne system and performance management integration partners. You can use the PPMI API only if you are licensed through JD Edwards.

2.6 JD Edwards EnterpriseOne Performance Monitor Data

After JD Edwards EnterpriseOne Performance Monitor is configured to retrieve and store data in the performance database tables, you can view the performance data using a variety of JD Edwards EnterpriseOne applications that:

- Provide a summary of performance data.
- Present raw performance data.

You can access all of the forms that are related to JD Edwards EnterpriseOne Performance Monitor using the JD Edwards EnterpriseOne Performance Monitor menu. Some performance information appears in grids, while other performance information appears in charts. You have the option to view chart information in grids for sorting and downloading to Microsoft Excel.

2.7 Scope of JD Edwards EnterpriseOne Performance Monitor

JD Edwards EnterpriseOne Performance Monitor provides performance metrics for:

- JD Edwards EnterpriseOne application server components
 - Business function processing
 - Database operations (insert, delete, update, select).
- JD Edwards EnterpriseOne WEB Server (also known as JAS) components.
 - Business function processing
 - Database operations (insert, delete, update, select)
 - JD Edwards EnterpriseOne application events.

Setting Up JD Edwards EnterpriseOne Performance Monitor

This chapter contains the following topics:

- [Section 3.1, "Configuring JD Edwards EnterpriseOne Performance Monitor"](#)
- [Section 3.2, "Enabling the Required Elements on the Monitoring System"](#)
- [Section 3.3, "Enabling the Required Elements on the Monitored System"](#)

3.1 Configuring JD Edwards EnterpriseOne Performance Monitor

Configuring JD Edwards EnterpriseOne Performance Monitor involves enabling the required elements on the monitored system and the monitoring system.

When you install the JD Edwards EnterpriseOne System and transfer the appropriate files to the web server and application server, all of the required elements are in place. You do not need to install additional modules for JD Edwards EnterpriseOne Performance Monitor.

3.2 Enabling the Required Elements on the Monitoring System

This section covers the configuration of your monitoring system, which includes:

- Setting up JD Edwards EnterpriseOne Performance Monitor database.
- Setting up the web server for the monitoring system.
- Configuring JAS.INI file on the monitoring server.

3.2.1 Setting Up JD Edwards EnterpriseOne Performance Monitor Database

The monitoring database requires the use of at least the JD Edwards EnterpriseOne System database for the production monitoring system. Set up this database up using the instructions in the JD Edwards EnterpriseOne Installation Guide. To improve performance of the monitored server, you may want to map the PerfMon database tables to a database other than the production database. These tables would need to be mapped:

Table	Server
F95900	PM System Definitions (PSPMSYSDEFN)
F95900D	PM System Definition Languages (PSPMSYSDEFN_LAN)
F95901	PM System Defaults (PSPMSYSDEFAULTS)

Table	Server
F95902	PS System Version (PSPMSYSVERSION)
F95903	PM Table Map (PSPMTABLEMAP)
F95904	PM Global Administration (PSPMMONITORGBL)
F95905	PM Monitor Clustering (PSPMMONINST)
F95906	PM Frequent Users (PSPMOPRDEFN)
F95907	PM Trace Names (PSPMPERFTACE)
F95908	PM Frequent Components (PSPMPNLGRPDEFN)
F95909	PM Agent Definition (PSPMAGENT)
F95910	PM Agent Filters (PSPMAGENTFILTER)
F95911	PM ArchiveParms (PSPMARCHRUNCNTL)
F95912	PM Collator Instances (PSPMCOLLINST)
F959310	PM Event Set (PSPMEVENTSET)
F959310D	PM Event Set Language (PSPMEVENTSET_LN)
F959311	PM Event Definition (PSPMEVENTDEFVN)
F959311D	PM Event Definition Language (PSPMEVENT_LNG)
F959320	PM Transaction Set (PSPMTRANSSET)
F959320D	PM Transaction Set Language (PSPMTRANSSET_LN)
F959321	PM Transaction Definition (PSPMTRANSDEFN)
F959321D	PM Transaction Definition Language (PSPMTRANS_LAN)
F959331	PM Metric Definition (PSPMMETRICDEFN)
F9593311	PM Metric Definition UDC Display (n/a)
F959331D	PM Metric Definition Language (PSPMMETRIC_LNG)
F959332	PM Metric Value (PSPMMETRICVALUE)
F959332D	PM Metric Value Language (PSPMMETRICVALUE_LN)
F959341	PM Context Definition (PSPMCONTEXTDEFN)
F959341D	PM Context Definition Language (PSPMCONTEXTDEFN_LNG)
F95951	PM Event History (PSPMEVENTHIST)
F95952	PM Event Archive (PSPMEVENTARCH)
F95953	PM Event Clone (PSPMEVENTHISTCL)
F95960	PM Transaction Current (PSPMTRANSCURR)
F95961	PM Transaction History (PSPMTRANSHIST)
F95962	PM Transaction Archive (PSPMTRANSARCH)
F95963	PM Transaction Clone (PSPMTRANSHISTCL)

3.2.2 Setting Up the Web Server for the Monitoring System

The JD Edwards EnterpriseOne Transaction Server (RTE) Installer program installs all of the elements that JD Edwards EnterpriseOne Performance Monitor requires on the web server. No additional installations are required.

JD Edwards EnterpriseOne Performance Monitor servlets that run on the web server of the monitoring system are:

- Monitor servlet.
- PPMI servlet.
- Data Collator servlet.

When you start the WEB Server on the monitoring server, different components of the Server start in this order:

1. The JD Edwards EnterpriseOne PerfMon Java Adapter is started and waits for incoming messages from JD Edwards EnterpriseOne application servers.
2. The Data Collator is started.
3. The Data Collator sends a registration request to the PPMI servlet.
4. PPMI servlet registers the Data Collator.
5. Data Collator waits to receive performance data from Monitor servlet.
6. JD Edwards EnterpriseOne JAS Server Agents register with the Monitor servlet.
7. JD Edwards EnterpriseOne CallObject processes on the application server (C-Agents) send the Agent Initialization request to the JD Edwards EnterpriseOne Java Adapter.
8. JD Edwards EnterpriseOne Java Adapter registers the C-Agents with the Monitor servlet.
9. JAS Server Agents send performance data to the Monitor servlet.
10. C-Agents send performance data to the JD Edwards EnterpriseOne Java Adapter.
11. JD Edwards EnterpriseOne Java Adapter sends performance data to the Monitor servlet.
12. Monitor servlet sends performance data to the Data Collator.
13. Data Collator writes performance data to the database.

Note: Make note of the URL of the web server for the monitoring system. On the monitored system, you need to specify the appropriate URL to which performance data is sent.

3.2.3 Configuring JAS.INI file on the Monitoring Server

The JD Edwards EnterpriseOne Transaction Server installer installs the PerfMon components on the Monitoring System. The following sections refer to the JD Edwards EnterpriseOne Transaction Server Install Directory by JD Edwards EnterpriseOneTransInstallDir.

The JAS.INI file on the monitoring server is located in EnterpriseOneTransInstallDir/cfg/jas.ini. This .ini file has some settings in common with the Java Application servers's (JAS) JAS.INI file.

These settings in the jas.ini file on the monitoring server are in addition to these common settings.

3.2.3.1 [JDENET_KERNEL_DEF2]

The [JDENET_KERNEL_DEF2] section defines the JD Edwards EnterpriseOne PerfMon Java Adapter component of the monitoring server. You do not need to change any settings for this section and it is correctly configured after the install.

```
# Name of PerfMon Java adapter kernel.  
#  
krnlName=E1PerfMon Adapter  
  
#  
# Class of PerfMon Java adapter kernel to be loaded; contains dispatch methods.  
#  
processClassName=com.JD Edwards.pt.e1.ppm.E1PmAdapter.E1PmDispatchMessageV1  
  
#  
# Start of message range for this kernel  
#  
startMessageRange=11000  
  
#  
# End of message range for this kernel  
#  
endMessageRange=11100  
  
#  
# Maximum number of threads that this kernel will run.  
#  
maxNumberOfThreads=1
```

3.2.3.2 [JDENET]

In order for the EnterpriseOne PerfMon Java Adapter to start correctly on the monitoring server, these settings need to be set in the [JDENET] section. Note that the value of the variable MaxKernelRanges should be 2.

```
#  
# Maximum number of dedicated kernel types  
#  
maxKernelRanges=2  
  
#  
# JAS Server JDENET Listening port. This is the port on which the JDENet Listener=>  
# will  
# listen for the performance data coming from E1 application server. Note that the  
# value of this setting should be equal to the E1 application server JDE.INI file=>  
# setting  
# for:  
# [PERFMON_JAVA_ADAPTER_1]  
# javaAdapterPort=6545  
  
serviceNameListen=6545
```

3.2.3.3 [PERFMON]

This section defines the settings for JD Edwards EnterpriseOne Performance Monitor servlets on the Server.

```
# This is the URL of the monitor Servlet.  
# ~MONITOR_HOST_NAME~ is the name of the machine where you installed the
```

```

# E1 monitoring server.
# ~MONITOR_LISTEN_PORT~ is the port number on which the Monitor Servlet will
# be listening.
# E1 Transaction Server installer will update this setting in the jas.ini file⇒
# with the
# correct host name and port number.
monitorUrl=http://~MONITOR_HOST_NAME~:~MONITOR_LISTEN_PORT~/ppm/monitor

# A String used to distinguish the host and port of the machines where different⇒
# Agents
# are running. Please use the same host name and port number used in the previous⇒
# setting.
# E1 Transaction Server installer will update this setting in the jas.ini file⇒
# with the
# correct host name and port number.
hostPort=~MONITOR_HOST_NAME~:~MONITOR_LISTEN_PORT~

#
# Transactions will be timed out by the Monitor Server if the "End Transaction"
# is⇒
# not
# received this number of seconds after the Start Transaction.
#
transactionTimeout=60

#
# not currently being used.
#
sampler=60

#
# PPMI_HOST_NAME is the name of the machine where the monitoring server
# is running on.
# PPMI_LISTEN_PORT is the port number on which the monitoring server
# is listening.
#
ppmiUrl=http://~PPMI_HOST_NAME~:~PPMI_LISTEN_PORT~/ppm/ppmi

#
# Maximum number of threads that will be used by Java adapter.
# You can use one thread per CallObject process/thread on the application server.
# For example, if there are 5 CallObject processes on the application server and⇒
# each one
# is configured to have 4 threads then you can set this setting to 20.
#
#
dispatchMaxThread=20

#
# if true, then users can snoop monitor servlet's status.
# Some HRML pages can be used to view the status of different PerfMon Servlets and
# E1PmJavaAdapter process. In order to activate the access to those pages you⇒
# should,
# set this flag to true.
#
PPMConsole=false

#
# Maximum size (megabytes) of memory used by Monitor servlet queues.

