

# Oracle Best Practices for Utilizing Application Management Pack for JD Edwards EnterpriseOne with Enterprise Manager Cloud Control 12c

ORACLE WHITE PAPER | OCTOBER 2014





## Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.



## Contents

Executive Summary	1
Introduction	1
Deployment Architecture	3
Discovering JDE Components in EMCC	4
Rediscovery or Resynchronize Targets	6
JD Edwards Target Types	7
Server Manager versus AMS4JDE	8
Monitor Metrics for JD Edwards EnterpriseOne Targets	10
JD Edwards Targets Metrics	10
Metrics, Thresholds, and Collection Settings	12
Collection Frequency for JD Edwards	13
Metrics Threshold	14
Use Cases for Metrics	16
End-to-End Monitoring	17
Configuration and Compliance	18
Configuration Comparison	20
Compliance Rules	20
JD Edwards Groups	21
Leveraging Incident Manager	22
Establish Service Level (SLA)	24
JD Edwards Service Level Management (SLM)	24
Generic Service	25
Oracle Real User Experience Insight (RUEI)	28
Conclusion	32
References	33

## Executive Summary

JD Edwards customers manufacture, distribute, build, extract, service, and grow things that the world needs, and they are recognized among the leaders in their industries. JD Edwards customers are renowned for maintaining highly competitive enterprises and operating efficiently in markets of every size and in every geography. Operating efficiently necessarily means keeping costs low, including technology costs. Historically, JD Edwards products have enjoyed a well-earned reputation of providing tangible business benefits with extraordinarily low total cost of ownership.

Today, as part of the comprehensive hardware and software portfolio of Oracle, JD Edwards products have never been better. Synergy among Oracle product teams, Oracle's continued investment in JD Edwards products, and the Oracle overt strategy to create "Hardware and Software Engineered to Work Together" combine to provide JD Edwards customers with solutions that preserve their traditional high-value/low-TCO value proposition. The JD Edwards EnterpriseOne Application Pack for Oracle Enterprise Manager Cloud Control attempts to lower the cost of ownership of running JD Edwards applications by proactively managing, monitoring, and diagnosing faults in the JD Edwards components.

## Introduction

The Application Management Suite for JD Edwards EnterpriseOne for Oracle Enterprise Manager Cloud Control (*AMS4JDE*) leverages Oracle Enterprise Manager 12c (EMCC) as its technical foundation to enable the business-driven approach of managing the comprehensive abilities of JD Edwards EnterpriseOne Enterprise Manager. These comprehensive abilities (such as service-level management, compliance creation and management, application performance management, configuration management, real user experience insight (RUEI), integration with My Oracle Support (MOS), and third-party management tool integration) are made available through the suite. From a single Enterprise Manager console, you can manage many critical components of JD Edwards applications and their underlying IT infrastructure.

The intended audiences for this guide are JD Edwards EnterpriseOne administrators, system administrators, and other operational personnel responsible for managing JD Edwards EnterpriseOne applications that have bought JD Edwards AMS4JDE and require a guide to best solve the pressing management problems with the Application Management Suite. The guide discusses some challenges that JD Edwards administrators face and how they can leverage AMS4JDE to solve them. This is not a



detailed implementation or step-by-step how-to guide; instead, it contains references to specific guides for detailed understanding.

AMS4JDE enables you to use the graphical user interface of EMCC to monitor and manage your JD Edwards server products. AMS4JDE leverages the existing JD Edwards Server Manager functionality that monitors all entities constituting an EnterpriseOne installation. For example, Server Manager collects real-time data for user sessions, Java memory usage, JDBj connection cache, EnterpriseOne kernel-specific metrics, and prepared statement configurations.

The integration of AMS4JDE with RUEI enables enterprises to maximize the value of their JD Edwards applications by delivering insight into real end-user experiences. The integration of AMS4JDE with RUEI enables you to identify initial user requests, track them through all responses, and collect, process and present details of every transaction. The passive or non-intrusive monitoring capability of RUEI does not require any modification changes or instrumentation of applications. Once installed and running, RUEI can help eradicate blind spots and lost revenue for frustrated users, and accelerate problem resolution for applications that are performing poorly.

In particular, AMS4JDE offers the following key capabilities for JD Edwards EnterpriseOne components and services<sup>[3]</sup>:

- » Discovers and registers JD Edwards EnterpriseOne targets
- » Monitors metrics for JD Edwards EnterpriseOne targets
- » Displays configuration data in a graphical user interface that is driven by metadata
- » Creates compliance rules from the configuration data and performs a regular compliance check; for example, security compliance for JD Edwards EnterpriseOne targets
- » Enables users to create a graphical topology that displays the relationships between targets, and to execute selected actions on targets from this view
- » Enables users to create a service that simulates a transaction, such as login and logout, to monitor the availability of an application
- » Creates a JD Edwards EnterpriseOne system in EMCC that enables you to see how all of the targets in the system are related to the JD Edwards EnterpriseOne application database
- » Integrates with RUEI and enables you to see the captured user transactions in detail

You can add or see additional EMCC by using the EMCC Advanced Configuration [Guide](#).

## Deployment Architecture

The following architectural diagram shows a conceptual view of the deployment architecture for the various operational components required to integrate existing JD Edwards EnterpriseOne functionality of Server Manager with AMS4JDE. The diagram highlights various components involved in the AMS4JDE offering: Enterprise Manager (EMCC), Server Manager, EMCC Agents, JD Edwards EnterpriseOne Repository, EnterpriseOne HTML Server, EnterpriseOne Enterprise Server, and the DB Server. The EMCC is subdivided into EM Agent, Oracle Management Server (OMS), and the Management Repository. For details about the various EMCC components, see the EMCC Guide<sup>[1]</sup>.

To enable the communication of the EM Agent with JD Edwards Server Manager, a JD Edwards EnterpriseOne connector runs on the EMCC Agent. The Agent using the connector (JMX) periodically polls the Server Manager console to collect the monitoring and configuration statistics, which are defined out-of-box for the JD Edwards targets. The collected statistics are then pushed into the EM repository through HTTP(s) protocol. Server Manager to the EnterpriseOne Embedded Agent communication is an existing channel and is not changed or modified for the AMS4JDE.

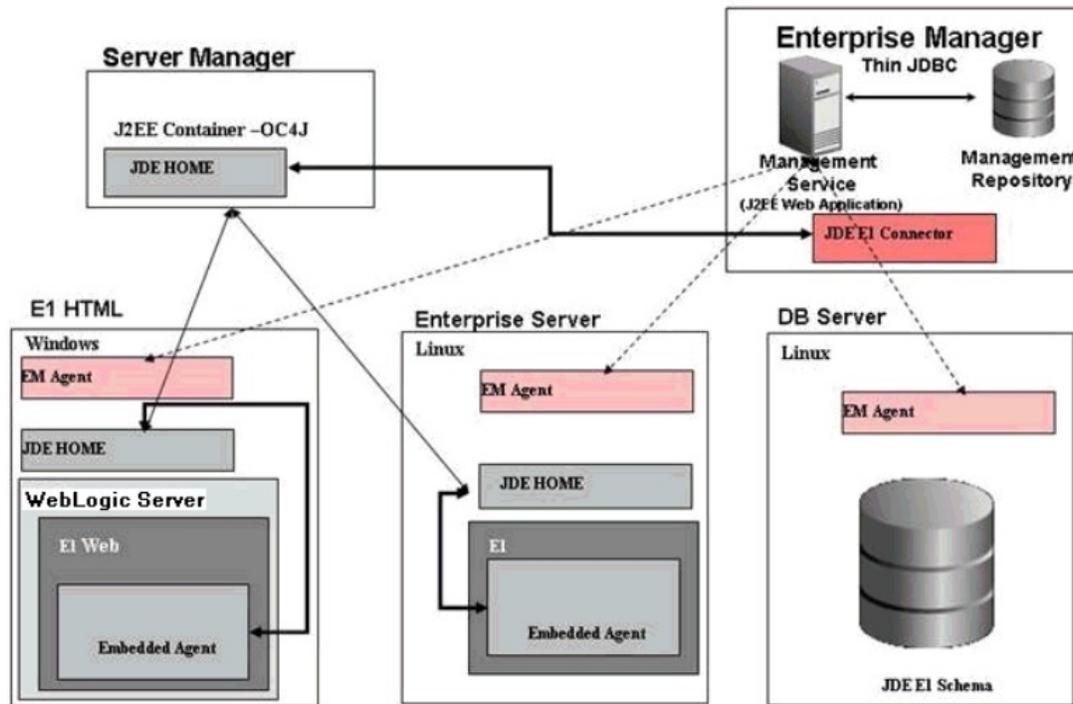


Figure 1 JD Edwards Deployment Architecture

The preceding diagram shows the EM Agents for the Linux platform.

Please note that the diagram shows one scenario. All JD Edwards EnterpriseOne open architecture operating systems are supported, such as:

- » Windows
- » Linux



» Solaris

See the [supported platforms](#) for specific details and installation guidelines for different platforms. If there is no EM agent available or supported for a platform that is running JD Edwards EnterpriseOne, you can continue with the monitoring by installing the agent on one of the supported platforms from the above list and remotely monitoring the JD Edwards EnterpriseOne components. AMS4JDE uses JMX to collect the statistics, which makes cross-platform monitoring possible as long as the JD Edwards EnterpriseOne connector is able to communicate with the Server Manager instance.

## Discovering JDE Components in EMCC

When the AMS4JDE components are successfully installed on both the Agent and OMS side (See the [Guide](#) for deployment details), the next step is to start discovering the JD Edwards Components in AMS4JDE. As a precondition, it is critical that there are no errors while deployments. If there are any errors, it is recommended that you clear those and only then proceed with the discovery. Also ensure that the EMCC agent used for discovering targets is functioning.

The discovery process identifies all Server Manager Instances and registers the associated JD Edwards targets so that they can be monitored and managed from the EMCC console. If you have many Server Manager Instances configured, then you can add or discover the related JD Edwards targets one instance at a time. The JD Edwards EnterpriseOne connector on the EM Agent is used to talk with Server Manager and get the associated targets. To add the JD Edwards Enterprise Domain, enter the following details on the discovery screen. The first field is the identifier for the domain target and is prepended to each target that is discovered. This identifier can be used to differentiate targets from different environments; for example, production, development, and so on.