```

```
# You can increase this setting if error messages appear on the monitoring server
# complaining about transferring data between different components of the
# monitoring server.
#
PPMMonitorBufferSize=50

#
# Interval (in minutes) until a partially filled batch is flushed by the Monitor⇒
# servlet.
# The batch is flushed if it is filled before this time.
PPMSyncInterval=1
```

3.2.3.4 [PERFCOL]

```
# URL of PPMI to register with.
#
ppmiUrl=http://~PPMI_HOST_NAME~:~PPMI_LISTEN_PORT~/ppm/ppmi
#
# Location of Collator servlet.
# COLLATOR_HOST_NAME is the name of the machine where the Monitoring System
# is installed.
# COLLATOR_PORT_NUMBER is the port number on which the monitoring server
# is listening.
#
collatorUrl=http://~COLLATOR_HOST_NAME~:~COLLATOR_LISTEN_PORT~/ppm/Collator

#
# Group of streaming performance data to be associated with the collator.
# This is a string to distinguish different data collators. Different data
# collators
# running on different monitoring systems can have their own groups.
#
GROUP=group1
#
# Number of elements (for example, transaction start or end) the Monitor servlet
# sends to PPMI clients. #
#
BATCHSIZ=1000
#
# Number of Collator threads to start that will listen to one group.
#
NumCollThreads=1
```

3.3 Enabling the Required Elements on the Monitored System

This section discusses enabling the required elements on the monitored system which is the system for which you intend to collect performance data. A monitored system requires these items to be in place:

- JD Edwards EnterpriseOne application server.
- JD Edwards EnterpriseOne web server (also known as JD Edwards EnterpriseOne JAS Server).
- Valid monitor servlet URL value.

3.3.1 Setting Up the Database of the Monitored System

As long as your JD Edwards EnterpriseOne application server is running against your JD Edwards EnterpriseOne database, no further database configuration is required for JD Edwards EnterpriseOne Performance Monitor.

3.3.2 Specifying the Monitor URL

The monitor URL identifies the Monitor servlet. The monitor URL value must reflect the URL of the monitor servlet that is running on the monitoring system. The readme.txt generated by the installer contains the correct value for the URL. The agents on the monitored system check the monitor URL value periodically. When the monitor URL value is set to NONE, JD Edwards EnterpriseOne Performance Monitor components on the monitored system are disabled.

3.3.3 Setting Up the Web Server of the Monitored System

No additional installation procedure is required. When JD Edwards EnterpriseOne Performance Monitor is turned on (by providing the monitor URL or by setting the flag in the JAS.INI file), agents on the JD Edwards EnterpriseOne web server automatically become active and begin relaying performance data to the monitoring system.

3.3.4 Configuring JAS.INI file on the Monitored Server (JD Edwards EnterpriseOne JAS Server)

The JAS.INI file on the monitored web server requires new settings for PerfMon. If these settings do not exist, you need to add them. The JAS.INI file on the monitored web server is located in: E1JASInstallDir/WebClinet.war/WEB-INF/jas.ini. The following line must exist in the [SERVER COMPONENTS] section in the jas.ini file on the Monitored Server: com.jdedwards.jas.PerfMonInit The following section is added to the JAS.INI file to define the settings for JD Edwards EnterpriseOne Performance Monitor.

3.3.4.1 [PERFMON]

The settings that you need to set after the install are in brackets ({}). For the other settings you can use the default installed values and change them later if needed.

```
# This is the URL of the monitor Servlet.
# The readme.txt file generated by the installer contains the correct value for=>
url_value.
{monitorUrl}---Add the URL value from the readme.txt generated by the installer---

# A String used to distinguish the host and port of the machines where different=>
Agents
# are running. Please use the same host name and port number used in the previous=>
setting.
# E1 Transaction Server installer will update this setting in the jas.ini file=>
with the
# correct host name and port number.
{hostPort}---Add the HOST:PORT from the readme.txt generated by the installer---

#
# Transactions will be timed out by the Monitor Server if the "End Transaction"
is=>
not
```

```
# received this number of seconds after the Start Transaction.  
#  
transactionTimeout=60  
  
#  
# not currently being used.  
#  
sampler=60  
  
#  
# Please use the host name and port number from the readme.txt generated by the  
# installer.  
#  
ppmiUrl=http://~PPMI_HOST_NAME~:~PPMI_LISTEN_PORT~/ppm/ppmi  
  
#  
# if true, then users can snoop monitor servlet's status.  
#  
PPMConsole=false  
  
#  
# Maximum size (megabytes) of memory used by Monitor servlet queues.  
#  
PPMMonitorBufferSize=50  
  
#  
# Interval (in minutes) until a partially filled batch is flushed by the  
# Monitor servlet.  
#  
PPMSyncInterval=1  
  
#  
# Number of Java Agents to initialize at the startup of the JAS Server. Agent  
# initialization is a time consuming process. This many Agents are initialized at⇒  
# the startup  
# and the groups of Agents will be initialized if more Agents are needed. If this⇒  
# setting is  
# missing from the JAS.INI file the default value of 100 will be used by the⇒  
# system.  
ClientCachePerfSize=100
```

3.3.5 Setting Up the Application Server

Performance data is gathered for CallObject Kernel processes running on the monitored application server. No installation or configuration procedures are required. When the monitored system is enabled, agents in every instrumented server process automatically become active and begin relaying performance data to the monitoring system.

Ensure that the agentActive setting in the [PERFMON] section of the JDE.INI file is set to true (enabled). By default, it is set to false (disabled). If the agentActive setting is commented out or set to anything other than true, then the agents are disabled for this server.

3.3.6 Configuring JDE.INI file on the Monitored Server (JD Edwards EnterpriseOne Application Server)

PerfMon requires new settings in the JDE.INI file on the monitored Server.

3.3.6.1 [PERFMON]

The settings that you need to set after the install are in brackets ({}). The other settings you can use the default installed values and change them later if needed.

```

# flag to enable/disable JD Edwards EnterpriseOne Performance Monitor Agents on=>
# the application Server. If=>
set
# to anything other than true no performance data will be sent from the=>
# application Server to
# the Monitor Server.
#
agentActive = true

# This is the URL of the monitor Servlet.
# The readme.txt file generated by the installer contains the correct value for=>
url_value.

{monitorUrl}---Add the URL value from the readme.txt generated by the installer---

# A String used to distinguish the host and port of the machines where different=>
# Agents
# are running. Please use the host name and port number of the JD Edwards=>
# EnterpriseOne application server.
{hostPort}={E1AppServer_HOST_NAME}:{E1AppServer_LISTEN_PORT}
#
# JDENET messages will be timed out by the Application Server if the response is=>
# not
# received before this number of seconds.
#
jdenetTimeout=120

# PerfMon Transactions will be timed out by the Monitor Server if the "End=>
# Transaction"
# is not received this number of seconds after the Start Transaction.
#
transactionTimeout=60

# not currently being used.
#
sampler=60
#
# Default filter mask (Standby) for Agents.
# valid values are:
# Standby 1 No performance data is collected.
# standard4 Minimum amount of performance data is collected.
# verbose5 More performance data is collected.
# debug6Maximum amount of performance data is collected.
#
filterMask = 1

[PERFMON_JAVA_ADAPTER_1]
#
# Please use the host name of the monitoring server provided in the readme.txt=>
# generated
# by the installer.
#
{javaAdapterhost} = -- use the host name of the monitoring System --
#
# Port on which the E1JavaAdapter is listening. Should be the same as the
# [JDENET] serviceNameListen value used by the Monitoring system.

```

```
{javaAdapterPort} = -- use the same value as the serviceNameListen in the [JDENET]
secion in the JAS.INI file on the Monitoring System.
```

Administering JD Edwards EnterpriseOne Performance Monitor

This chapter contains the following topics:

- [Section 4.1, "Understanding JD Edwards EnterpriseOne Performance Monitor Administration"](#)
- [Section 4.2, "Setting Global System Options"](#)
- [Section 4.3, "Setting System Definition Defaults"](#)
- [Section 4.4, "Modifying System Definitions"](#)
- [Section 4.5, "Viewing Agent Definitions"](#)
- [Section 4.6, "Setting Agent Filter Options"](#)
- [Section 4.7, "Scheduling the Reaper Program"](#)

4.1 Understanding JD Edwards EnterpriseOne Performance Monitor Administration

This section discusses JD Edwards EnterpriseOne Performance Monitor administration and lists the forms used for JD Edwards EnterpriseOne Performance Monitor administration.

4.1.1 JD Edwards EnterpriseOne Performance Monitor Administration

JD Edwards EnterpriseOne Performance Monitor administration includes:

- Specifying global settings.
- Viewing performance definitions such as those related to systems, agents, metrics, and so on.
- Setting system defaults.
- Scheduling batch programs that maintain performance data.

4.1.2 Forms Used for JD Edwards EnterpriseOne Performance Monitor Administration

Form Name	FormID	Navigation	Usage
Global Administration - Global Administration Revisions	W95900A	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Global Administration	View and modify global administration settings, such as the PPMI URL value.
System Defaults - System Default Revisions	W95900B	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, System Defaults	Set global system defaults for all monitored systems.
System Definitions - Work With System Definitions	W9500C	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, System Definitions	View and select system definition that is associated with each of the systems that are being monitored.
System Definitions - System Definition Revisions	W95900D	From the System Definitions - Work With System Definitions form, click Find.	View and modify the system definition that is associated with each of the systems that are being monitored. For example, you can set archive, PMU timeout, and agent buffer size.

Form Name	FormID	Navigation	Usage
Agent Definitions - Work With Agent Definitions	W9500E	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Agent Definitions	View and select the definitions of the agents that are running on the monitored system's application server, and web server.
Agent Definitions - Agent Definition Revisions	W95900F	From the Agent Definitions - Agent Definition Revisions form, click Find.	Make active or inactive the agent that you selected.
Agent Filters - Work With Agent Filters	W9500J	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Agent Filters	Search for the filter levels for the agents that are running on the monitored system.
Agent Filters - Agent Filter Revisions	W95900G	From the Agent Filters - Work With Agent Filters form, click Find.	Set the filter levels for the agents.
Submit Job - Work With Batch Versions - Available Versions	W98305WA	JD Edwards EnterpriseOne Menus, Submit Job.	Enter R95900 in the Batch Application field and click Find. Select the Reaper and click Select.
Submit Job - Work With Batch Versions - Available Versions	W98305WA	From the Submit Job - Work With Batch Versions - Available Versions, search for and select a batch application, and click Select.	Select Data Selection, Data Sequencing, or both, and click Submit.

4.2 Setting Global System Options

Access the Global Administration - Global Administration Revisions form.

URL

Displays the URL.

User

User ID used to access the PPMI servlet.

Password

Password used to access the PPMI servlet.

Maximum Rows

Limits the amount of data that can be inserted into the JD Edwards EnterpriseOne Performance Monitor tables.

Last Updated User ID

Specifies the last user to update the record.

Last Updated Date/Time

Specifies the Utme when the record was last updated.

4.3 Setting System Definition Defaults

Access the System Defaults - System Default Revisions form.

The System Defaults form enables you to set default values for all of the monitored systems. When a new systems register with the monitoring system for the first time, the system adopts the default values that you have set.

Using the System Defaults form enables you to set global values for each monitored system rather than modifying the values for each system separately.

Archive Nothing, Delete Data, Archive Data

Choose an archive mode. The archive mode specifies how the archive program should process the performance data stored in the monitoring database.

Allow Performance Trace

Enables authorized users connected to this monitored system to start a specific performance trace.

Agent Heartbeat Interval (sec)

Indicates the interval at which agents that do not have any performance data to report or that are in standby mode connect to the monitor servlet to be notified of any configuration changes. If agents post data to the monitoring system before this interval expires, they will be notified of any configuration changes and the timer will be reset. The lower the number, the greater the responsiveness of the agents to configuration changes, but the greater the network traffic. This value cannot be set to zero.

PMU Timeout (days)

Indicates the interval in which an open PMU is considered timed out. PMU timeouts are measured in days. The PMU timeout value cannot be set to zero. After the specified interval, the system assumes that a PMU that has not completed encountered an error and should no longer be considered open. The reaper program moves timed out PMUs from the current PMU tables to the history PMU tables and sets the status to Reaper Timed Out.

Agent PMU Sample Rate (1/X)

Enables you to reduce the amount of PMU data generated by monitoring just 1 of every N server trips. Sampling does not affect PMUs initiated in a performance trace.

Agent Buffering Interval (sec)

Specifies the rate (in seconds) at which an agent sends performance data to the monitoring system. This value cannot be set to zero. A smaller interval decreases the delay between the time when the monitored system generates performance data and the time it is displayed on the monitoring system. Larger intervals enable more efficient transmission of performance data across the network because the system can consolidate the data into packets. The larger the interval, the greater the Agent Max Buffer size should be set.

Agent Max Buffer Size (bytes)

Determines the maximum size of the buffer containing performance data. This enables you to cap the amount of data being stored by the agent on the monitored system and the amount of data sent across your network. If this limit is reached, the agent automatically discards new performance data until the current data has been posted to the monitor servlet. The agent posts an alarm to the monitoring if the buffer size threshold is exceeded. The minimum must be 10240 bytes (10 KB).

Agent Filter Level

Select a filter level stored in the system default table applied to monitoring system when no specific definition is applied. Each PMU definition has a filter level set to one of the following: standard, verbose, or debug. Marking a PMU as standard, verbose, or debug will tell JD Edwards EnterpriseOne Performance Monitor system whether or not the PMU metrics should be captured based on what level JD Edwards EnterpriseOne Performance Monitor is capturing data at. JD Edwards EnterpriseOne Performance Monitor filter level can be one of the following: standby (no metric data captured), standard (only standard PMUs captured), verbose (standard and verbose PMUs captured), or debug (all PMUs captured).

4.4 Modifying System Definitions

Access the System Definitions - System Definition Revisions form.

A *system* refers to a particular monitored system. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system. The database name and GUID (a JD Edwards EnterpriseOne value used to uniquely identify a JD Edwards system) are provided by the agent during its registration process.

This section describes the properties and configuration options for each monitored system.

System ID

Identifies each monitored system. JD Edwards automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

Database Name

Identifies the name assigned to the database during installation, such as HPDEVORAP or HP9000. - - - FORM SPECIFIC HELP - - - The name of the database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

Description

Identifies the description for JD Edwards EnterpriseOne Performance Monitor usage. - - - FORM SPECIFIC HELP - - - A description of the monitoring system to assist recognition. The default value is the database name.

UID

Identifies the unique identifier for JD Edwards EnterpriseOne Performance Monitor defined systems. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

Archive Mode

Displays the archive mode you set specifies how the archive program should process the performance data stored in the monitoring database.

Allow User Trace

Enables authorized users connected to this monitored system to start a specific performance trace.

Agent Heartbeat Interval (sec)

Indicates the interval at which agents that do not have any performance data to report or that are in standby mode connect to the monitor servlet to be notified of any configuration changes. If agents post data to the monitoring system before this interval expires, they will be notified of any configuration changes and the timer will be reset. The lower the number, the greater the responsiveness of the agents to configuration changes, but the greater the network traffic. This value cannot be set to zero.

PMU Timeout (days)

Indicates the interval in which an open PMU is considered timed out. PMU timeouts are measured in days. The PMU timeout value cannot be set to zero. After the specified interval, the system assumes that a PMU that has not completed encountered an error and should no longer be considered open. The reaper program moves timed out PMUs from the current PMU tables to the history PMU tables and sets the status to Reaper Timed Out.

Agent Buffering Interval (sec)

Specifies the rate (in seconds) at which an agent sends performance data to the monitoring system.

This value cannot be set to zero.

A smaller interval decreases the delay between the time when the monitored system generates performance data and the time it is displayed on the monitoring system. Larger intervals enable more efficient transmission of performance data across the network because the system can consolidate the data into packets. The larger the interval, the greater the Agent Max Buffer size should be set.

Agent PMU Sample Rate (1/X)

Enables you to reduce the amount of PMU data generated by monitoring just 1 of every N server trips. Sampling does not affect PMUs initiated in a performance trace.

Agent Max Buffer Size (bytes)

Determines the maximum size of the buffer containing performance data. This enables you to cap the amount of data being stored by the agent on the monitored system and the amount of data sent across your network. If this limit is reached, the agent automatically discards new performance data until the current data has been posted to the monitor servlet. The agent posts an alarm to the monitoring if the buffer size threshold is exceeded. The minimum must be 10240 bytes (10 KB).

Last Updated User ID

Specifies the last user to update the record.

Last Updated Date/Time

Specifies the Utime when the record was last updated.

4.5 Viewing Agent Definitions

Access the Agent Definitions - Agent Definition Revisions form.

Agent definitions enable you to view the details about the agents in monitored systems.

System ID

Identifies each monitored system. JD Edwards EnterpriseOne automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

Database Name

Identifies the name of the database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

Agent ID

Identifies an agent within a domain. This is automatically generated by the monitor the first time an agent registers.

Domain Monitor

Displays as Yes or No. If yes, then this agent is responsible for sending resource events for its host and domain to the monitor at the specified sampling rate for the monitored system.

Domain Name

Identifies the name of the domain in which an agent operates.

Domain Type

Indicates whether the domain is an application server, a web server, or a Process Scheduler server domain.

Domain Host/Port

Identifies the name or IP address of the server on which the domain resides including the port number to which the domain listens for requests.

Domain Directory

Specifies the directory in which the domain is installed on the server.

Server Instance

Specifies the collator server instance number.

Agent Inactive

Select this box to make the agent inactive. If selected, the agent's events and PMUs do not appear in JD Edwards EnterpriseOne Performance Monitor forms showing current information. You can still view information about events and PMUs sent by inactive agents using the forms that display historical information.

4.6 Setting Agent Filter Options

Agent Filters - Agent Filter Revisions form.

Agent filters determine the amount of performance data that is generated and sent to the monitoring system. Depending on the situation, different levels of performance data may be needed to assist in your performance-related decisions. The levels range from no information to extremely detailed information.

Each type of PMU and event is associated with a filter level, which is the lowest level at which the system generates performance data for that PMU or event.

System ID

Identifies each monitored system. The JD Edwards EnterpriseOne system automatically generates this value incrementally. System definitions are created

automatically when the first agent of a monitored system registers with the monitoring system.

Database Name

Identifies the name assigned to the database during installation, such as HPDEVORAP or HP9000. - - - FORM SPECIFIC HELP - - - The name of the database running on the monitored system. The monitoring system automatically inserts this value when it recognizes and creates a monitored system.

Description

Identifies the description for JD Edwards EnterpriseOne Performance Monitor usage. - - - FORM SPECIFIC HELP - - - A description of the monitoring system to assist recognition. The default value is the database name

Last Update User ID

Specifies the last user to update the record.

Last Update Date/Time

Specifies the Utme when the record was last updated.

4.7 Scheduling the Reaper Program

The reaper process (R95900/XJDE0001) maintains the tables JD Edwards EnterpriseOne Performance Monitor uses to store performance data for current, real-time processing. It should be scheduled to run (via scheduler) periodically based on the filter level you are capturing data at and the load your enterprise is under. Performance of JD Edwards EnterpriseOne Performance Monitor will degrade if the tables JD Edwards EnterpriseOne Performance Monitor uses to store performance data for current, real-time processing are not maintained by the reaper process.

When JD Edwards EnterpriseOne Performance Monitor gets notified that a PMU has finished (it receives a STOP for an open PMU), it:

- Flags the corresponding start and update rows in the current PMU table F95960 for deletion.
- Inserts a row for the completed PMU in the F95961 table.

When the reaper program (R95900/XJDE0001) runs, it:

- Deletes all rows in current PMU table (F95960) that are flagged for deletion.
- Sets the status to timed out for expired PMUs in the current table.

Caution: If you do not schedule the reaper program to run often enough, the F95960 table will grow very large over time, and it may contain many old, open PMUs.

Monitoring System Performance

This chapter contains the following topics:

- [Section 5.1, "System JD Edwards EnterpriseOne Performance Monitoring"](#)
- [Section 5.2, "Viewing Open PMUs"](#)
- [Section 5.3, "Monitoring Individual User Performance"](#)
- [Section 5.4, "Viewing Open PMU Trees"](#)

5.1 System JD Edwards EnterpriseOne Performance Monitoring

This section contains an overview of monitoring system performance and lists the forms that are used to monitor system performance.

5.1.1 Monitoring System Performance

The activities that are related to monitoring system performance are primarily for viewing and analyzing the most recent performance data that is received from agents in a monitored system.

Note: The information that is presented in the System Performance forms is as current as the last form refresh.

Note: The standard deviation is a statistic that tells you how tightly all the values that are used to compute the average are clustered around the average. Large standard deviations warn that the averages appearing in the chart are not a reliable indicator of response times that are experienced by individual users.