The following figure demonstrates the discovery process of AMS4JDE on a JD Edwards EnterpriseOne domain:

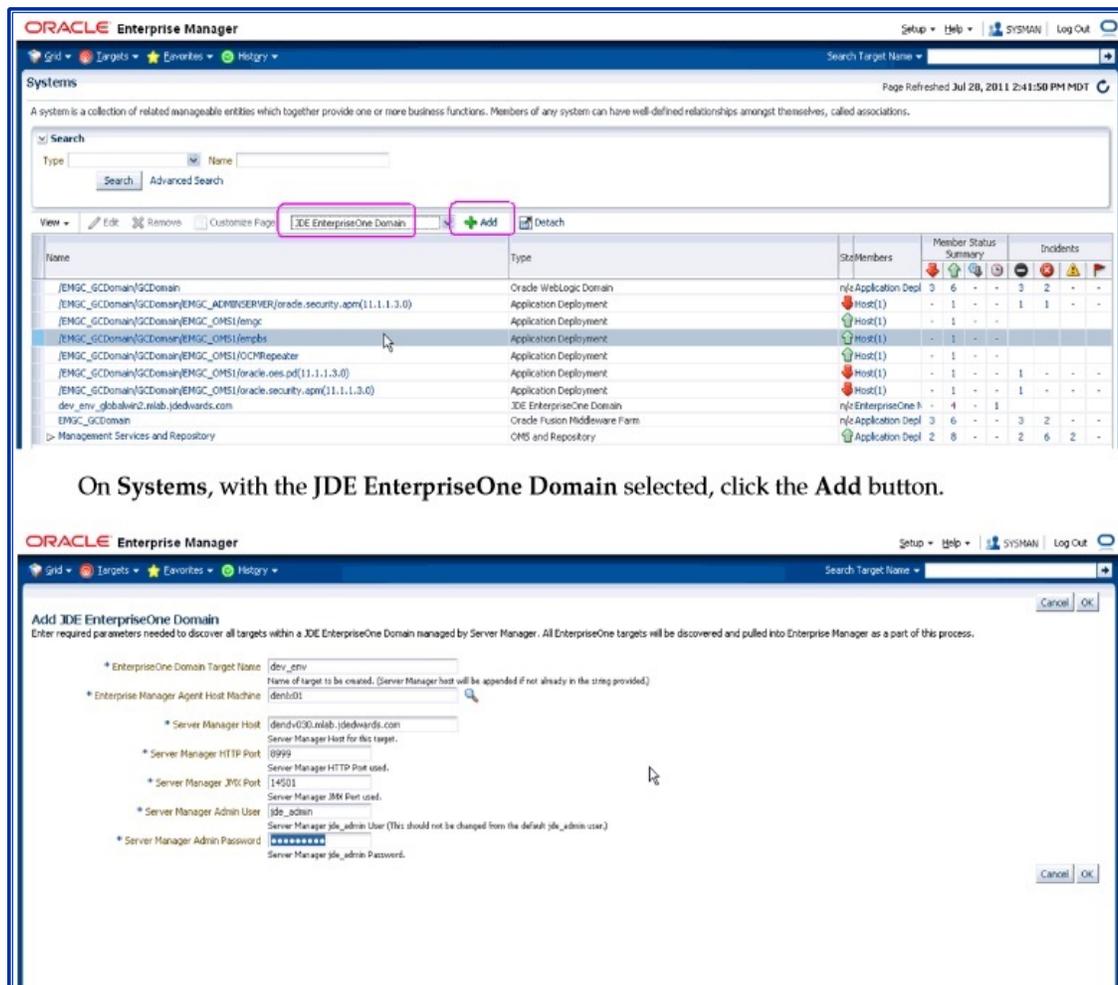


Figure 2 AMS4JDE Discovery

This table describes each of the discovery input parameters in greater detail.

TABLE 1 AMS4JDE DISCOVERY PARAMETERS

Field	Details
EnterpriseOne Domain Target Name	The name of the domain for JD Edwards EnterpriseOne. The name of the Server Manager host will be appended to this name if you do not specify it. For example, your target name can be <b>dev_env</b> .
Enterprise Manager Agent Host Machine	The machine name on which the Enterprise Manager agent is installed. For example, your machine name can be <b>denlx01</b> .
Server Manager Host	The fully qualified machine name of your Server Manager host. For example, your machine name can be: <b>denv030.mlab.jdedwards.com</b> .
Server Manager HTTP Port	The HTTP port that will be used to connect to Server Manager. The default value is <b>8999</b> .
Server Manager Admin User	The admin user that will be used to connect to the JMX Server. The default value, which must not be changed, is <b>jde_admin</b> .
Server Manager Admin Password	Valid password for the Server Manager admin user. This password is stored in the EMCC repository for future management actions.

After discovery, you can see the entire JD Edwards topology (graphic as well as tabular) on a single JD Edwards Enterprise Domain page and manage all of the member targets from the single JD Edwards Enterprise domain target, which is the highest target in the JD Edwards target hierarchy.

The following configuration topology figure shows the graphical representation of JD Edwards targets through the Configuration Topology viewer in AMS4JDE. The JD Edwards Domain target is at the top of the hierarchy and has the following members: Enterprise Servers, Enterprise HTML Servers, Business Servers, and Transaction Servers. The topology viewer enables you to perform some administrative actions from the screen, such as notifications, alerts, and up/down status. You can view the JD Edwards hierarchy in a tabular form by selecting the Table option in the Display field.

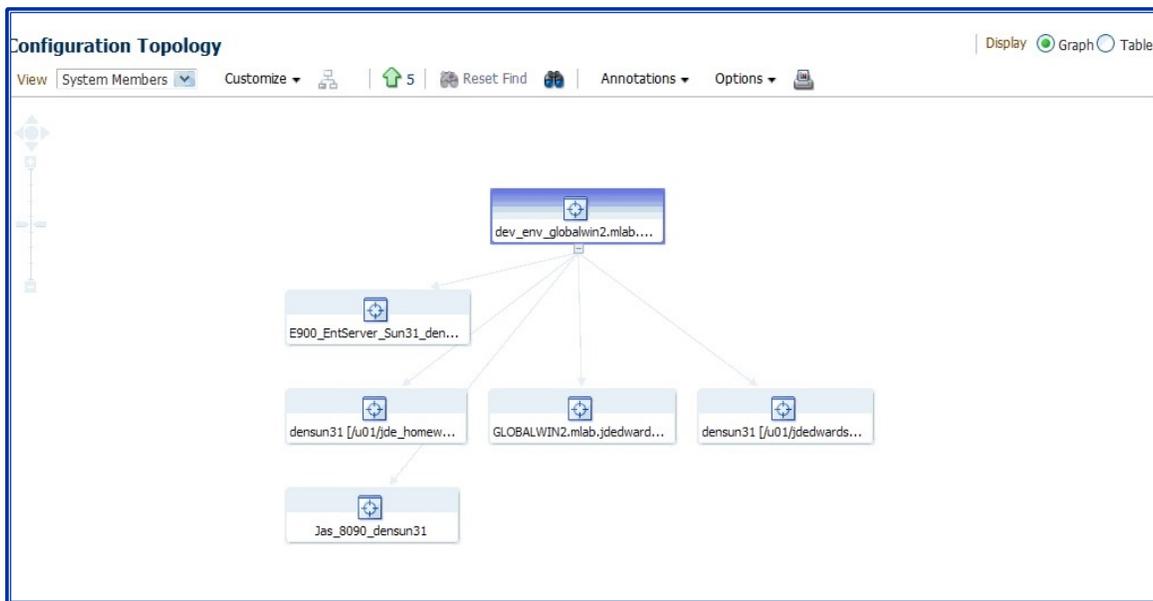


Figure 3 A Sample of JD Edwards Deployment Topology

### Rediscovery or Resynchronize Targets

After the initial discovery is performed, the following can happen:

Either the JD Edwards system load has increased and you decide to scale horizontally; or, by adding a new instance of System, the load has gone down and you decommission a server instance.

In either of the scenarios, it is important that EMCC is synchronized with the system changes to continuously monitor and manage JD Edwards components. This resynchronization can be achieved by rediscovery or *Refresh Discovery* from the JD Edwards EnterpriseOne Domain Target *CustomMenu* -> *Refresh Discovery* option, as shown here:

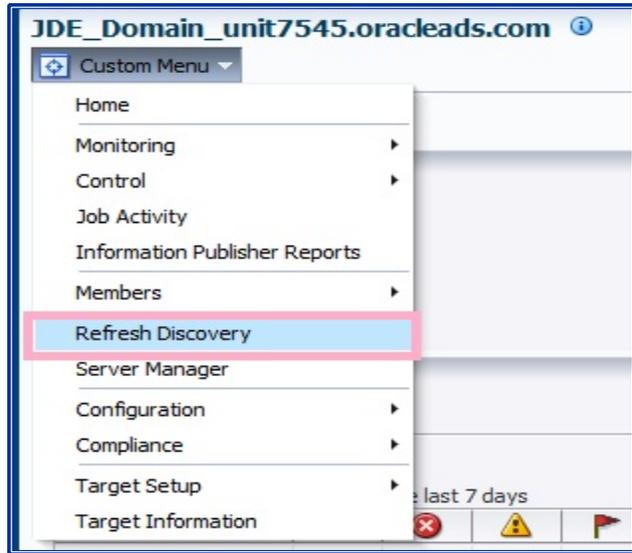


Figure 4 Refresh Discovery

Refresh Discovery resynchronizes EMCC and JD Edwards components to correctly reflect the JD Edwards topology into the EMCC.

## JD Edwards Target Types

As part of the discovery action, different JD Edwards entities are added to the EMCC. In the EM world, such entities are called the target and are anything that can be monitored and managed. AMS4JDE is available out-of-the-box to manage various JD Edwards targets that are critical for the functioning of the system. Table 2 shows different targets that were added to EMCC to manage and monitor JD Edwards EnterpriseOne applications.

TABLE 2 AMS JD EDWARDS TARGET TYPES

Enterprise Manager Target	JD Edwards Entity	Purpose
JD Edwards EnterpriseOne Domain	None	The logical target to associate each JD Edwards EnterpriseOne component.
EnterpriseOne Server	Enterprise Server	The server that executes core transactions, reports, and batch processes.
EnterpriseOne Data Access Server	DAD Server	The server that is responsible for providing the database (driver) access to different databases.
EnterpriseOne HTML Server	JAS Server	The basic HTTP server that directs requests from a browser to the Application Server.
EnterpriseOne Managed Home	JDE Management Home	Each location in which the Server Manager agent software has been installed is referred to as a managed home.
EnterpriseOne Transaction Server	Transaction Server	The J2EE server that is responsible for the guaranteed business event delivery.
EnterpriseOne Business Services Server	Business Server	The J2EE server that supports the inbound and outbound web services for JD Edwards EnterpriseOne.

## Server Manager versus AMS4JDE

Customers might understand AMS4JDE to be in direct competition with Server Manager, a native JD Edwards tool shipped with JD Edwards EnterpriseOne for management and diagnostics. Customers, therefore, are not sure of which tool to use under various circumstances: AMS4JDE or Server Manager.

Server Manager is a JD Edwards flagship product for system management, and it has served our customers efficiently. Server Manager is adept in collecting real-time statistics, updating configuration for different components, starting or stopping various components, and creating new instances. However, the abilities of Server Manager are limited. JD Edwards deployments are very complex and involve different components such as databases, application servers, storage components, networking gears, and many other cross-vendor pieces that serve large deployments, but Server Manager can manage only the JD Edwards components. To manage a diverse and complex topology, you require the robust and feature-rich AMS4JDE.

JD Edwards has an ongoing effort to continuously evolve AMS4JDE to meet the system management concerns of our customers, including features that are currently exposed through Server Manager. An integration of AMS4JDE with the Server Manager is available, and users can navigate to the *CustomMenu* -> *Server Manager* Option to use it. This action launches the associated Server Manager instance, and users can sign in with their Server Manager credentials, as shown here:

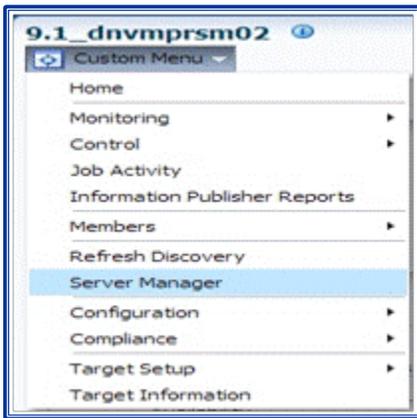


Figure 5 Server Manager Integration

Table 3 compares AMS4JDE and Server Manager and highlights strengths for different use cases. This comparative study can help customers use the right tool for the circumstances. A check mark in a column indicates that the feature is available. Empty cells in the table indicate capability feature or capability that is currently not supported. These cells also show comments for partial or limited support.

**TABLE 3 SUMMARY TABLE COMPARISON OF FEATURES BETWEEN AMS4JDE AND SERVER MANAGER**

Capabilities	AMS4JDE	JD Edwards EnterpriseOne Server Manager
Deploying JD Edwards Targets		✓
Single Management Console for Entire JD Edwards Hierarchy	✓	Only targets registered with a given Server Manager instance. Larger deployments might need multiple Server Manager consoles.