5.1.2 Forms Used to Modify System Performance

Form Name	FormID	Navigation	Usage
System Performance - Work With System Performance	W95910J	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, System Monitor, System Performance	Search for performance data to analyze.
System Performance - System Performance	W95910E	From the System Performance - Work With System Performance form, select an item and click Select.	View performance indices and daily averages. You can also view open PMUs and completed PMUs by clicking the corresponding links.
Open PMUs - Open PMUs	W95910D	D Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, System Monitor, Open PMUs.	Search for open PMUs.
Open PMU Trees - Open PMU Trees	W95910G	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, System Monitor, Open PMU Trees.	Search for open PMU trees.
Open PMU Trees - Open PMU Trees Revision form.	W95910H	From the PMU Trees - Open PMU Trees form, select an open PMU Tree and click Select.	View and modify open PMU Trees.
User Level Trace	W95900M	From the Agent Filter Revisions form, click the Tools menu, then click User Level Trace.	Enter a user from whom you want to collect data.

5.2 Viewing Open PMUs

Access the Open PMUs - Open PMUs form.

An open PMU is a PMU that started but has not finished prior to an agent reporting performance metrics to the monitor. Information regarding open PMUs are stored in the F95960 table. When the PMU finishes, the collator flags the PMU for deletion and inserts a matching row into the F95961 table, which stores completed PMU data.

If an end user is reporting that a JD Edwards EnterpriseOne form is slow, use the Open PMU form to see where the request has stalled. Search for that user's ID to see the current state of that user's open PMUs.

Before you view current PMU details, you need to enter search criteria. If you do not specify criteria, the form displays *all* of the open PMUs for the selected system.

User ID

Specifies the code that identifies a user profile.

PMU Set

Categorizes meta-data transaction definitions into a set. Currently, only one definition set exists, and it is reserved for JD Edwards EnterpriseOne development.

PMU ID

Specifies the identifier for a particular PMU you can monitor.

Performance Trace

Identifies the name of the trace used to capture performance data that a user can start and stop to capture specific results.

Context 1, 2, 3

Specifies values (1-3) that are common to the entire user request or a specific tier. PMU metrics contain data specific to an individual PMU.

System ID

Identifies each monitored system. JD Edwards EnterpriseOne automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

Agent ID

Identifies an agent within a domain. This is automatically generated by the monitor the first time an agent registers.

Duration >=sec

Identifies the duration in milliseconds that a single transaction took to complete.

Duration < (sec)

Identifies the duration in milliseconds that a single transaction took to complete.

Note: Some of the metrics for various PMUs do not have values until the PMU finishes, so in some cases, metrics may appear with no values.

5.3 Monitoring Individual User Performance

You can monitor individual users' system performance using the User Level Trace application. You set the performance data according to the user override of JD Edwards EnterpriseOne Performance Monitoring level. You can override JD Edwards EnterpriseOne Performance Monitoring level for one user at a time, however you cannot override JD Edwards EnterpriseOne Performance Monitoring level for multiple users.

Access the User Level Trace form.

User ID

Specifies the code that identifies a user profile.

Standard, Verbose, Debug

Select a filter value to view only the information returned of a specific filter level. Each meta-data definition has a filter level set to one of the following: Standard, verbose, or debug.

5.4 Viewing Open PMU Trees

Access the Open PMU Trees - Open PMU Trees form.

Every user request generates a set of PMUs that you can display as a tree. The PMU with the top instance ID is the root of the tree and PMUs with no children are the leaves. A node represents each PMU or child PMU. Clicking a node reveals its detail, or the Select button on the parent nodes reveals its details.

A search may return multiple PMU trees. For example, if a user ID is shared across multiple users, or a single user has initiated multiple browser sessions, then more than one user request may be currently processing, and therefore more than one PMU tree open.

The data that is related to open PMUs is dynamic because the system is currently processing the PMU. Therefore, the composition of trees, and even their presence, is likely to change each time you click Refresh.

Note: For each PMU in the tree, the duration value represents the period of time that has elapsed since the monitoring system received the start timestamp for each PMU. In some cases, the display may indicate that a child PMU is "older" than its parent. For example, if the application server sent PMU information before the web server sent PMU information, the child PMUs that are running on the application server will display a smaller duration than the associated parent PMUs that are running on the web server.

User ID

Specifies the code that identifies a user profile.

Evaluating JD Edwards EnterpriseOne Performance Monitor Meta-Data

This chapter contains the following topics:

- [Section 6.1, "Understanding JD Edwards EnterpriseOne Performance Monitor Meta-Data Definitions"](#)
- [Section 6.2, "Revising Metric Definitions"](#)
- [Section 6.3, "Revising Event Definitions"](#)
- [Section 6.4, "Revising Context Definitions"](#)
- [Section 6.5, "Revising PMU Definitions"](#)

6.1 Understanding JD Edwards EnterpriseOne Performance Monitor Meta-Data Definitions

JD Edwards EnterpriseOne Performance Monitor uses these meta-data definitions:

- Metrics.
- Event Definitions.
- Contexts.
- PMUs.

6.1.1 Metrics

JD Edwards EnterpriseOne Performance Monitor agents send PMUs and events to the monitor. Each PMU and event contains one or more metrics. Each metric has a unique identifier. Metric definitions are the building blocks for creating PMUs and events. PMUs and events comprise up to six numeric values and one string metric value. One metric definition can appear in multiple event and PMU definitions.

6.1.2 Events

Events are not implemented for the 8.95 release of JD Edwards EnterpriseOne Performance Monitor. You can view and add them, however they are not functional.

Events are notifications containing performance metrics that are different from PMUs in that they are not hierarchical, and they do not have durations. JD Edwards EnterpriseOne has defined a set of event types, and each type of event is reported at a specific location in the instrumented code.

Each event has:

- Type.
- Instance identifier (a unique identifier for a specific event instance).
- Timestamp.
- Severity.
- Metrics (such as CPU usage and memory allocation).

Event definitions group as many as seven metrics to measure the intended performance data. Some events do not have metrics.

6.1.3 Contexts

A context definition applies only to PMU definitions. Contexts provide additional information so that PMU performance data can be displayed and searched more effectively. For example, a context definition enables you to group and sort numeric values, such as an execute count, by form name. Contexts enable the system to assign the data to various elements such as forms, components, service calls, applications, business functions, event rules, and so on. Without contexts, you have only numeric data in no understandable grouping.

PMU metrics contain data that is specific to that PMU. Context values, on the other hand, are common to the entire user request or a specific tier. The system uses contexts to "flatten" a PMU tree. For example, you do not have to navigate up from a SQL PMU to an event rule PMU to see what application generated that SQL statement.

6.1.4 PMUs

A PMU is a unit of measure that reflects the execution of a section of code. The system starts and stops a PMU at specific code locations, and the system may update a PMU anytime between the start and stop times. JD Edwards EnterpriseOne has defined a set of PMU types, and each type of PMU corresponds to the instrumentation at a specific code location, such as a SQL Execute in application or a business function call on the JD Edwards EnterpriseOne server.

Each PMU includes:

- PMU Type.
- Instance identifier (a unique identifier for a specific PMU instance).
- Start time.
- Stop time.
- Status.
- Metrics (such as number of SQL fetches).

A PMU represents a section of code that is bracketed by calls to an internal instrumentation API that signal the start and stop of that logical unit of code.

PMU definitions group as many as seven metrics to measure the intended performance data. Some PMUs do not have metrics.

6.1.5 Forms Used to Evaluate JD Edwards EnterpriseOne Performance Monitor Meta-Data

Form Name	FormID	Navigation	Usage
Metric Definitions - Work With Metric Definitions	W95905A	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Meta-Data, Metric Definitions	Search for metric definitions.
Metric Definitions - Metric Definition Revisions	W95905C	On Metric Definitions - Work With Metric Definitions, after retrieving metric definitions, select a metric definition you want to revise and click Select.	View and modify metric definitions and opt to display the metric value alone, or with its label.
Metric Definitions - Metric Reference	W95905E	On Metric Definitions - Metric Definition Revisions, click Metric References.	Review metric references.
Context Definitions - Work With Context Definitions	W95905D	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Meta-Data, Context Definitions	Search for Context Definitions.
Context Definitions - Context Definition Revisions	W95905F	On Context Definitions - Work With Context Definitions, after retrieving context definitions, select a context definition you want to revise and click Select.	View and modify context definitions and access the Context Definitions - Context Reference form.
Context Definitions - Context Reference	W95905E	On the Context Definitions - Context Definition Revisions form, click Context Reference.	View reference data for Context Definitions.

Form Name	FormID	Navigation	Usage
PMU Definitions - Work With PMU Definitions	W95905J	D Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Meta-Data, PMU Definitions	Search for and select PMU Definitions.
PMU Definitions - PMU Definition Revisions	W95905G	On PMU Definitions - Work With PMU Definitions, after retrieving PMU definitions, select a PMU definition you want to revise and click Select.	View and revise PMU Definitions.
Event Definitions - Work With Event Definitions	W95905K	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, Administration, Meta-Data, Event Definitions	Search for and select Event Definitions.
Event Definitions - Event Definition Revisions	W95905M	On Event Definitions - Work With Event Definitions, after retrieving Event definitions, select an Event definition and click Select.	View and revise Event Definitions.

6.2 Revising Metric Definitions

This section discusses the attributes of a metric definition.

Access the Metric Definitions - Metric Definition Revisions form.

Note: Only JD Edwards EnterpriseOne should modify metric definitions. Modifying metric definitions at your site could cause unexpected results.

Metric ID

Identifies Metric meta-data definitions.

Metric Type

Indicates the type of metric meta-data definition (Counter, Gauge, NumID, StrID).

- Counter: A counter metric is designed to enable sums of values from a specific time range to be calculated.

Examples are bytes printed and records written. The values can also be averaged, maximums and minimums can be calculated, and other kinds of statistical calculations can be performed

- Gauge: A gauge metric is designed to be used instead of a counter when it is not meaningful to calculate sum values that are recorded within a time range.

Calculations that are performed on gauge metrics include: average, standard deviation, median, maximum value, and minimum value. For example, the amount of memory that is used on a server is a gauge metric type. If you measure the amount of memory that is used over 20 transactions in a time range, the sum of the memory that is used is not necessarily useful. However, the average, median, and standard deviation provide insight into usage per transaction.

- Numeric Identifier: A numeric identifier is a numeric value that is used as an identifier, not as a measurement value.

Creating sums and averages, or manipulating these values in any arithmetic way is not meaningful. For example, message numbers and error codes are numeric identifier metric types.

- String: Used with metric definition attributes that need to be represented as text, not a numeric value.

Arithmetic operations are not performed on string metric types. For example, descriptive attributes, such as site path, file name, and so on, are string metric types.

Metric Label

Identifies the description for JD Edwards EnterpriseOne Performance Monitor usage. - - - FORM SPECIFIC HELP - - - Displays a more detailed description of the purpose of the metric.

Description

Identifies the description for JD Edwards EnterpriseOne Performance Monitor usage. - - - FORM SPECIFIC HELP - - - Displays a more detailed description of the purpose of the metric.

Metric Multiplier

Identifies the multiplier used for metric definition meta-data.

Display Metric as an Integer

Indicates where metric meta-data definition is an integer or double value. This is a Yes/No flag.

Metric Display

Enables you to determine what appears on the forms displaying a metric. These options are mutually exclusive. Select Display Value to display the actual value. Select Display Label to display the label describing the metric value. If using Display Label select the UDC you wish to use as Display Labels.

Metric References

Calls References (W95905E). Shows all the places this metric is used within PMU and Event Definitions.

6.3 Revising Event Definitions

This section discusses the attributes of an Event definition. Events are not implemented for the 8.95 release of JD Edwards EnterpriseOne Performance Monitor. You can view and add them, however they are not functional.

Access the Event Definitions - Event Definition Revisions form.

Event Definition Set

Categorizes meta-data event definitions into a set. Currently, only one definition set exists, and it is reserved for JD Edwards EnterpriseOne development.

Set Description

Describes a definition set.

Event ID

Identifies a particular event you can monitor.

Event Label

Identifies the label for Event meta-data definitions.

Event Description

Describes JD Edwards EnterpriseOne Performance Monitor usage. - - - FORM
SPECIFIC HELP - - - Displays a more detailed description of the purpose of the Event.

Additional Label

Identifies the label for PMU additional data.

Standard, Verbose, Debug

Select a filter value to view only the information returned of a specific filter level. Each meta-data definition has a filter level set to one of the following: Standard, verbose, or debug.

6.3.1 Context Tab

Context ID 1–3

Identifies context values 1–3.

Display Content 1–3

Flag to indicate whether or not context value 1 should be displayed when showing PMU data.

6.3.2 Metric Tab

Metric ID 1–7

Identifies meta-data for definitions 1–7 used for giving context to metric values.

Display Metric 1–7

Flag to indicates whether or not context value 1 should be displayed when showing PMU data.

6.4 Revising Context Definitions

This section discusses the attributes of a context definition.

Access the Context Definitions - Context Definition Revisions form.

Context Identifier

Identifies Context meta-data definitions.

Context Label

Identifies the context label appears on any form that displays a context to describe the value.

Description

Displays a more detailed description of the purpose of the context.

Context Reference

Calls References (W95905E). Shows all the places this metric is used within PMU and Event Definitions

6.5 Revising PMU Definitions

This section describes the attributes of PMU definition.

Access the PMU Definitions - PMU Definition Revisions form.

PMU Definition Set

Categorizes metadata event definitions into a set. Currently, only one definition set exists, and it is reserved for JD Edwards EnterpriseOne development.

Set Description

Describes a definition set.

PMU ID

Identifies a particular PMU that you can monitor.

PMU Label

Identifies the label for PMU additional data.

PMU Description

Describes a PMU metadata definition.

Additional Data Label

Identifies the label for PMU additional data.

Standard, Verbose, Debug

Select a filter value to view only the information returned of a specific filter level. Each meta-data definition has a filter level set to one of the following: Standard, verbose, or debug.

Contexts 1, 2, 3

Identifier to context value 1 through 3.

Display Content 1, 2, 3

Flag to indicate whether or not context value should be displayed when showing PMU data.

Metric ID 1, 2, 3, 4, 5, 6, 7

Identifies metric values 1, 2, 3, 4, 5, 6, 7.

Display Metric ID

Flag to indicate whether or not metric value should be displayed when showing PMU data.

Analyzing Historical Performance Data

This chapter contains the following topics:

- [Section 7.1, "Historical Performance Data"](#)
- [Section 7.2, "Viewing Completed PMUs"](#)

7.1 Historical Performance Data

This section provides an overview of historical performance data and lists the forms used to analyze historical performance data.

7.1.1 Historical Performance Data

The system stores historical data from the moment a PMU finishes processing or an event occurs. When the archive program runs (R95900/XJDE0002), it either deletes the historical data or moves it to the archive tables.

Analyzing historical data can help you:

- Identify trends.

By comparing historical data, you can spot upward and downward performance trends.

- Investigate past user complaints.

For example, a user may complain that performance was slow during the previous week. You search historical performance data to find the PMUs that were generated by the user at a specific time a week ago.

7.1.2 Forms Used to Analyze Historical Performance Data

Form Name	FormID	Navigation	Usage
Completed PMUs - Completed PMUs	W95920D	JD Edwards EnterpriseOne Menus, JD Edwards EnterpriseOne Lifecycle Tools, System Administration Tools, JD Edwards EnterpriseOne Performance Monitor, History, Completed PMUs	Enables you to search for and display completed PMUs.

7.2 Viewing Completed PMUs

Access the Completed PMUs - Completed PMUs form.

7.2.1 Entering Search Criteria

Before you can view information about completed PMUs, you need to enter criteria so that you can locate the appropriate completed PMUs.

User ID

Specifies the code that identifies a user profile.

PMU Set

Categorizes meta-data transaction definitions into a set. Currently, only one definition set exists, and it is reserved for JD Edwards EnterpriseOne development.

PMU ID

Specifies the identifier for a particular PMU you can monitor.

Performance Trace

Identifies the name of the trace used to capture performance data that a user can start and stop to capture specific results.

Context 1,2,3

Specifies values (1-3) that are common to the entire user request or a specific tier. PMU metrics contain data specific to an individual PMU.

System ID

Identifies each monitored system. JD Edwards EnterpriseOne automatically generates this value incrementally. System definitions are created automatically when the first agent of a monitored system registers with the monitoring system.

Agent ID

Identifies an agent within a domain. This is automatically generated by the monitor the first time an agent registers.

Duration >= (ms)

Identifies the duration in milliseconds that a single transaction took to complete.

Duration < (ms)

Identifies the duration in milliseconds that a single transaction took to complete.

From Date/Time

Identifies the date and time the monitor started the request. This is related to the duration of a PMU.

To Date/Time

Identifies the date and time the monitor started the request. This is related to the duration of a PMU.

7.2.2 Working with Completed PMUs

At the bottom of the Completed PMUs - Completed PMUs form, the grid contains the information that is related to the PMUs that are retrieved by your search criteria.

You can access the Completed PMUs - PMU Details - C Call Object details form from a variety of JD Edwards EnterpriseOne Performance Monitor forms, such as the Completed PMUs - Completed PMUs form and the Open PMUs - Open PMUs form.

You can also access it readily from trees. To access the form, you click the link that is created from the PMU name.

You can access the PMU details form from a variety of JD Edwards EnterpriseOne Performance Monitor forms, such as the Completed PMUs form and the Open PMUs form. You can also access it readily from trees. To access the form, you click the link that is created from the PMU name.

The PMU Details form indicates:

- PMU ID and the PMU definition set to which it belongs.
- Identification information.
- Durations information, including metric values.
- Context information.
- Additional data.

The information that appears on the PMU Details form varies depending on the type of PMU that is displayed. All metrics are formatted according to information from the metric definition.

7.2.2.1 Completed PMUs

PMU

Describes a unit of measure that reflects the execution of a section of code.

Duration (ms)

Identifies the duration in milliseconds that a single transaction took to complete.

Tree View

Displays completed PMUs in a tree view. Useful to show the hierarchy of nested PMUs. Node description will be the concatenated PMU Label (alternate language if applicable) and PMU duration. Only PMUs that are completed will show. Only top level parents will be displayed at the first level under the root. Children of the parents can be expanded. The PMU selected on the calling form will be set to bold in the tree. The PMU will be shown in the hierarchical context it was created from. Therefore, the selected PMU can be at the parent level or the child level. All nodes can be selected and PMU Detail (W95920F) will be called to show all available information.

PMU Status

Specifies the status the PMU of the PMU.

User ID

Specifies the code that identifies a user profile.

Monitor Start

Identifies the date and time the monitor started the request. This is related to the duration of a PMU.

Context 1, 2, 3

Specifies values (1-3) that are common to the entire user request or a specific tier. PMU metrics contain data specific to an individual PMU.

Agent ID

Identifies an agent within a domain. This is automatically generated by the monitor the first time an agent registers with it.

Performance Trace

Identifies the name of the trace used to capture performance data that a user can start and stop to capture specific results.

7.2.2.2 Context

The Context tab reveals the context definition and value that is associated with each PMU in the grid. To view the details of the contexts in a PMU, examine the context and PMU definitions.

This Description column shows data for the additional description value that is part of the incoming PMU. The description stores large character values and generally stores information that exceeds the 128-character limit of Metric 7 and the 254-character limit of the contexts.

7.2.2.3 Metrics

The Metrics tab presents the metric values for all of the metrics that are defined for each PMU in the grid.

Metric information appears in a raw format. The values reflect exactly what is stored in the database.

7.2.2.4 Additional Data

The Additional Data tab enables you to view textual information about the data being captured by the PMU.

7.2.2.5 Agent Details

The Agent details tab presents information about the agent from which the PMU was executed.

7.2.2.6 Identification

The Identification tab enables you to view specific identifications associated with the PMUs.

7.2.3 Completed PMU Status

A completed PMU can be assigned one of these statuses:

Status	Description
Success	The PMU finished successfully.
Failed	The PMU finished unsuccessfully.
Abort	Instrumentation code terminated a PMU due to an unexpected condition.
Unknown	Assigned in situations when an agent closes a child PMU if the top-level PMU closes out of sequence.
Timeout	Instrumentation code timed out the PMU.
Reaper Timeout	Long running PMU closed by the Reaper.
Admin Timeout	An administrator manually closed the PMU. This is <i>not</i> currently implemented.

Understanding JD Edwards EnterpriseOne Performance Monitor Architecture

This appendix contains the following topics:

- [Section A.1, "JD Edwards EnterpriseOne Performance Monitor Architecture Components"](#)

A.1 JD Edwards EnterpriseOne Performance Monitor Architecture Components

This section discusses the major components of JD Edwards EnterpriseOne Performance Monitor Architecture, which are:

- JD Edwards EnterpriseOne application server.
- JD Edwards EnterpriseOne web server (also known as JAS, Java application server)
- JD Edwards EnterpriseOne monitoring server
- User interface
- Database host
- Data Flow

A.1.1 JD Edwards EnterpriseOne Application Server

This section discusses the components of the application server.

A.1.1.1 AppServer C-Agent

All instrumented C processes on the JD Edwards EnterpriseOne application server are linked to the JD Edwards EnterpriseOne PerfMon DLL to call into instrumentation APIs. Instrumentation APIs send performance data to the PerfMon Java Adapter on the Monitoring Server using JD Edwards EnterpriseOne Proprietary JDENT communication protocol.

A.1.1.2 Archive Batch Process

This component is a batch process scheduled to periodically move data from current and history table to archive tables and also can be configured to remove data from all tables to manage disk space. This process uses JD Edwards EnterpriseOne proprietary JDB APIs to access and modify database tables.