Capabilities	AMS4JDE	JD Edwards EnterpriseOne Server Manager
Proactive Monitoring	 Define critical and warning thresholds	
Read - Write Configuration	Read only	
Configuration Comparison		Limited support for configuration comparison.
Compliance Rules Check		
Configuration Snapshot		
Historical and Trend Analysis		Only real-time data.
Alerts and Notifications		Alerts only for specific stats.
Reports		
Graphical Topology		
Clear JD Edwards EnterpriseOne Cache		
Start and Stop Services		 For some components.
View EnterpriseOne Logs		
System Targets to Monitor Related Targets Together		
Service Targets to Monitor Critical Business Service		

Capabilities	AMS4JDE	JD Edwards EnterpriseOne Server Manager
Synthetic Tests for End User Transactions	✓	
End-to-End Stack Monitoring	✓	Only JD Edwards targets.

## Monitor Metrics for JD Edwards EnterpriseOne Targets

The EMCC agent collects the statistics or metrics for different targets (members) of the JD Edwards domains. To view the statistics on the target page, navigate to the *Target Monitoring -> All Metrics* page for Enterprise Server target, as shown here:

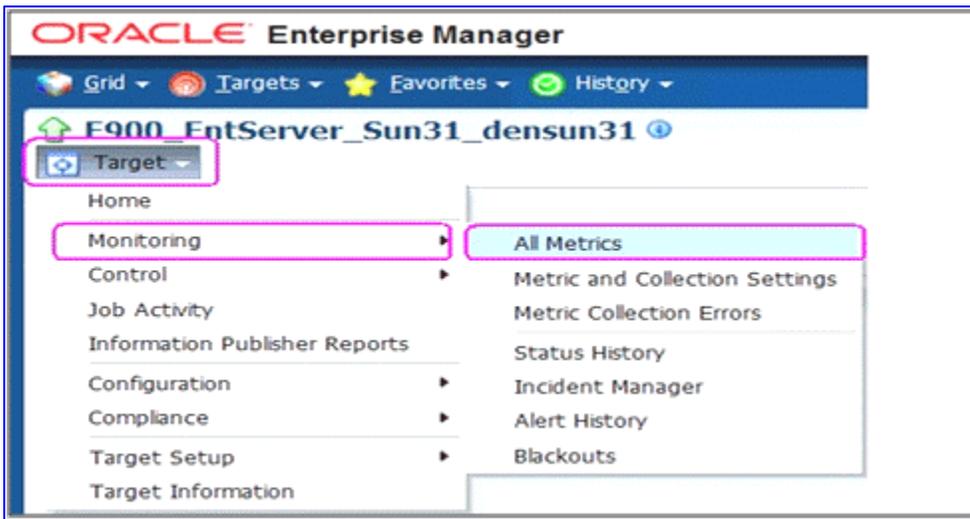


Figure 6 Metrics Monitoring for JD Edwards Enterprise Server

## JD Edwards Targets Metrics

This section describes some critical metrics collected for different JD Edwards targets.

### JD Edwards EnterpriseOne Enterprise Server

A single management kernel process is created when you start the enterprise server. The management kernel is responsible for securely networking between the enterprise server and the JD Edwards management console while providing the runtime details of the enterprise server. JD Edwards targets expose rich diagnostic data. The following is a list of some critical metrics and their details:

- » Average CPU Percentage Used by all Kernels

This is an aggregated metric that gives the average CPU usage percentage for different JD Edwards kernels: SAW, Metadata, Security, Queue, Workflow, and so on. This metric is useful for identifying overall resource consumption by the JD Edwards kernels. A high number here indicates a deadlock or some fault in business function—a loop or a bug. It can also mean that operating system resources are stuck.

- » Kernel Process

This is a tabular metric and is critical for the diagnostics of the JD Edwards NET\_K processes. It captures all information about different kernel processes that are running: logs, thread, DB Cache, database connection, and other details related to kernel processes.

The following table provides more details about critical columns:

**TABLE 4 JD EDWARDS ENTERPRISEONE ENTERPRISE SERVER NETWORK PROCESS STATISTICS**

Column	Description	Thresholds	
		Warning	Critical
Process Name	The name of the kernel process. The name indicates the definition that the kernel belongs to.	N/A	N/A
Active User Count (Kernel)	Total number of active users for each kernel. This statistic is useful to understand the load on the system at any given point. It can be used for capacity planning.	70	100
Messages Received	The total number of messages (requests) that have been processed by the kernel process.	*	*
Outstanding Requests	The number of requests that are queued and waiting for processing by the kernel process. A high number of outstanding requests indicate that the kernel is occupied either because of excessive load on the system or because of some issue with the kernel code.	2	4
Threads (Kernel)	The number of active threads for the Enterprise Server process.	*	*
Last Active Time	The last time the kernel performed any activity, such as processing incoming JD Edwards NET messages. This is useful for interoperability users that can stay logged on for a long time. In case of a leak, this statistic can give indicators because there would be some activity against such users.	*	*
Database Connection	Number of open database connections for each kernel process. A high number here indicates that one of the kernels or run-batch processes is consuming high database resources.	*	*
Total Open JDB Transactions	Total (Manual + Auto) commits # of Open JDB Transactions. A very high number for this metric indicates high level of database activity, which could possibly be caused by some faulty loop in business function, or any unclosed/leaking transaction.	*	*
JD Edwards Cache	Session data stored per kernel process. A high number for this statistic could mean the process is consuming high memory.	*	*
Messages Received (Network)	The total number of messages (requests) that have been processed by each kernel process.	*	*
Total Message Time (Network)	The total time in millisecond to process all of the messages.	*	*
Incoming Connection (Network)	<b>The network listeners receive JD Edwards NET-based connection requests from foreign clients and servers and dispatch the messages to the appropriate kernel process. This statistic shows a number of connections for the JD Edwards NET listener process. This metric is useful to understand the load on the system.</b>	*	*

\*See [Metrics Threshold](#) to calculate these thresholds on your system.

## JD Edwards EnterpriseOne HTML Servers

CallObject statistics is one of the metrics for the target JD Edwards EnterpriseOne HTML server.

The CallObject statistics detail the business functions that have been invoked by a web product. Each business function that is called is recorded and displayed along with timing, invocation counts, and error counts. This is a tabular metric and is critical for the diagnostics of the JD Edwards CallObject processes.

This table shows the JD Edwards EnterpriseOne server targets and their critical metrics and threshold values:

**TABLE 5 JD EDWARDS ENTERPRISEONE HTML SERVER STATISTICS**

Column	Description	Thresholds	
		Warning	Critical
JVM Node Id	An identifier that uniquely identifies an individual JVM. This column is displayed only if an instance with multiple JVMs is detected.	n/a	n/a
Enterprise Server	The enterprise server name and port to which the metrics apply.	n/a	n/a
Business Function Name	The business function that was invoked.	n/a	n/a
Total Invocations	The total number of business function invocations.	*	*
Average Time	The average amount of time, in milliseconds, that the invocations of this business function took. The first invocation of a business function is not counted in the average due to its initial overhead. This statistic is a valid indicator for adjusting the number of users for each call object kernel.	120 ms	150 ms
Longest Time	The longest amount of time, in milliseconds, that an invocation of this business function took.	150 ms	200 ms
Timeout Errors	The number of invocations of this business function that resulted in a JD Edwards NET timeout instead of a successful completion.	*	*
Application Error	The number of invocations of this business function that resulted in an application returned error instead of a successful completion.	1	2
System Errors	The number of invocations of this business function that resulted in a system error instead of a successful completion.	1	*

\*See [Metrics Threshold](#) to calculate these thresholds on your system.

## Metrics, Thresholds, and Collection Settings

The statistics that are collected by AMS4JDE can broadly be categorized into performance or usage; however, we do not label them as such. These statistics are very critical and can give a deeper insight to the system behavior at any given time; however, the raw statistics by themselves do not add a lot of value. As a system administrator, you might have to define boundaries to proactively act upon any service outage or any other major disruption.

EMCC has the ability to define thresholds or boundary values against which monitored metric values are compared. When a threshold is reached, an alert or event is generated and system administrators are notified to either take some corrective action or at least be informed.

For example, you receive an alert when the *Number of Outstanding Requests* for an Enterprise Server instance exceeds a certain maximum threshold; otherwise a potential problematic condition can go unnoticed and result in a bigger problem—service outage or degraded performance. You can configure a threshold for this metric and get notified. This can be done from *All Metrics* -> *Metric and Collection Settings*, as illustrated here:

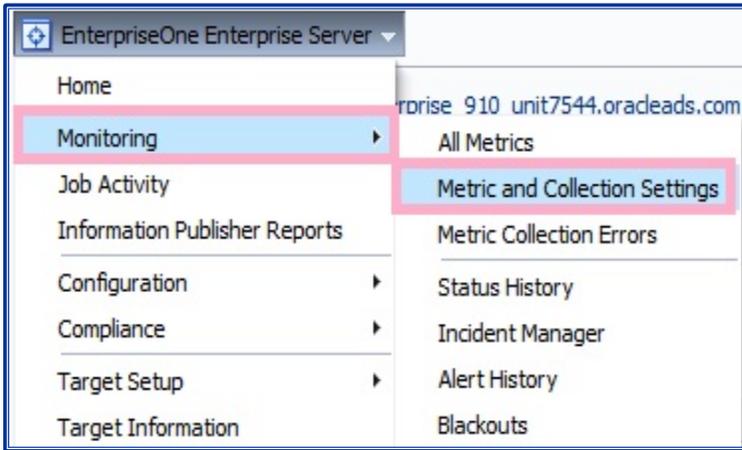


Figure 7 Metrics Collection Settings for JD Edwards Enterprise Server

### Collection Frequency for JD Edwards

For optimal performance, Oracle recommends to not set a very low value (<5 sec) for poll frequency. Setting low values means frequent collection, and there is a possibility that it might cause additional load on the JD Edwards components which in turn can affect system performance. Users based on the criticality of metrics must set the correct polling frequency.

Each target has a status metric, which returns 1 when the target is up, and 0 when the target is down. The status metric is critical and should be collected at least every 5 seconds; a delay will only defer the outage condition. Similarly for other statistics, Oracle recommends that the polling frequency should be set anywhere between 15 minutes to 30 minutes.

The screen below shows the threshold values and polling frequency for some of the critical statistics for the JD Edwards HTML Servers, such as *Java Heap Memory* and the *Total User Count* statistics:

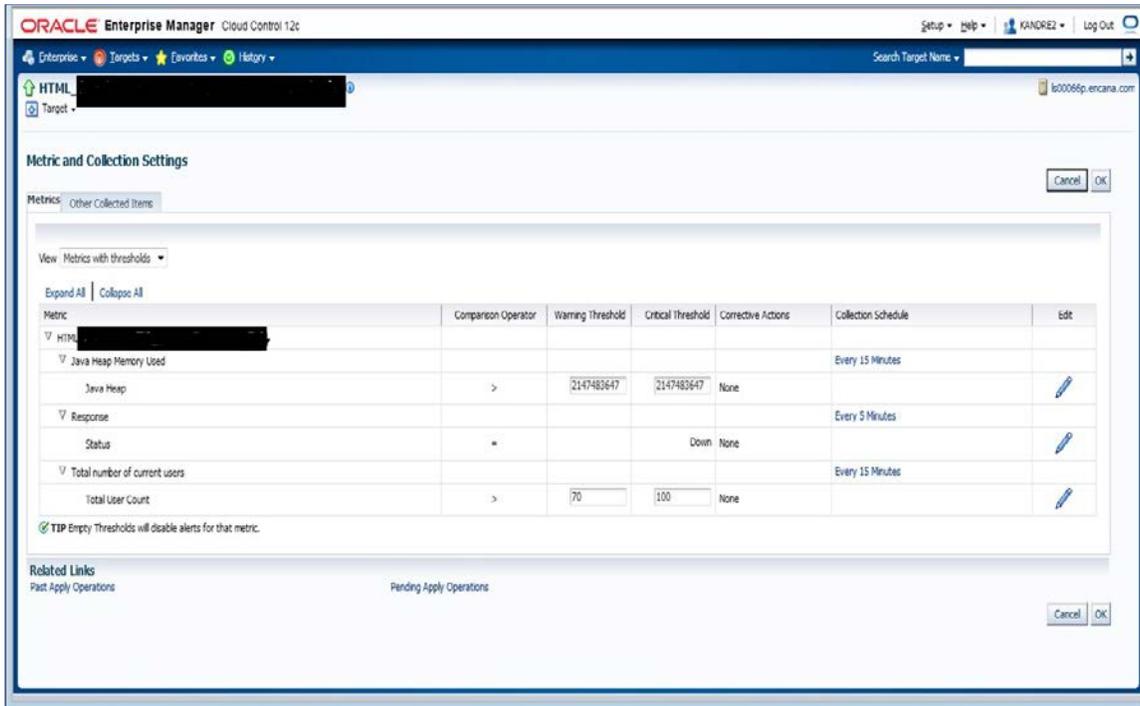


Figure 8 Threshold Values for JD Edwards HTML Server

The *Edit* column in the preceding screen lets you edit the threshold values. To be able to perform the Edit operation, a user requires operator privileges on the target (for details about the various privileges, see the EM [Guide](#)). To open the advanced threshold page, click the icon in the edit column. The following screen shows the threshold settings section from the page.