A.1.1.3 Reaper Batch Process

This component is a batch process scheduled to periodically move data from current tables to history tables for completed transactions and also to delete flagged rows in current tables. This process uses JD Edwards EnterpriseOne proprietary JDB APIs to access and modify database tables.

A.1.2 JD Edwards EnterpriseOne JAVA Application Server

This section discusses the components of the JAS server.

A.1.2.1 JAS Runtime Engine

This component is used to display HTML pages for JD Edwards EnterpriseOne PerfMon User Interface. Retrieves meta-data from database and creates HTML Pages to send back to user browser over HTTP. It also changes Agent configuration in the database and sends a notification to the PPMI Servlet so the changes can get to the Agents. Jas runtime engine uses JD Edwards EnterpriseOne proprietary JDBj APIs to access and modify database tables.

A.1.3 JD Edwards EnterpriseOne Monitoring Server

Enterprise One Monitoring Server uses the same components as the Enterprise solution which are: PPMI Servlet, Monitor Servlet and Data Collator. Changes have been made to these components for integration with JD Edwards EnterpriseOne environment. For example these components have been changed to use JD Edwards EnterpriseOne JDBj mechanism for database operations. No changes have been made in the way 3rd part solutions will register and retrieve data from the Monitoring Server. One new component (JD Edwards EnterpriseOne PerfMon Java Adapter) has been added to act as a bridge between JD Edwards EnterpriseOne application server and Monitor Server. This adapter receives the performance data from the Enterprise Server (through JDENet messages) and converts them to HTTP and sends them to the Monitor Servlet by calling into existing PerfMon Java APIs.

A.1.3.1 PerfMon Java Adapter

PerfMon Java Adapter acts as a bridge between JD Edwards EnterpriseOne application server and the Monitoring Server. C processes on the application Server (also known as C-Agents) send the performance data to the PerfMon Java Adapter using the JD Edwards EnterpriseOne JDENet protocol. PerfMon Java Adapter calls into Java APIs and passes the performance data to the monitor Servlet which runs on the Monitor Server. Note that the Java APIs called by the PerfMon Java Adapter are the same APIs that are called in JAS Runtime Engine instrumented code.

A.1.3.2 Monitor Servlet

The Monitor Servlet receives the registration notification from all the Agents and updates the database with the registration information. It also receives the performance data from all the Agents in the environment and hands the data to all registered PPMI clients. In JD Edwards EnterpriseOne solution the "out of the box" PPMI client for performance data is the JD Edwards EnterpriseOne data collator which is another component of the Monitor Server. Each time the Monitor Server receives the performance data from an Agent, it sends back the Agent configuration as the response to the Agents. This is how the Agents will be notified from any Agent configuration changes. The Monitor Servlet uses the JD Edwards EnterpriseOne proprietary JDBj APIs to access and modify database tables.

A.1.3.3 Data Collator Servlet

The Data Collator registers itself to the PPMI Servlet as a PPMI Client Proxy to show interest in receiving performance data for JD Edwards EnterpriseOne environment. By receiving performance data from the Monitor Servlet, the data collator will update the database accordingly: If the data is indicating the end of an open transaction, then the data collator will add the data to the history table and remove all the corresponding data for this transaction from the current table. Otherwise the data is added to the current table. The Data Collator uses the JD Edwards EnterpriseOne proprietary JDBj APIs to access and modify database tables.

A.1.3.4 PPMI Servlet

The PPMI Servlet manages PPMI clients. PPMI Client is a component that registers to the PPMI Servlet to receive JD Edwards EnterpriseOne performance data. When a client registers, the PPMI Servlet will notify the Monitor Servlet of the newly registered client and the Monitor Servlet will make sure that the performance data will be sent to the client. In JD Edwards EnterpriseOne Performance Monitor architecture, the JD Edwards EnterpriseOne Data Collator is the default PPMI client for the system. The PPMI Servlet also receives the Agent configuration change messages from the JD Edwards EnterpriseOne JAS Server. This happens when a JD Edwards EnterpriseOne administrator changes the configuration for a group of Agents using the JD Edwards EnterpriseOne PerfMon User Interface. By receiving the notification, the PPMI Servlet reads the new Agent configuration from the database and updates the Agent configuration in the memory. This configuration in the memory is the one that is sent back to all the Agents by the Monitor Servlet. The PPMI Servlet uses the JD Edwards EnterpriseOne proprietary JDBj APIs to access and modify database tables.

A.1.4 Data Flow

This section describes how data flows through JD Edwards EnterpriseOne Performance Monitor.

A.1.4.1 Sending Performance Data to Monitor Server

Arrows 1

On the JD Edwards EnterpriseOne application server the C-Agent calls into C instrumentation API to send the performance data to the JD Edwards EnterpriseOne PerfMon Java Adapter on the Monitoring Server.

Arrow 2

On the Monitoring Server the Java Adapter calls into Java Agent APIs to send the performance data to the Monitor Servlet component of the Monitoring Server. The Java Agent APIs use a separate thread per Agent to communicate with the Monitoring Server.

Arrow 3

The JAS Runtime Engine calls into Java Agent APIs to send the performance data to the Monitor Servlet component of the Monitoring Server. The Java Agent APIs use a separate thread per Agent to communicate with the Monitoring Server.

A.1.4.2 Saving performance Data in database

Arrow 4

Inside the Monitor Server, the Monitor Servlet receives the performance data from Java Agents in HTTP format and converts the data to XML before sending it to the Data

Collator. The data collator component of the Monitor Server calls into JD Edwards EnterpriseOne JDBj APIs to save performance data to the JD Edwards EnterpriseOne PerfMon database.

A.1.4.3 Sending configuration changes to the Agents

Arrows 5,6,7,8 and 9

The administrator uses the JD Edwards EnterpriseOne PerfMon User Interface to change Agent configuration for a group of Agents. (arrow number 5).

The JAS Runtime engine changes the Agent configuration data in JD Edwards EnterpriseOne PerfMon database (arrow number 6).

The JAS Runtime engine sends a notification message (in HTTP format) to the PPMI Servlet component of the Monitor Server (Arrow number 7).

The PPMI Servlet reads the new Agent configuration from PerfMon database (arrow number 8).

The PPMI Server updates the Agent configuration data in the memory, where it will be accessed by the Monitor Servlet. The Monitor Servlets sends the new Agent configuration to all the Agents the next time it communicates with the Agents (arrow number 19).

On the JAS Server no additional work is needed. The PerfMon Java Adaptor sends a JDENet message containing the new Agent configuration to all the C-Agents processes on the AppServer (arrow number 10). The Java Adapter manages a C-Agent process table where it keeps the host name and port number (among others) of all the C-Agent processes registered to the Java Adapter.

A.1.4.4 Archiving data in database

Arrow 11

A scheduled batch process will archive data in the database. The Archive batch process can be scheduled to run periodically to either move the performance data from database tables to the Archive tables or to remove them from Archive tables to free disk space on the database host. This batch process uses the EnterpriseOne proprietary JDB APIs to access data in the database.

A.1.4.5 Reaping data in database

Arrow 15

A scheduled batch process will move timed out data from current tables to history tables. This batch process uses the EnterpriseOne proprietary JDB APIs to access data in the database.

B

PMU Definition Reference

This appendix contains the following topics:

- [Section B.1, "PMU 100 C Call Object"](#)
- [Section B.2, "PMU 101 Java Call Object"](#)
- [Section B.3, "PMU 200 JDB/JDBj DML APIs"](#)
- [Section B.4, "PMU 201 SQL DML APIs"](#)
- [Section B.5, "PMU 300 Execute Form"](#)
- [Section B.6, "PMU 301 Event Rules"](#)
- [Section B.7, "PMU 302 Fetch Process"](#)
- [Section B.8, "PMU 303 OK Process"](#)
- [Section B.9, "PMU 304 Asynch OK Process"](#)

Note: Currently, only one PMU set definition exists; set 2.

B.1 PMU 100 C Call Object

PMU 100 has these identification attributes.

Attribute	Value
ID	100
Label	C Call Object
Description	Measure total execution time when jdeCallObject is called.

B.1.1 Filter Level

PMU 100 has a filter level of Standard.

B.1.2 Contexts

Context	Value	Description
1	BSFN Information	Application/Form/BSFN/BSFN Version/Previous BSN

Context	Value	Description
2	Login Information	Host/Port/User/Environment/Role
3	Other Information	Module/BSFN Library/PID/Thread

B.1.3 Metrics

Metric	Value	Description
1	BSFN Level	Business Function Level in jdeCallObject
2	BLC Specs Retrieval Time in ms	Total retrieval time for BLC specs
3	DSTemplate Spec Retrieval Time	Total retrieval time for DSTemplate specs
4	Call BSNF Retry	Number Of Retry to Call BSNF
5	Not Used	Not Used
6	Not Used	Not Used
7	Additional Information	BSFN Author/Override location/Commit Mode

B.1.4 Additional Label

Call Object Error/Server Status.

B.2 PMU 101 Java Call Object

PMU 101 has these identification attributes.

Attribute	Value
ID	101
Label	Java Call Object
Description	Java Call Object Execution Time

B.2.1 Filter Level

PMU 101 has a filter level of Standard.

B.2.2 Contexts

Context	Value	Description
1	BSFN Information	Application/Form/BSFN/BSFN Version/Previous BSNF
2	Login Information	Host/Port/User/Environment/Role
3	Other Information	Module/BSFN Library/PID/Thread

B.2.3 Metrics

Metric	Value	Description
1	BSFN DS Flattening Time	Java Call Object BSNF DS Flattening Time
2	BSFN Inflating Time	Java Call Object BSNF DS Inflating Time
3	BSFN Plug-in Execution Time	Total Execution Time For Business Function Plug-in
4	Serialized Object Time	Retrieval time for DSTMPL & JDEBLC serialized object time
5	Retry	Total Number Of Retry
6	Additional Information	BSFN Author/Override location/Commit Mode
7	Not Used	Not Used

B.2.4 Additional Label

CallObject/BSFN Error List

B.3 PMU 200 JDB/JDBj DML APIs

PMU 200 has these identification attributes.

Attribute	Value
ID	200
Label	JDB/JDBj DML APIs
Description	E1PM_JDB_DML_DEBUG

B.3.1 Filter Level

PMU 102 has a filter level of Debug.

B.3.2 Contexts

Context	Value	Description
1	Object	Object Name
2	Data Source	Data Source Name

B.3.3 Metrics

Metric	Value	Description
1	DB Request	DB request Type
2	Commit Mode	DB transaction mode
3	Return Code	Function return code

Metric	Value	Description
4	Number of Records	Number of Returned Records (Commit/Rollback Excluded)
5	Not Used	Not Used
6	Not Used	Not Used
7	Transaction ID	Transaction ID

B.3.4 Additional Label

JDB/JDBj

B.4 PMU 201 SQL DML APIs

PMU 201 has these identification attributes.