Figure 9 Advanced Threshold Settings

For more details about this feature see the section of the Administrator's [Guide](#).

### Metrics Threshold

AMS4JDE is predefined out-of-box for some targets. These values are acceptable for some monitoring conditions; however, some other environments might require customized threshold values to more accurately reflect the operational norms specific to their environment. Setting accurate threshold values can be more challenging for



certain categories of metrics, and there is no one-size-fits-all solution for setting thresholds. A few approaches for setting metric thresholds are discussed in this section.

### **Metric Snapshots**

A metric snapshot is a collection of a target's performance metrics that have been collected at a specific time. You use a metric snapshot as an aid in calculating metric threshold values based on the target's past performance.

The key to defining a metric snapshot for a target is to select a date during which target performance was acceptable under typical workloads. Given this date, actual values of the performance metrics for the target are retrieved and these represent what is normal or expected performance behavior for the target. Using these values, you can then use EMCC to calculate warning and critical thresholds for the metrics that are a specified percentage worse than the actual metric snapshot values. These represent values which, when crossed, can indicate performance problems. After thresholds are calculated, you can edit the calculated values if required.

You can define a metric snapshot for a target based on a date and (optionally) time. If you specify only a date, the metric snapshot is the set of average daily values of the target's performance metrics for that date. If you also specify an hour within the date, then the metric snapshot is the set of low and high metric values for the preceding hour.

### **Metric Baselines**

Metric baselines are statistical characterizations of system performance over well-defined time periods. You use metric baselines to implement adaptive alert thresholds for certain performance metrics as well as provide normalized views of system performance. Adaptive alert thresholds are used to detect unusual performance events. Baseline normalized views of metric behavior help administrators explain and understand such events. Metric baselines are well-defined time intervals (baseline periods) over which Enterprise Manager has captured system performance metrics. The underlying assumption of metric baselines is that systems with relatively stable performance should exhibit similar metric observations (that is, values) over times of comparable workload. Two types of baseline periods are supported: moving window baseline periods and static baseline periods. Moving window baseline periods are defined as some number of days prior to the current date (for example, last 7 days). This allows comparison of current metric values with recently observed history. Moving window baselines are useful for operational systems with predictable workload cycles (for example, batch processes). Static baselines are periods of time that you define that are of particular interest to you (for example, end of the fiscal year). You can use these baselines to characterize workload periods for comparison against future occurrences of that workload (for example, compare end of the fiscal year from one calendar year to the next).

### **Adaptive Thresholds**

After metric baselines are defined, they can be used to establish alert thresholds that are statistically significant and adapt to expected variations across time. For example, you can define alert thresholds to be generated based on significance level, such as the HIGH significance level thresholds are values that occur 5 in 100 times. Alternatively, you can generate thresholds based on a percentage of the maximum value observed within the baseline period. You can use these thresholds to generate alerts when performance metric values exceed normal peaks within that period.

## Use Cases for Metrics

This section discusses some use cases for collecting metrics for diagnostic purposes.

### Problem #1 Determine the Optimal Number of Users for the Environment

The JD Edwards EnterpriseOne Enterprise Server makes use of various jobs called kernels to complete work for a web user request. CallObject kernels, the primary process for processing user interface requests, runs business functions (C-based business logic and database queries) for web users. The configuration interface provides two types of settings related to these jobs. The first setting, called maximum number of processes, is the maximum number of jobs allowed to run. The second setting, called auto-start process count, is the number of kernel processes automatically started when JD Edwards EnterpriseOne services are initiated. Additional jobs are started if a request arrives and all jobs are currently active—but only up to the maximum number of jobs allowed. Automatically starting a reasonable number of CallObject kernels is recommended, but depends on the environment.

In an environment where the workload is high and users all log in at the same time, it is reasonable to set the value for *AutoStartProcess* equal to the value for *maxNumberOfProcesses*. In an environment where the workload is low and work does not start at the same time, setting the value for *AutoStartProcess* equal to half of the value for *maxNumberOfProcesses* can yield better results. CallObject kernel jobs are capable of handling the requests for multiple users. Each job request caches information and uses system resources. Proper configuration becomes important because starting too many kernel processes can cause excessive memory and CPU overhead. However, the presence of too few CallObject kernels could limit resources for each user, causing the user's response times to increase.

One way to establish the optimal number of users per kernel for a specific environment is to vary the call object kernel-to-user ratio, use AMS4JDE to monitor the *average response time* metric for the EnterpriseOne HTML server, and compare the results. The number of CallObject kernels that minimizes these response times during peak workload is the correct value for the environment.

### Problem #2 Users Unable to Log In to JAS Server

The maximum users setting in the JAS.INI acts as a threshold. After the maximum number of users has been reached, no additional user is allowed to log in to JD Edwards EnterpriseOne until another active user logs out. It is useful to monitor the active users in Enterprise Manager by setting the warning threshold for the *User Count* metric in the JD Edwards EnterpriseOne HTML Server target to about 70 percent of the value in the JAS.INI and a critical threshold to about 85 percent of the threshold value in the JAS.INI. Continuously monitoring and setting right thresholds enables continuity of service for the users.

### Problem #3 Business Functions Taking Time to Execute or General System Slow Response

To address the problem of long-running business processes (LRBFs), the multithreading capability was added to the CallObject kernels. Multithreading CallObject kernels guarantees that a business function call will be processed immediately. Function call requests are never queued when multithreading is enabled. For smooth functioning of the multithreaded requests, it is very important that the thread-pool size is correctly setup and a warning is generated when the number of threads is approaching the maximum number of threads each kernel can handle. The *Kernel Process* metric defined for the JD Edwards Enterprise Server Target has a column called *Threads (kernel)* that shows the number of threads for each kernel. Setting the warning and critical thresholds for these threads to about 80 percent and 90 percent of the thread pool size value defined in the .INI file ensures that when there are a large

number of concurrent requests there is no overhead of creating and destroying the 'Overflow' thread, and that the thread pool size can accommodate the load.

## End-to-End Monitoring

JD Edwards applications and the underlying infrastructure form a critical part of an enterprise's IT infrastructure. As a JD Edwards administrator, you will want to manage and monitor all of the JD Edwards components, from the business applications to the databases, to other infrastructure components, such as servers and ideally the single management console. In absence of an effective management tool, most administrators write some silo scripts that are neither scalable nor manageable as the system evolves. Enterprise Manager EMCC from Oracle can be leveraged to manage the entire stack as well as the third-party vendors. There are management packs available for application servers and database.

Enterprise Manager captures JD Edwards host performance metrics in the repository and these details can be accessed from Cloud Control by selecting the JD Edwards Host target, and then selecting Monitoring CPU/Memory/Disk Details. This displays a graphical view of the JD Edwards Enterprise Host performance metrics as shown in the following screenshot. You can display host performance data for the last 31 days by selecting the appropriate option from the drop-down box. CPU utilization metrics and the *Top 10 Host processes (ordered by CPU)* are seen in this screenshot (some of the jdenet\_k processes are shown):

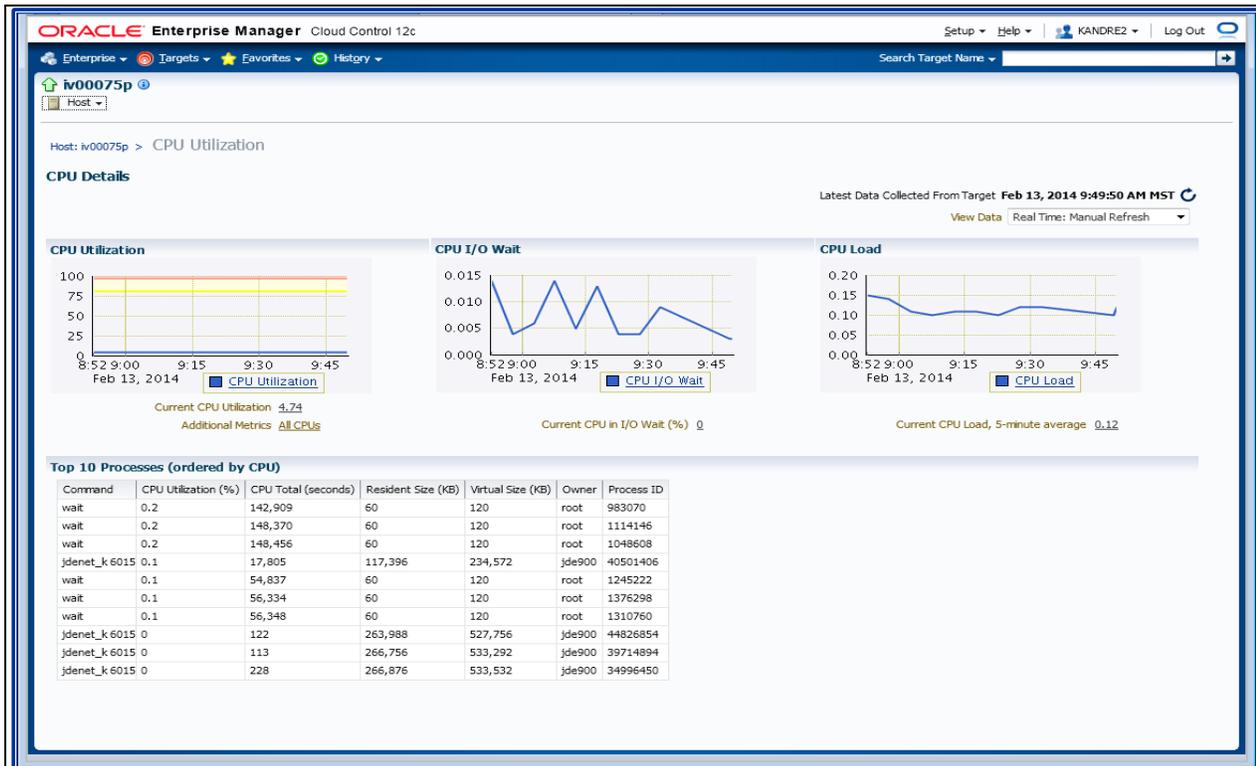


Figure 10 JD Edwards Hosts Monitoring

### Host Statistics

The host management capabilities in Enterprise Manager provide a quick glimpse of all of the hosts on your system. This includes lifecycle status and configuration changes.

The following table provides details related to host statistic and thresholds:

**TABLE 6 JD EDWARDS ENTERPRISEONE HOST STATISTICS AND THRESHOLDS**

Column	Description	Thresholds	
		Warning	Critical
% disk device busy	Excessive disk operations might indicate an OS level issue possibly leading to reduced performance or a system crash.	80	95
% file system space available	Reduced file system space might indicate the creation of excessive temp or log files. Exhausting file system space can result in a system crash.	20	5
% CPU Utilization	High CPU utilization might indicate an application issue possibly leading to degraded system performance.	80	95
% Swap utilization	Swap space is space on your disk drive that is reserved to supplement your memory. When there is not enough physical memory to store all the running programs, then programs that are not being actively used are moved to the swap space.  Excessive swap space utilization might indicate an application issue; exhausting system swap can result in a crash.	80	95
% free memory available	Freely available memory.	<10	<5

## Configuration and Compliance

Enterprise Configuration Management collects configurations as a collection of configuration snapshots. A configuration snapshot is a large collection of information that changes infrequently relative to performance metrics. Each configuration snapshot is associated with an Enterprise Manager target. To view the configuration for any target, navigate to *Target -> Configuration -> Last Collected*, as illustrated in the following figure:

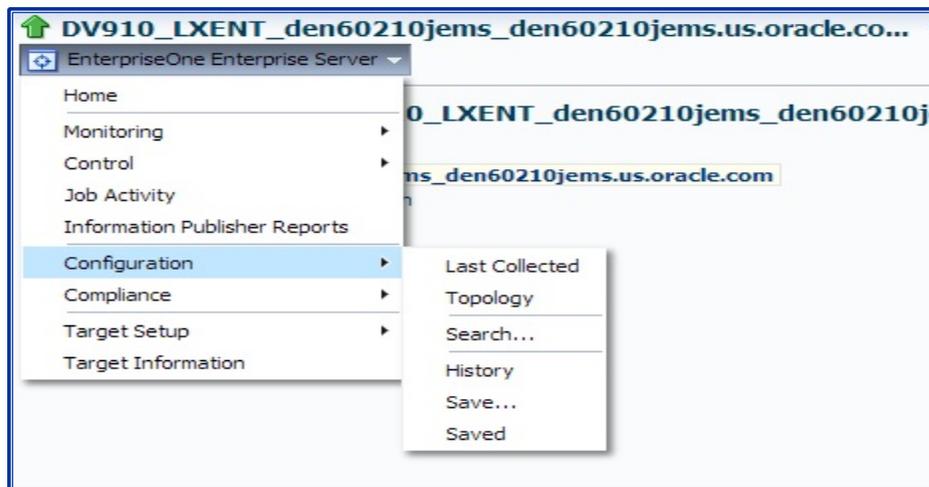


Figure 11 Configuration Collection

On the configuration screen, users can see the configuration defined for that target, as well as the relationship of JD Edwards targets with other Enterprise Manager targets. For example, if the Enterprise Server is hosted on any server, the relationship of the Enterprise Server target with that host is *hosted\_by*, or with the Enterprise Domain the top-level target is *contains*, and the relationship to the agent that is monitoring this target is *monitored\_by*.

The relationship is useful to understand the impact a target has on other components; for example, if the JD Edwards EnterpriseOne Server target is not functioning for any reason (a patch has to be applied), it negatively impacts the availability of the system of the group target.

When you click the configuration for the targets, you can view the configuration collected. For example, the following screen shows the configuration from the JD Edwards INI files collected for the JD Edwards EnterpriseOne Target:

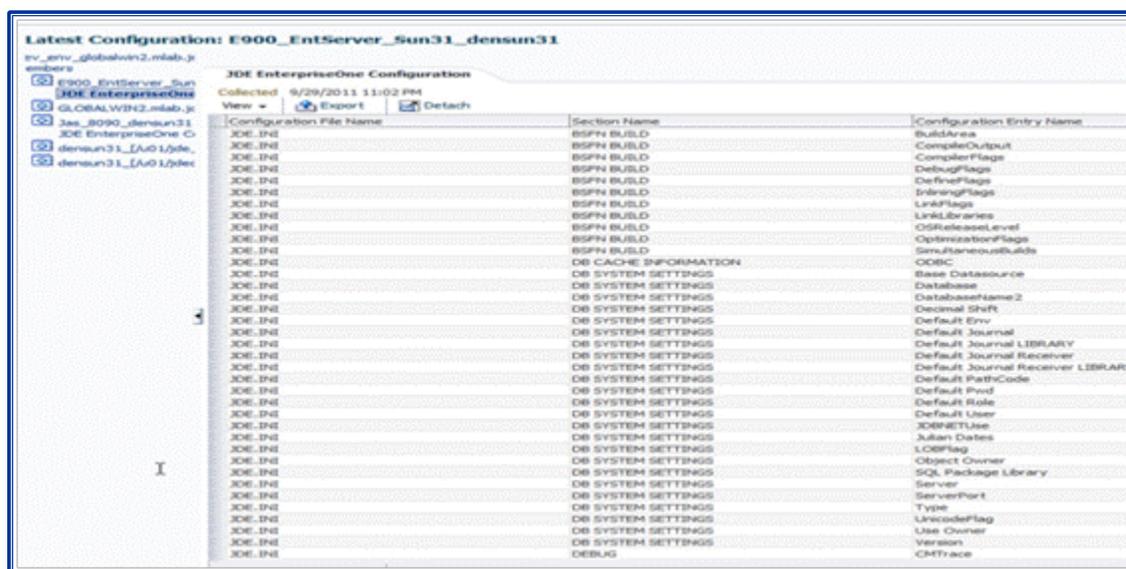


Figure 12 JD Edwards EnterpriseOne Configuration

## Configuration Comparison

Configuration comparison is useful to identify the drift and take corrective actions. You can do the comparison in ad-hoc fashion or periodically. System administrators can set up notification on comparison drift detection as well as create comparison templates. For more detail about setting up the configuration comparisons and template, see the EMCC [Guide](#).

**Use Case:** The long running transactions are failing on one of the JD Edwards EnterpriseOne HTML Servers and are constantly underperforming compared other instances of the server.

Using the configuration comparison, it can be identified that the '*TransactionTimeOut*' settings differ between the server instances. The following figure shows the Configuration Comparison results:

The screenshot displays the 'Compare Result' interface for a configuration comparison job. The job title is 'CONFIGURATION COMPARISON JOB THU SEP 29 17:08:02 MDT 2011'. Under 'Comparison Details', it shows two configurations: 'First Configuration: Jas\_8090\_densun31 (EnterpriseOne HTML Server)' collected on 2011-09-29 17:01:51 (Latest Configuration) and 'Second Configuration: Jas\_8090\_densun31 (EnterpriseOne HTML Server)' collected on 2011-09-29 16:44:26 (Saved Configuration). The 'Job Details' section has 'Show Differences Only' checked. A table below shows the comparison results for 'transactionTimeout', with the first configuration value at 60000 and the second at 120000. The table row is highlighted in red.

Result	ENTRY_NAME	FILENAME	SECTION_NAME	ENTRY_VALUE	Second Configuration
✖	transactionTimeout	jdkj.ini	JDE(SUNTIME PR...	60000	120000

Figure 13 JD Edwards EnterpriseOne Configuration Comparison

The above screen shows that the configuration comparison report was executed on two different instances of JD Edwards EnterpriseOne HTML servers. As highlighted in the report, the *Transaction Time* parameter is different between the two the instances and potentially could cause the transactions to time out on one of the HTML Servers.

The above example was for one-to-one comparison. Similarly in other scenarios, the EnterpriseOne HTML servers can be configured to understand the deltas in the configuration.

## Compliance Rules

You can set up the compliance rules to validate the name-value pair in any of the JD Edwards INI files that are within the range and when the values comply with the recommended Oracle Standards.

**Use Case:** To validate the browser session is closed and terminated when the user signs off.

This is useful to clear off hanging sessions and free up JAS memory.

A parameter called '*LogOutonBrowserClosed*' has to be set to true to ensure that sessions are closed. SQL-based rules can be set up to validate and flag a compliance error for a violation. Here is an example SQL that was used for the browser session closure use case:

SELECT

```
S1.TARGET_GUID,s2.SECTION_NAME,s2.FILENAME,s2.ENTRY_VALUE,s2.ENTRY_NAME FROM
CM$MGMT_ECM_ORACLE_JD Edwards_CONFIG
s2,MGMT$ECM_CURRENT_SNAPSHOTS s1gen1, MGMT$TARGET s1 WHERE
(s1gen1.TARGET_GUID = s1.TARGET_GUID AND (s1gen1.ECM_SNAPSHOT_ID = s2.ECM_SNAPSHOT_ID
AND ((NLS_UPPER(S2.ENTRY_NAME)LIKE NLS_UPEER('%'||'logoutOnBrowserClose' ||'%')))) AND
s1.TARGET_TYPE ='oracle_jde_e1_webserver' AND s1gen1.SNAPSHOT_TYPE
='oracle_jde_e1_webserver_configuration') AND NLS_UPPER(s1.TARGET_NAME)LIKE
NLS_UPPER('%'||'DENQAS7_JAS_82_DENQAS7'||'%')
```

The following figure shows a Configuration Compliance Rule Setup report:

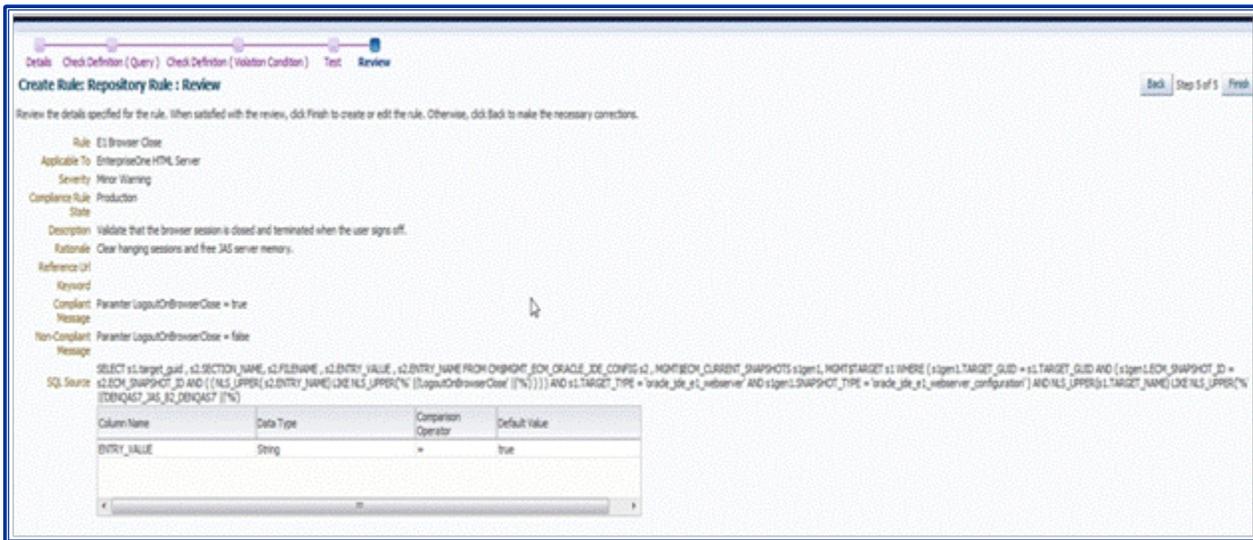


Figure 14 Configuration Compliance

## JD Edwards Groups

Groups enable the system administrators to collectively monitor and administer many targets as a single logical unit. Creating groups enables you to perform administrative operations, such as configuration changes for group members, incidents and problems, generated alerts, Compliance Summary, and Up/Down Status.

### Production Server and Development/Test Server —A Comparison

On several occasions, JD Edwards customers have servers from different environments (such as Production and Test) and want to cleanly isolate each environment for patching, performance analysis, or upgrades. In such cases, it is useful to identify components from one environment without intermixing with components from other environments. Oracle recommends that administrators define a group to contain all of the Enterprise Servers serving a production environment and define another group to contain all of the Enterprise Servers from development or test environments. In addition to isolating components, defining groups also helps in managing them as a single entity rather than individual entities. You can see all alerts related to components within a group on a single page, and you can navigate to a different member configuration from the group page. This simplifies the administration overhead and is less faulty.