Attribute	Value
ID	201
Label	SQL DML APIs
Description	E1PM_SQL_DML_DEBUG

B.4.1 Filter Level

PMU 106 has a filter level of Debug.

B.4.2 Contexts

Context	Value	Description
1	Database Name	Database Name
2	Host Name	Database Host Name
3	Database Type	Database Type

B.4.3 Metrics

Metric	Value	Description
1	Transaction Mode	Size of HTML Response in bytes.
2	Not Used	Not Used
3	Not Used	Not Used
4	Not Used	Not Used
5	Not Used	Not Used
6	Not Used	Not Used
7	Transaction ID	Transaction ID

B.4.4 Additional Label

JDB/JDBj

B.5 PMU 300 Execute Form

PMU 108 has these identification attributes.

Attribute	Value
ID	300
Label	Execute Form
Description	User Session logout, expiration, timeout, or error

B.5.1 Filter Level

PMU 300 has a filter level of Standard.

B.5.2 Contexts

Context	Value	Description
1	User/Environment/Role	User, Environment, and Role
2	Application Name	Application Name
3	Form Name	Form Name

B.5.3 Metrics

Metric	Value	Description
1	Reason	Reason for session termination.
2	Response Code	HTTP Response Code.
7	User ID	User ID of user logging on.

B.5.4 Additional Label

N/A

B.6 PMU 301 Event Rules

PMU 301 has these identification attributes.

Attribute	Value
ID	301
Label	Event Rules
Description	Event Rules

B.6.1 Filter Level

PMU 109 has a filter level of Verbose.

B.6.2 Contexts

Context	Value	Description
1	Event Name	Web server Session ID.
2	Object ID	Client IP Address.

B.6.3 Metrics

Metric	Value	Description
1	Grid Row	Grid Row
2	Grid Column	Grid Column

B.6.4 Additional Label

N/A

B.7 PMU 302 Fetch Process

PMU 113 has these identification attributes.

Attribute	Value
ID	302
Label	Fetch Process
Description	Fetch Process

B.7.1 Filter Level

PMU 113 has a filter level of Verbose.

B.7.2 Contexts

Context	Value	Description
1	Business View	Business View

B.7.3 Metrics

Metric	Value	Description
1	Bypass Signon?	Boolean - logging in with a guest ID?

B.7.4 Additional Label

N/A

B.8 PMU 303 OK Process

PMU 303 has these identification attributes.

Attribute	Value
ID	303
Label	OK Button
Description	OK Button Process

B.8.1 Filter Level

PMU 114 has a filter level of Standard.

B.8.2 Contexts

Context	Value	Description
1	TP Usage	Transaction Processing Usage

B.8.3 Metrics

Metric	Value	Description
0	Not Used	Not Used

B.8.4 Additional Label

N/A

B.9 PMU 304 Asynch OK Process

PMU 304 has these identification attributes.

Attribute	Value
ID	304
Label	Asynch OK Process
Description	Asynch OK Button Process

B.9.1 Filter Level

PMU 304 has a filter level of Standard.

B.9.2 Contexts

Context	Value	Description
1	User/Environment/Role	User, Environment, and Role
2	Application Name	Application Name
3	Form Name	Form Name

B.9.3 Metrics

Metric	Value	Description
1	Queue Elements	Queue Elements
2	Not Used	Not Used
3	Not Used	Not Used
4	Not Used	Not Used
5	Not Used	Not Used
6	Not Used	Not Used
7	TP Usage	Transaction Processing Usage

B.9.4 Additional Label

N/A

Glossary

Accessor Methods/Assessors

Java methods to “get” and “set” the elements of a value object or other source file.

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

Software that provides the business logic for an application program in a distributed environment. The servers can be Oracle Application Server (OAS) or WebSphere Application Server (WAS).

Auto Commit Transaction

A database connection through which all database operations are immediately written to the database.

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

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.

best practices

Non-mandatory guidelines that help the developer make better design decisions.

BPEL

Abbreviation for Business Process Execution Language, a standard web services orchestration language, which enables you to assemble discrete services into an end-to-end process flow.

BPEL PM

Abbreviation for Business Process Execution Language Process Manager, a comprehensive infrastructure for creating, deploying, and managing BPEL business processes.

Build Configuration File

Configurable settings in a text file that are used by a build program to generate ANT scripts. ANT is a software tool used for automating build processes. These scripts build published business services.

build engineer

An actor that is responsible for building, mastering, and packaging artifacts. Some build engineers are responsible for building application artifacts, and some are responsible for building foundation artifacts.

Build Program

A WIN32 executable that reads build configuration files and generates an ANT script for building published business services.

business analyst

An actor that determines if and why an EnterpriseOne business service needs to be developed.

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 service

EnterpriseOne business logic written in Java. A business service is a collection of one or more artifacts. Unless specified otherwise, a business service implies both a published business service and business service.

business service artifacts

Source files, descriptors, and so on that are managed for business service development and are needed for the business service build process.

business service class method

A method that accesses resources provided by the business service framework.

business service configuration files

Configuration files include, but are not limited to, interop.ini, JDBj.ini, and jdlog.properties.

business service cross reference

A key and value data pair used during orchestration. Collectively refers to both the code and the key cross reference in the WSG/XPI based system.

business service cross-reference utilities

Utility services installed in a BPEL/ESB environment that are used to access JD Edwards EnterpriseOne orchestration cross-reference data.

business service development environment

A framework needed by an integration developer to develop and manage business services.

business services development tool

Otherwise known as JDeveloper.

business service EnterpriseOne object

A collection of artifacts managed by EnterpriseOne LCM tools. Named and represented within EnterpriseOne LCM similarly to other EnterpriseOne objects like tables, views, forms, and so on.

business service framework

Parts of the business service foundation that are specifically for supporting business service development.

business service payload

An object that is passed between an enterprise server and a business services server. The business service payload contains the input to the business service when passed to the business services server. The business service payload contains the results from the business service when passed to the Enterprise Server. In the case of notifications, the return business service payload contains the acknowledgement.

business service property

Key value data pairs used to control the behavior or functionality of business services.

Business Service Property Admin Tool

An EnterpriseOne application for developers and administrators to manage business service property records.

business service property business service group

A classification for business service property at the business service level. This is generally a business service name. A business service level contains one or more business service property groups. Each business service property group may contain zero or more business service property records.

business service property key

A unique name that identifies the business service property globally in the system.

business service property utilities

A utility API used in business service development to access EnterpriseOne business service property data.

business service property value

A value for a business service property.

business service repository

A source management system, for example ClearCase, where business service artifacts and build files are stored. Or, a physical directory in network.

business services server

The physical machine where the business services are located. Business services are run on an application server instance.

business services source file or business service class

One type of business service artifact. A text file with the .java file type written to be compiled by a Java compiler.

business service value object template

The structural representation of a business service value object used in a C-business function.

Business Service Value Object Template Utility

A utility used to create a business service value object template from a business service value object.

business services server artifact

The object to be deployed to the business services server.

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.

check-in repository

A repository for developers to check in and check out business service artifacts. There are multiple check-in repositories. Each can be used for a different purpose (for example, development, production, testing, and so on).

checksum

A fixed-size datum computed from an arbitrary block of digital data for the purpose of detecting accidental errors that may have been introduced during its transmission or storage. JD Edwards EnterpriseOne uses the checksum to verify the integrity of packages that have been downloaded by recomputing the checksum of the downloaded package and comparing it with the checksum of the original package. The procedure that yields the checksum from the data is called a checksum function or checksum algorithm. JD Edwards EnterpriseOne uses the MD5 and STA-1 checksum algorithms.

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.

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.

correlation data

The data used to tie HTTP responses with requests that consist of business service name and method.

credentials

A valid set of JD Edwards EnterpriseOne username/password/environment/role, EnterpriseOne session, or EnterpriseOne token.

cross-reference utility services

Utility services installed in a BPEL/ESB environment that access EnterpriseOne cross-reference data.

database credentials

A valid database username/password.

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.

deployment artifacts

Artifacts that are needed for the deployment process, such as servers, ports, and such.

deployment server

A server that is used to install, maintain, and distribute software to one or more enterprise servers and client workstations.

direct connect

A transaction method in which a client application communicates interactively and directly with a server application.

See also batch-of-one and store-and-forward.

Do Not Translate (DNT)

A type of data source that must exist on the iSeries because of BLOB restrictions.

embedded application server instance

An OC4J instance started by and running wholly within JDeveloper.

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.

Enterprise Service Bus (ESB)

Middleware infrastructure products or technologies based on web services standards that enable a service-oriented architecture using an event-driven and XML-based messaging framework (the bus).

EnterpriseOne administrator

An actor responsible for the EnterpriseOne administration system.

EnterpriseOne credentials

A user ID, password, environment, and role used to validate a user of EnterpriseOne.

EnterpriseOne development client

Historically called “fat client,” a collection of installed EnterpriseOne components required to develop EnterpriseOne artifacts, including the Microsoft Windows client and design tools.

EnterpriseOne extension

A JDeveloper component (plug-in) specific to EnterpriseOne. A JDeveloper wizard is a specific example of an extension.

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.

EnterpriseOne resource

Any EnterpriseOne table, metadata, business function, dictionary information, or other information restricted to authorized users.

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.

explicit transaction

Transaction used by a business service developer to explicitly control the type (auto or manual) and the scope of transaction boundaries within a business service.

exposed method or value object

Published business service source files or parts of published business service source files that are part of the published interface. These are part of the contract with the customer.

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.

foundation

A framework that must be accessible for execution of business services at runtime. This includes, but is not limited to, the Java Connector and JDBj.

FTP server

A server that responds to requests for files via file transfer protocol.

HTTP Adapter

A generic set of services that are used to do the basic HTTP operations, such as GET, POST, PUT, DELETE, TRACE, HEAD, and OPTIONS with the provided URL.

instantiate

A Java term meaning “to create.” When a class is instantiated, a new instance is created.

integration developer

The user of the system who develops, runs, and debugs the EnterpriseOne business services. The integration developer uses the EnterpriseOne business services to develop these components.

integration point (IP)

The business logic in previous implementations of EnterpriseOne that exposes a document level interface. This type of logic used to be called XBPs. In EnterpriseOne 8.11, IPs are implemented in Web Services Gateway powered by webMethods.

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.

interface table

See Z table.

internal method or value object

Business service source files or parts of business service source files that are not part of the published interface. These could be private or protected methods. These could be value objects not used in published methods.

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.

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.

JDeveloper Project

An artifact that JDeveloper uses to categorize and compile source files.

JDeveloper Workspace

An artifact that JDeveloper uses to organize project files. It contains one or more project files.