The summary for group members, as shown in following screen, is a group created for all JD Edwards 9.1 release members and their availability status and membership details:

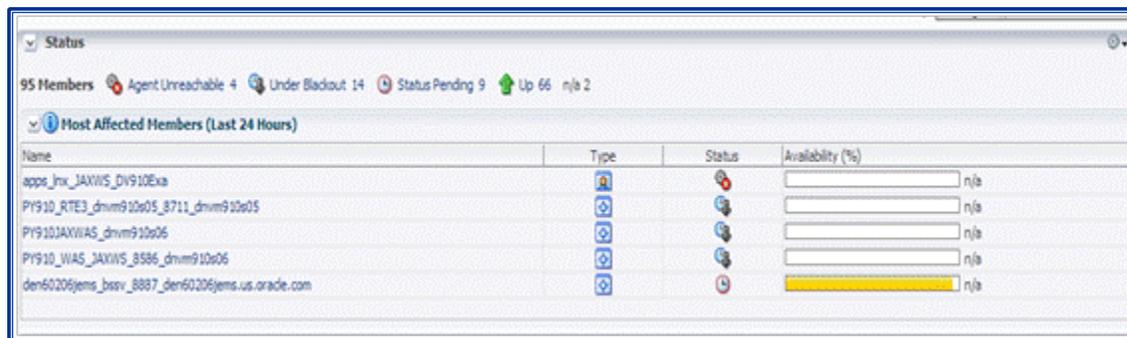


Figure 15 JD Edwards 9.1 Group Members

## Leveraging Incident Manager

As a JD Edwards administrator, you might have to deal with high loads on the Enterprise System caused by high traffic. In such a situation, you need to decide if you should migrate to another powerful machine with better computing and memory or add an extra node to the servers. A long-running business function or some other unnecessary process could possibly be utilizing the CPU. Therefore, you require more indicators in addition to the CPU utilization, such as memory utilization, table space details (if database is involved), disk utilization, and thread level details. These indicators enable you to isolate a system load condition from occasional spikes or a transient condition. Enterprise Manager, with its ability to manage events and incidents, can be leveraged in this situation.

An event is a significant occurrence of interest on a target that has been detected by Enterprise Manager. For example, the JD Edwards Enterprise Server target is not functioning or some threshold violation occurred. *Incident*, on the other hand, is a set of significant events or a combination of related events that pertain to the same issue. For details about incident management and other related concepts, see the [Guide](#).

### Incident Raised Out of Single Event —Threshold Violation

The simplest incident is composed of a single event. Here, you are concerned whenever any threshold violation event has occurred for the JD Edwards EnterpriseOne target. The incident is raised by Enterprise Manager when it detects that the event has occurred. The incident manager can then be used to track and manage its resolution.

The following figure shows an incident report:



Figure 16 Incident Generated From Single Event Debug Log File Size for Kernel Process

The above screen shows how both the incident and event attributes are used to help you manage the incident. Note that the incident is raised because the debug log file size for the JD Edwards EnterpriseOne Server target is reported to have a size of 97 MB for the kernel process, and it has exceeded its threshold of Warning 10 and Critical 25. An incident is opened that can be updated manually or by automated rules to set owners and status, as well as other attributes to track the issue and manage its lifecycle.

### Incident Raised Out of Multiple Events

Situations of interest can involve more than a single event. It is an incident's ability to contain multiple events that enables you to monitor and manage complex and more meaningful issues. For example, you are monitoring a JD Edwards host. If you want to monitor a load being placed on one or more hosts, you might be interested in events, such as CPU utilization, memory utilization, and swap utilization exceeding acceptable metric thresholds. Individually, these events may or may not indicate an issue with the host, but together, these events form an incident indicating that an extreme load is being placed on a monitored host.

The following figure shows an incident report that was generated for multiple events:

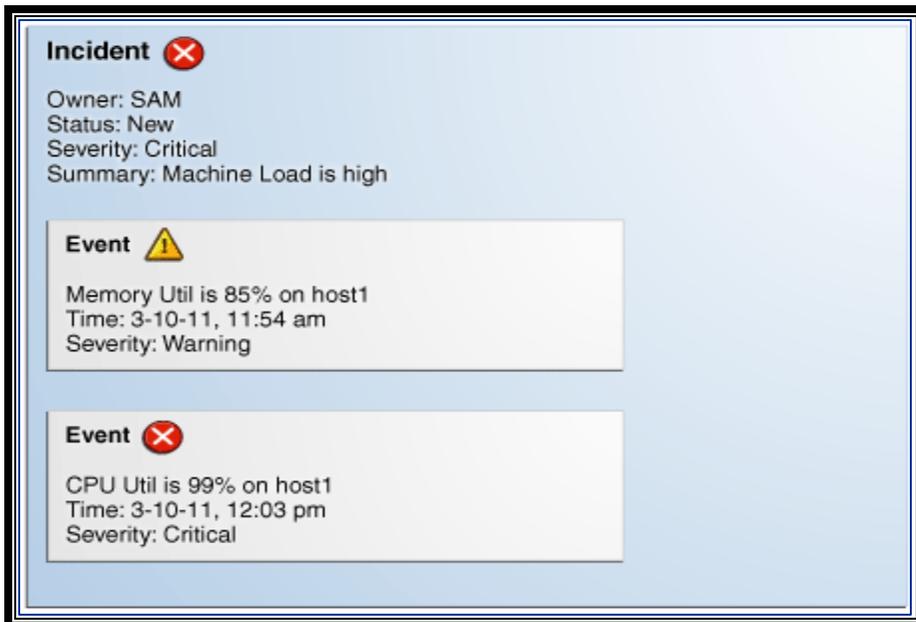


Figure 17 Incident Generated for Multiple Events on JD Edwards Hosts

Incidents inherit the worst severity of all the events. The incident summary indicates why this incident should be of interest (in this case, "Machine Load is high"). This message is an intuitive indicator for all administrators looking at this incident. By default, the incident summary is pulled from the message of the last event; however, any administrator working on the incident can change this message.

## Establish Service Level (SLA)

It is critical that the JD Edwards business applications are highly available and continue to offer various services to customers. It becomes important to monitor and manage at high standards of availability the application services and transactions that are constantly and automatically running behind the scenes and to avoid any performance degradations. Some examples of JD Edwards services include login service to one of JD Edwards web applications for availability and performance, and Purchase Order Transaction or Sales Order Transaction service to ensure that the transactions are going through from the system's perspective.

After the services are defined, you can define Service Levels (SLAs), which are used to evaluate service availability, performance, and usage. By constantly monitoring the service levels, JD Edwards customers can identify problems and their potential impact, diagnose root causes of service failure, and fix these in compliance with the SLAs. Monitoring a service helps to ensure that the operational goals and SLAs are met. For more detailed understanding of the concepts of service management, see the EM Service [Guide](#).

### JD Edwards Service Level Management (SLM)

Service level management involves:

- » Using OpenScript to record the JD Edwards Transaction. Create a Service Test in EM12c that uses the JD Edwards transaction.
- » Using EM12 to monitor and report on availability and performance metrics over a period of time. Measure performance against pre-defined levels (SLA).

Details of OpenScript are discussed in the [Guide](#). See this public [video](#) for recording a Purchase Order transaction.

## Generic Service

A generic service is the simple service model that can be created in EMCC. You can define one or more service models by defining service tests that represent a critical business function. JD Edwards Login Test is an example of a generic service. This service has a service test of type Application Testing Suite (ATS) called JD Edwards Login Test. The service test is defined as *Key Test*; that is, the availability of this service is based on the results of the associated service test. A system target is also associated with this service and the members of the system are marked as *key components*. Any component marked as *key* is used in computing the availability of the service and root cause analysis (RCA) if the service is down. The service is considered available as long as at least one or all key components are up and running, depending on the availability definition.

The following figure shows a Generic Service configured with JDE Login Test as Key Test:

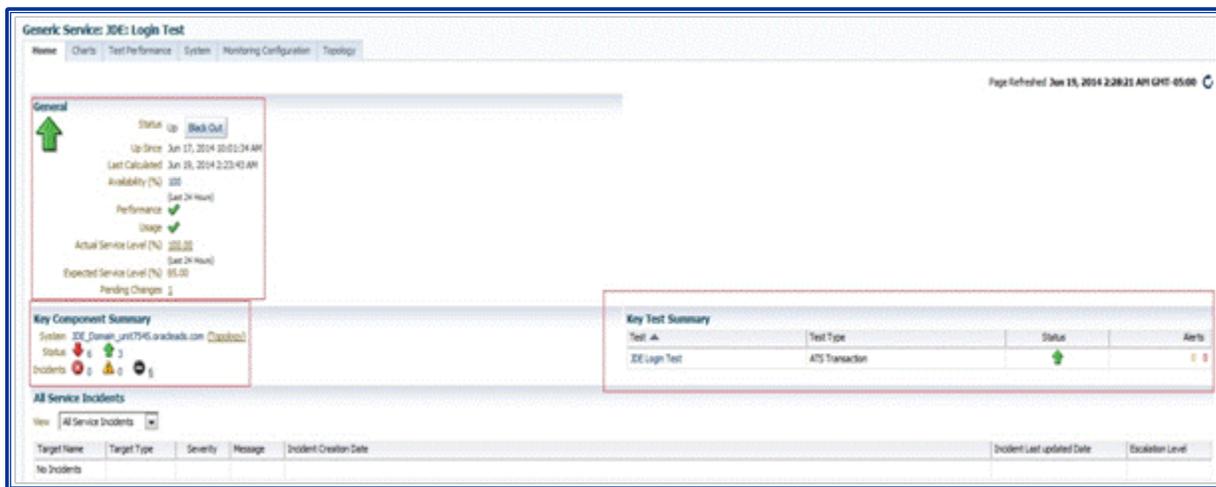


Figure 18 JD Edwards Generic Service with Key Test and Key Components

## Service Test

A service test is a synthetic test that emulates the way clients access the service (a user session). A service test consists of two parts, a test script and a test beacon, from which the script is executed. The following section shows a sample service test that was created to describe the underlying concepts. For specific details about service test creation see the [Guide](#).

The Login Service Test is an ATS transaction recorded by using the OpenScript tool, and it emulates the Login Transaction for the JD Edwards Capital Asset Management application. The steps are as follows:

1. Launch the JD Edwards Capital Assessment Application.
2. Log in to the application.
3. Look for a specific string in the application.
4. Log out.

During each of these steps, several critical performance metrics, such as login time, connect time, and so on, are computed. These statistics are useful to identify the exact performance bottlenecks and the swifter root cause analysis of any problem to ensure smoother business services.

The following figure shows a JD Edwards Login Test:

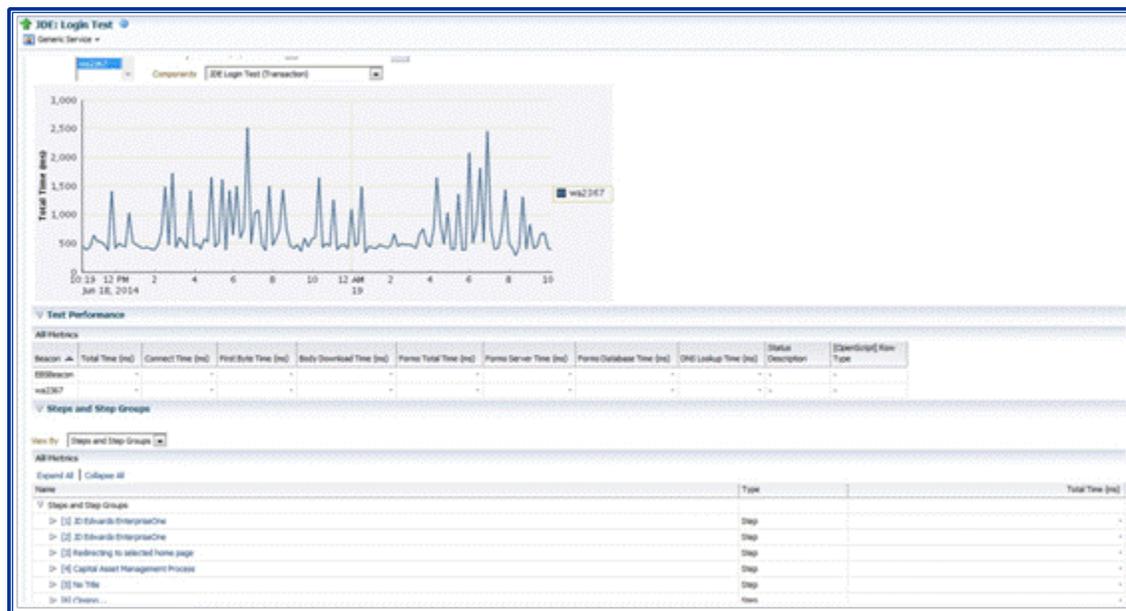


Figure 19 JD Edwards Key Test Performance and Metrics

**System**

A system is a set of infrastructure components that work together to host business functions. For example, for the JD Edwards log service to work, it is critical that the underlying infrastructure targets (such as JD Edwards Enterprise Server, EnterpriseOne Manage Home, HTML Server, Business Services Server, and hosts) all work together. If a single one of them is down, it can impact the login service availability.

The following figure shows the dependency between services, the system, and the system components:

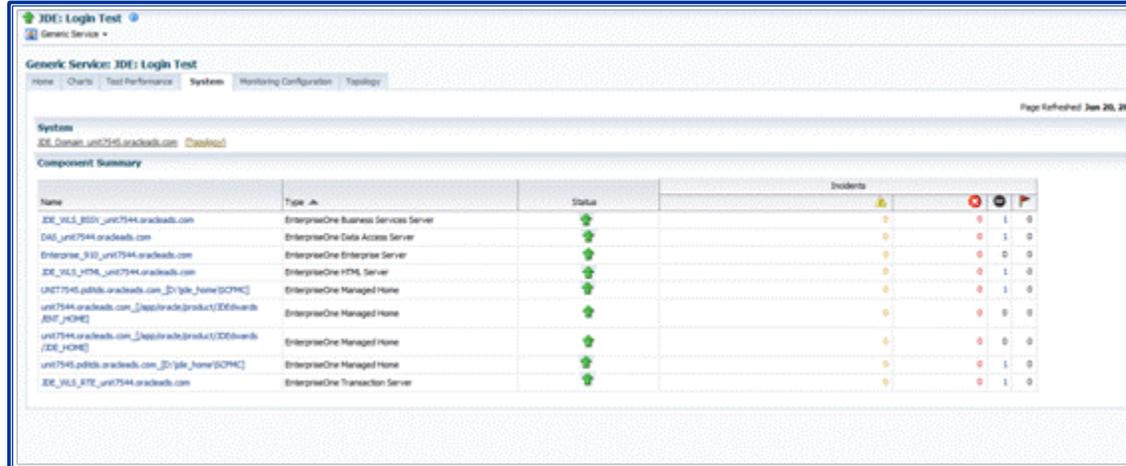


Figure 20 JD Edwards System for the Service

**Metric Dependencies in Service Targets**

Users can promote one or more metrics from the underlying system components to measure the performance of a service. These metrics can be added to any of the following categories: performance, usage, or business.

## Service Level Agreements

A Service Level Agreement (SLA) is a contract between the business service provider (JD Edwards hosting services or JD Edwards customer's IT department) and the customer defining the expected quality of service for a business period. An SLA can have one or more Service Level Objectives (SLOs) that define the service objectives to be provided. The SLA is a logical grouping of individual measurable indicators.

Service Level Indicators (SLIs) are measurable and quantifiable service metrics that can be used to evaluate the quality of a service. By selecting metrics and setting objectives to an SLA, you can determine the health and status of the service. You can create an agreement of how much time *SLO not met* is acceptable when there is a breach of service level agreement<sup>[1]</sup>.

You can use the service dashboard to monitor the status of a service, SLA, and SLO. By looking at the screen, you can correlate the SLA and its associated metrics.

The following figure shows the service level indicators as tracked by JD Edwards Login Test:

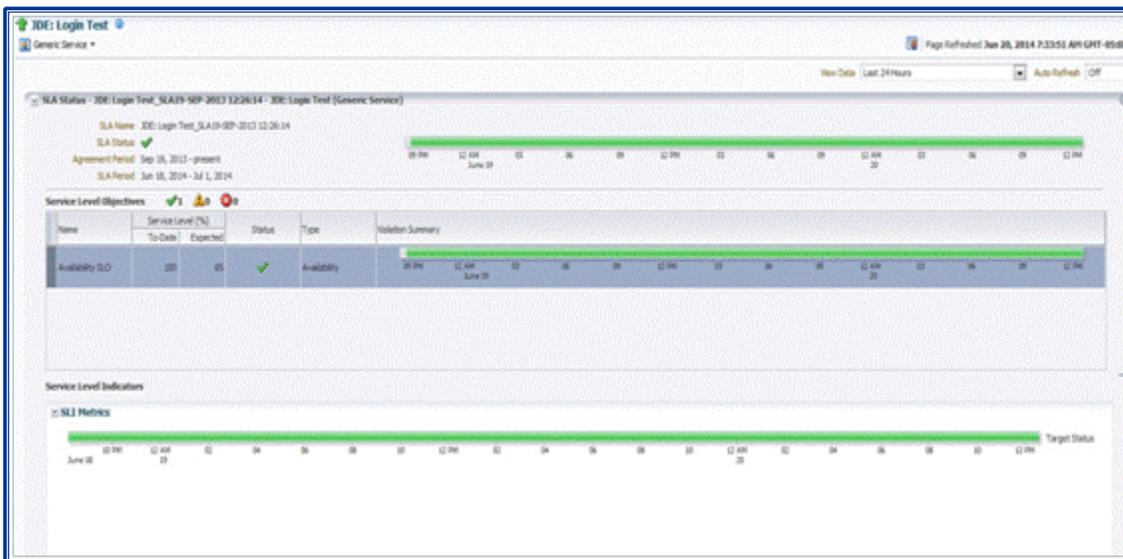


Figure 21 JD Edwards Login Service Dashboard with SLO and SLI Metrics

## SLA Dashboard Report

EMCC ships different out-of-the box reports, and one of them is for Services Dashboard. The reports display the overall health of the different services and the SLAs and are useful for executives to get a comprehensive view of the system.

The following figure shows a JD Edwards Services Dashboard Report:

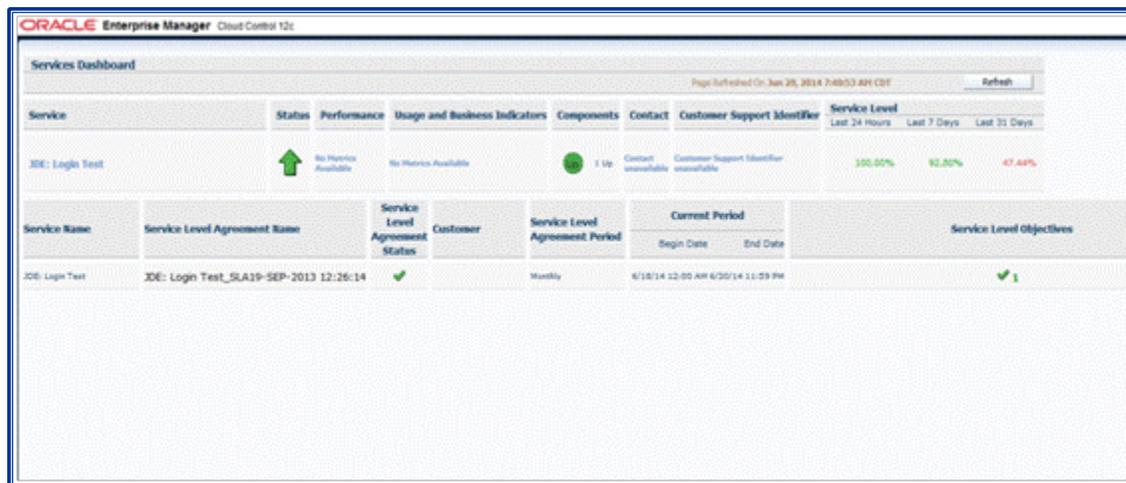


Figure 22 JD Edwards Services Dashboard Report

## Oracle Real User Experience Insight (RUEI)

Oracle Real User Experience Insight (RUEI) moves service delivery monitoring from the Data Center's perspective to the end user's perspective. RUEI's ability to see what the end-users experience, together with its powerful reporting facilities, enables direct insight into every component of even the most complex infrastructures and delivers an information foundation that services both IT and business. The session diagnostics facility supports performing root-cause analysis of operational problems. It enables you to isolate transactions for specific users without any performance overhead by using the passive monitoring ability.

Oracle RUEI for JD Edwards EnterpriseOne provides you with powerful analysis of your network and JD Edwards business applications. You can monitor the real-user experience, set Key Performance Indicators (KPIs), and automatically trigger alert notifications for incidents that violate them. Oracle RUEI comes with a library of powerful reports that provide both business-oriented and technical-oriented users with the information they need to make effective decisions.

### JD Edwards EnterpriseOne RUEI Accelerator

Oracle RUEI Accelerator for JD Edwards EnterpriseOne provides an out-of-box real-user monitoring solution that automatically discovers all used JD Edwards applications and translates network objects into business functions.

This ability enables the measurement and monitoring of real-user transactions from clicks to actions into business transactions. Each individual user action is matched to the correct form/responsibility/framework and action to provide contextual analyses. Oracle RUEI Accelerators for AMS4JDE capture and report all functional errors and discover end-user performance and application issues. Application managers can set up KPIs and report views of interest on their mission-critical applications.

JD Edwards EnterpriseOne is an integrated applications suite of a comprehensive enterprise resource planning software. It contains over 80 application modules, ranging from Financial Management, Project Management, and Order Management to Manufacturing Management running on a desktop, iPad®, and mobile platform and supports hundreds and thousands of users. For application suites that support diverse infrastructure components, it is critical that the application performance is measured by the end user and any performance bottleneck identified and removed.



The following discussion highlights a few challenges faced by organizations using JD Edwards EnterpriseOne and how Oracle RUEI can be used to monitor every user of the enterprise environment, track their experience, and improve the performance of the enterprise. For details about how to configure and set up RUEI see the configuration and install [Guide](#).

### **Problem Statement # 1 Proactive, Unobtrusive, and Secure Monitoring of JD Edwards Applications**

Proactive management of business-critical applications requires monitoring the entire application environment to capture the actual (not the Data Center's view) end-user experience of all users, at all locations, at all times, across the entire application suite. To achieve proactive, unobtrusive, and secure monitoring, RUEI employs a three-layered product architecture that consists of:

- » Data Collection
- » Data Processing
- » Data Presentation

#### **Data Collection**

RUEI provides end-to-end monitoring based on network protocol analysis (NPA). This analysis uses a process of decoding network protocol headers and trailers that is industry-standard, secure and unobtrusive, and has no negative impact on network performance. Each incoming page request is captured and matched with its outgoing response. The response time and status are stored in a repository for subsequent analysis and tracking.

RUEI Data Collector is responsible for acquiring the raw data from different sources. There are two connection options for the Data Collector: utilizing the span port of a network switch, or through a specialized tap device (passive) that replicates the network traffic in a read-only mode<sup>[4]</sup>. Both of these approaches have their pros and cons. The span port of a network switch does not require any additional hardware component between the switch and the RUEI collector, but there is a possibility that some packets might be dropped and there might be a loss of monitoring data. The specialized tap device (passive) requires additional hardware but there is no packet lost. For more details, see the RUEI [Guide](#).

To read HTTP(s) data streams, a proprietary software module reassembles TCP/IP packet streams. The operation system's TCP/IP stack is disabled for those network interfaces that are used to collect data. Because these network interfaces do not have an assigned IP number and the software using these network interfaces does not have a functional IP stack, RUEI is not able to respond to incoming traffic over these network interfaces. This makes RUEI invisible to the monitored networks and completely secure.