JMS Queue

A Java Messaging service queue used for point-to-point messaging.

listener service

A listener that listens for XML messages over HTTP.

local repository

A developer's local development environment that is used to store business service artifacts.

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.

Manual Commit transaction

A database connection where all database operations delay writing to the database until a call to commit is made.

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.

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.

Monitoring Application

An EnterpriseOne tool provided for an administrator to get statistical information for various EnterpriseOne servers, reset statistics, and set notifications.

named event rule (NER)

Encapsulated, reusable business logic created using event rules, rather than 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.

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-incapabilities 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.

Pathcode Directory

The specific portion of the file system on the EnterpriseOne development client where EnterpriseOne development artifacts are stored.

patterns

General repeatable solutions to a commonly occurring problem in software design. For business service development, the focus is on the object relationships and interactions.

For orchestrations, the focus is on the integration patterns (for example, synchronous and asynchronous request/response, publish, notify, and receive/reply).

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 Published Business Services Web Service

Published business services web service deployed to a production application server.

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 production 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 business service

EnterpriseOne service level logic and interface. A classification of a published business service indicating the intention to be exposed to external (non-EnterpriseOne) systems.

published business service identification information

Information about a published business service used to determine relevant authorization records. Published business services + method name, published business services, or *ALL.

published business service web service

Published business services components packaged as J2EE Web Service (namely, a J2EE EAR file that contains business service classes, business service foundation, configuration files, and web service artifacts).

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.

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 message triggered from EnterpriseOne application logic that is intended for external systems to consume.

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.

replication server

A server that is responsible for replicating central objects to client machines.

rules

Mandatory guidelines that are not enforced by tooling, but must be followed in order to accomplish the desired results and to meet specified standards.

secure by default

A security model that assumes that a user does not have permission to execute an object unless there is a specific record indicating such permissions.

Secure Socket Layer (SSL)

A security protocol that provides communication privacy. SSL enables client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, and message forgery.

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.

serialize

The process of converting an object or data into a format for storage or transmission across a network connection link with the ability to reconstruct the original data or objects when needed.

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. The application also updates the Server Plan detail record to reflect completion.

SOA

Abbreviation for Service Oriented Architecture.

softcoding

A coding technique that enables an administrator to manipulate site-specific variables that affect the execution of a given process.

source repository

A repository for HTTP adapter and listener service development environment artifacts.

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.

SSL Certificate

A special message signed by a certificate authority that contains the name of a user and that user's public key in such a way that anyone can "verify" that the message was signed by no one other than the certification authority and thereby develop trust in the user's public key.

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.

super class

An inheritance concept of the Java language where a class is an instance of something, but is also more specific. "Tree" might be the super class of "Oak" and "Elm," for example.

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.

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 processing method

A method related to the management of a manual commit transaction boundary (for example, start, commit, rollback, and cancel).

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.

user identification information

User ID, role, or *public.

User Overrides merge

Adds new user override records into a customer's user override table.

value object

A specific type of source file that holds input or output data, much like a data structure passes data. Value objects can be exposed (used in a published business service) or internal, and input or output. They are comprised of simple and complex elements and accessories to those elements.

versioning a published business service

Adding additional functionality/interfaces to the published business services without modifying the existing functionality/interfaces.

Versions 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.

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.

Web Service Description Language (WSDL)

An XML format for describing network services.

Web Service Inspection Language (WSIL)

An XML format for assisting in the inspection of a site for available services and a set of rules for how inspection-related information should be made.

web service softcoding record

An XML document that contains values that are used to configure a web service proxy. This document identifies the endpoint and conditionally includes security information.

web service softcoding template

An XML document that provides the structure for a soft coded record.

Where clause

The portion of a database operation that specifies which records the database operation will affect.

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.

wizard

A type of JDeveloper extension used to walk the user through a series of steps.

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.

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

A

Agent Definitions form, 4-6
Agent Filters form, 4-7
agents
 connecting to monitors, 3-2
 domain resource agents, 2-4
 monitoring definitions, 4-6
 process instrumentation agents, 2-4
 setting buffer sizes, 4-6
 setting event sampling rates, 4-6
 setting filtering, 4-7
 understanding, 2-4
 understanding monitor URLs, 3-7
Application Engine programs
 PSPM_ARCHIVE, 4-5, 7-1
 reaper program, 4-6
application server domains
 monitored system elements, 2-4
application servers
 monitoring system elements, 2-6
 setting up for monitored systems, 3-8
archiving
 setting archive mode, 4-5
 understanding performance history data, 7-1
authentication
 understanding PMU 113, B-6

B

buffers
 setting for performance data, 4-6

C

charts
 monitoring performance data, 2-8, 5-1
Completed PMUs form, 7-2
contexts
 PMU 100, B-1
 PMU 101, B-2
 PMU 102, B-3
 PMU 106, B-4
 PMU 108, B-5
 PMU 109, B-5
 PMU 113, B-6

PMU 114, B-7
PMU 115, B-7
 understanding, 6-2
 understanding definitions, 6-6
counter metrics, 6-5

D

databases
 setting up for monitored systems, 3-7
 setting up monitoring databases, 3-1
 understanding JD Edwards application databases, 2-4
 understanding monitoring system databases, 2-5
deviation, standard, 5-1
domain resource agents, 2-4
domains
 application server, 2-4
 domain resource agents, 2-4
 identifying for agents, 4-7
 identifying for PMU requests, 7-4

E

events
 understanding, 2-7

F

filtering
 setting for agents, 4-7
 setting for PMUs, 6-7, B-1

G

gauge metrics, 6-5
Global Administration form, 4-3
grids
 monitoring performance data, 2-8

I

instrumentation
 process instrumentation agents, 2-4
 understanding, 2-6
 understanding PMUs, 6-2

J

- JD Edwards EnterpriseOne
 - events, 2-7, 3-1
 - PMUs, 2-7
 - system database (PTSYS), 2-5, 3-1
- JD Edwards EnterpriseOne Performance Monitor
 - administering, 4-1
 - agents, 2-4
 - analyzing historical performance data, 7-1
 - configuring, 3-1
 - elements, 2-2
 - instrumentation, 2-6
 - metadata, 6-1
 - monitored systems, 2-4
 - monitoring data, 2-8
 - monitoring system performance, 5-1
 - monitoring systems, 2-5
 - performance metrics coverage, 2-8
 - setting global system options, 4-3
 - setting up, 3-1
 - sharing performance data with third-party systems, 2-8
 - understanding, 2-1
- JD Edwards EnterpriseOne Performance Monitor Integration (PPMI)PPMI, 2-8
- JD Edwards Multi-Platform Installer program, 3-2

L

- labels
 - viewing for contexts, 6-7
 - viewing for metrics, 6-5, 6-8
 - viewing for PMUs, 6-7, B-1

M

- metadata
 - contexts, 6-6
 - metrics, 6-4
 - PMUs, 6-7
 - understanding, 6-1
- metrics
 - modifying definitions, 6-4
 - PMU 100, B-1
 - PMU 101, B-2
 - PMU 102, B-3
 - PMU 106, B-4
 - PMU 108, B-5
 - PMU 109, B-5
 - PMU 113, B-6
 - PMU 114, B-7
 - PMU 115, B-7
 - selecting for PMUs, 6-7
 - understanding, 6-1
 - understanding definitions, 6-4
- monitor servlets
 - identifying, 3-7
- monitored systems
 - defining, 4-5
 - enabling required elements, 3-6

- setting up application servers, 3-8
- setting up databases, 3-7
- setting up Process Scheduler servers, 3-8
- setting up web servers, 3-7
- specifying monitor URLs, 3-7
- understanding, 2-3, 2-4
- understanding application servers, 2-4
- understanding databases, 2-4
- understanding web server, 2-5
- monitoring systems
 - enabling required elements, 3-1
 - setting archive mode, 4-5
 - setting up monitoring databases, 3-1
 - setting up web servers, 3-2
 - understanding, 2-3, 2-5
 - understanding application servers, 2-6
 - understanding databases, 2-5
 - understanding Process Scheduler servers, 2-6
- Multi-Platform Installer program, 3-2

N

- notifications
 - events, 2-7
 - notifying monitor servlets of configuration changes, 4-6
- numeric identifier metrics, 6-5

O

- Open PMU Trees form, 5-4
- Open PMUs form, 5-2

P

- performance measurement units (PMUs), 2-7
- Performance Trace Console
 - launching, 4-6
- PMUs
 - entering search criteria, 7-2
 - monitoring completed, 7-2
 - monitoring open PMU trees, 5-4
 - monitoring open PMUs, 5-2
 - monitoring within tree context, 7-4
 - PMU 100, B-1
 - PMU 101, B-2
 - PMU 102, B-3
 - PMU 106, B-4
 - PMU 108, B-5
 - PMU 109, B-5
 - PMU 113, B-6
 - PMU 114, B-7
 - PMU 115, B-7
 - setting sampling rates, 4-6
 - setting timeouts, 4-6
 - understanding, 2-7, 6-2
 - understanding completed PMU status, 7-4
 - understanding definitions, 6-7
- PPMI
 - specifying passwords, 4-3
 - specifying user IDs, 4-3

understanding, 2-8
PPMI servlets
accessing, 4-3
process instrumentation agents, 2-4
Process Scheduler servers
monitoring systems, 2-6
setting up for monitored systems, 3-8
production monitoring systems
using the PTSYS database, 3-1
PSPM_ARCHIVE program, 4-5, 7-1
PSPM_REAPER program, 4-6
PSPPMSRV processes
registering servlets, 3-2
using PMUs, 4-8
PTSYS database, 2-5, 3-1

R

reaper program
Reaper Timeout status, 7-4
scheduling, 4-8
understanding PMU timeouts, 4-6
refreshes, form
System Performance forms information, 5-1
registration
agents, 4-7
servlets, 3-2

S

sampling
PMUs with enabled sampling, B-1
setting for agent events, 4-6
setting for agent PMUs, 4-6
searches
PMUs, 7-2
servers
application, 2-4
web, 2-5
servlets
running on web servers, 3-2
Simple Object Access Protocol (SOAP), 2-8
SOAP, 2-8
standard deviation
understanding, 5-1
status
understanding for completed PMUs, 7-4
string metrics, 6-5
System Definitions form, 4-5

T

tables
maintaining performance data tables, 4-8
moving timed-out PMUs, 4-6
third-party systems, sharing performance data, 2-8
timeouts
setting for PMUs, 4-6
understanding PMU timeout statuses, 7-4
tracing
enabling for monitored systems, 4-6

trees, PMU, 5-4, 7-4

U

URLs
specifying monitor URLs, 3-7
users
identifying for PMUs, 7-3
monitoring for PMU trees, 5-4

W

web servers
monitored system elements, 2-5
setting up for monitored systems, 3-7
setting up for monitoring systems, 3-2