Data collection can be configured to log encrypted data (HTTPS). To facilitate this, a copy of the web server's private SSL keys needs to be set up in the Collector. In addition, RUEI can be configured to omit logging of sensitive data (also called masking) in POST URL arguments, HTTP headers, cookies, and the content of URLs.

#### **Data Processing**

The processor converts raw data into relevant online analytic processing (OLAP) data sets (or views). At installation time, the correct views are defined and configured. Page identification within RUEI is based on applications. Essentially, an application is a collection of web pages. This is because pages on a website are typically bound to a particular application. For each page that the system detects, the RUEI uses the available application definitions to assign a name to it.

#### **Data Presentation**

The data presentation is performed by RUEI's analysis and reporting environment. This is a web-based information portal that can be accessed from any supported browser.

## Problem Statement # 2 Tracing User Transactions and Identifying Bottlenecks

Typically, support desks receive calls that an application is responding very poorly and pages are taking a lot of time to load. Sometimes it becomes difficult to find out the actual challenge that the user is facing and identify the root cause of the problem.

Using RUEI, the system administrator can diagnose this problem in an efficient manner. The following steps highlight what an administrator must do to identify the problem a user (JD Edwards in this case) is facing when accessing one of the application pages.

1. Use the RUEI dashboards to view users on the dashboard. You can observe the user analysis, the problematic forms, and the top application page hits. The following figure shows the JD Edwards application dashboard:

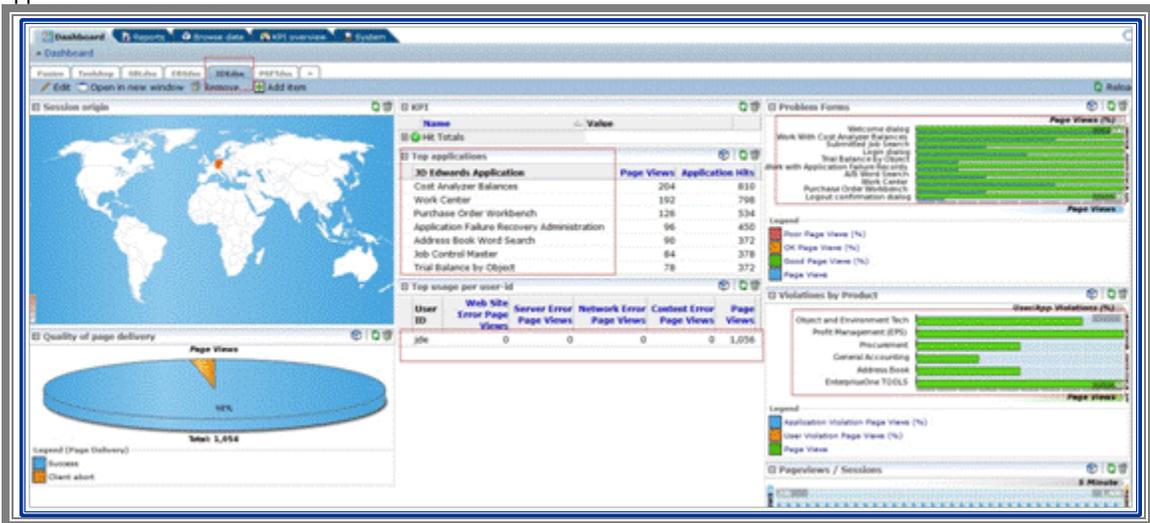


Figure 23 RUEI JD Edwards Application Dashboard

2. Set the filters on the User (JD Edwards) to isolate the user-specific transaction.

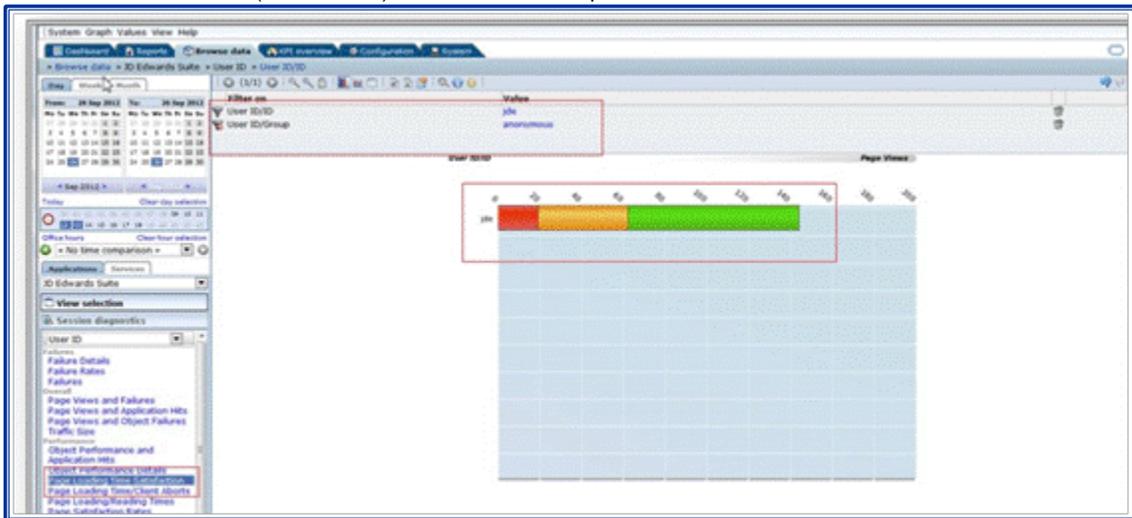


Figure 24 RUEI Data Filtered by Specific User

3. Look for the user activity to identify the problematic transaction for the user and identify all the steps in the session. The performance data for each step is color-coded by satisfaction level. Also, you can

identify errors (colored red) at every step as shown in the following screen:

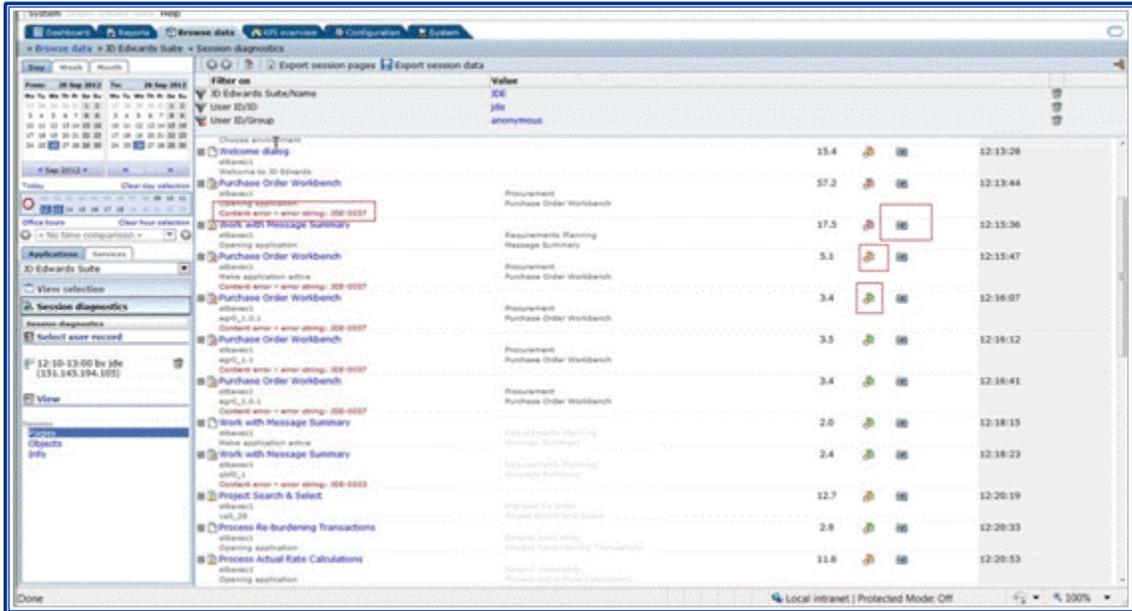


Figure 25 Errors in User Transaction

### Problem Statement # 2 Root Cause Analysis for Problematic Transaction

Having identified problematic transactions and errors, it is critical to understand what caused them. RUEI can be used in conjunction with Server Manager to identify active sessions for the user, open applications, and process-level details for the user session. Currently, this is a manual step and requires you to copy the user name and search the active user session in Server Manager.

The following figure shows the active user sessions:

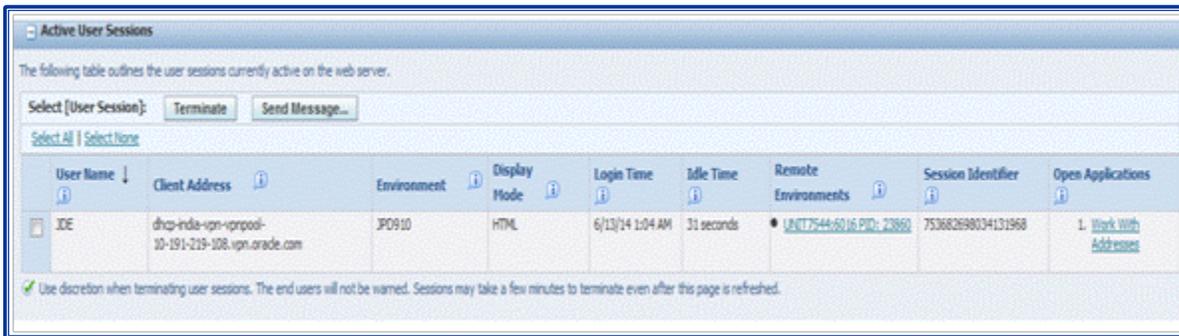


Figure 26 Active User Sessions for Diagnostics



## Conclusion

AMS4JDE for Enterprise Manager 12c Cloud Control (EM 12c) provides some of the most advanced features to manage and own JD Edwards components and the underlying infrastructure. Effective leveraging of EM12 features for metrics collection, service level agreement, configuration management, and RUEI integration can dramatically reduce the amount of time that system administrators spend on the management of JD Edwards EnterpriseOne.

The benefits of using AMS4JDE to manage JD Edwards Applications are substantial, and include the following:

- » Manage all of the JD Edwards components from a single management console.
- » Perform Root Cause Analysis (RCA).
- » Experience with RUEI the end-user insights, session diagnostics, and transaction management.
- » Easily set up SLAs for critical business applications and roll up the service level details to executive dashboards.
- » Define configuration comparison reports and track compliance.



## References

1. Oracle® Enterprise Manager Cloud Control Administrator's Guide 12c Release 4 (12.1.0.4)
2. JD Edwards EnterpriseOne Tools Server Manager Guide Release 9.1.x
3. JD Edwards EnterpriseOne Application Pack for Oracle Enterprise Manager Cloud Control Implementation Guide Release 9.1
4. Oracle Real User Experience Insight Implementation Case Study – Monitoring Oracle Global Single Instance <http://www.oracle.com/technetwork/oem/grid-control/overview/twp-ruei-gsi-casestudy-131363.pdf>



**Oracle Corporation, World Headquarters**

500 Oracle Parkway  
Redwood Shores, CA 94065, USA

**Worldwide Inquiries**

Phone: +1.650.506.7000  
Fax: +1.650.506.7200

CONNECT WITH US



[blogs.oracle.com/oracle](http://blogs.oracle.com/oracle)

[facebook.com/oracle](http://facebook.com/oracle)

[twitter.com/oracle](http://twitter.com/oracle)

[oracle.com](http://oracle.com)

**Hardware and Software, Engineered to Work Together**

Copyright © 2014, Oracle and/or its affiliates. All rights reserved. This document is provided for information purposes only, and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document, and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

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 of The Open Group. 1014

White Paper Title: Oracle Best Practices for Utilizing Application Management Pack for JD Edwards EnterpriseOne with Enterprise Manager Cloud Control 12c

October 2014  
Author: Anand Chandak



Oracle is committed to developing practices and products that help protect the environment