

JD Edwards EnterpriseOne

In-Memory Applications Configuration Guide for Oracle
Engineered Systems

Release X9.1.3.2

E40602-04

April 2014

E40602-04

Copyright © 2011, 2014, 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 END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

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 of The Open Group.

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	v
Audience	v
Documentation Accessibility	v
Conventions	v
 1 Understanding JD Edwards EnterpriseOne In-Memory Applications	
1.1 Understanding This Guide	1-1
1.2 Certifications (Formerly Known as Minimum Technical Requirements)	1-1
1.3 JD Edwards EnterpriseOne Architecture	1-2
1.3.1 VM Configuration	1-2
1.3.2 Separation of Interactive Applications from Batch Jobs	1-2
 2 In-Memory Application Specific Tuning Configurations	
2.1 Project Portfolio Management	2-1
2.1.1 Clearing Work Tables for Project Portfolio Management	2-1
2.1.2 Tuning Recommendations for Project Portfolio Management	2-2
2.1.2.1 Number of Users and Accounts (Jobs)	2-2
2.1.2.2 Billing vs Non-Billing	2-2
2.1.2.3 Use of 'View Later'	2-3
2.2 Sales Advisor	2-3
2.2.1 Clearing Work Tables for Sales Advisor	2-3
2.2.2 Tuning Recommendations for Sales Advisor	2-3
2.2.2.1 P42X00 configuration considerations - Configuring ON/OFF	2-3
2.2.2.2 Sales Advisor - Time Range for Cross-Sell	2-4
2.2.2.3 Sales Advisor - Order Advisor Customer versus Customer Group	2-4
 3 Configuring the JDE.INI File for the JD Edwards EnterpriseOne Enterprise (Logic) Server	
 4 Oracle WebLogic Application Server and the JD Edwards EnterpriseOne HTML Server	
4.1 Configuring Oracle WebLogic	4-1
4.2 WebLogic Server Startup Parameters for Managed Instances	4-3

5 Tuning the Oracle Database Machine

5.1	Primary Indexes	5-1
5.1.1	Project Portfolio Management	5-1
5.1.2	Sales Advisor	5-2
5.2	Table Pinning in Flash Memory on the Oracle Database Machine	5-2
5.2.1	Discovering Suitable Candidates for Table Pinning.....	5-2
5.2.2	Manually Pinning Table into Flash Cache Memory	5-3
5.3	Configuring the Oracle Exadata Database Server.....	5-3
5.3.1	Example - Alter Commands.....	5-4
5.3.2	Example - Database Settings After 'alter' Commands.....	5-4
5.3.3	Oracle Shared and Dedicated Connections	5-5
5.3.4	Scaling Project Portfolio Management and Sales Advisor Using Oracle Database Tuning Settings 5-6	
5.3.5	Table and Index Partitioning for the Project Portfolio Management In-Memory Application 5-7	
A.1	Selecting Additional Fields.....	A-1
A.2	Clearing the Form	A-2

Preface

Welcome to the *In-Memory Applications Configuration Guide for Oracle Engineered Systems*.

Audience

In-Memory Applications are features available to JD Edwards EnterpriseOne on the engineered Oracle Exalogic platform consisting of the Exalogic Elastic Cloud and the Exadata Database Machine. This document describes additional configuration and setup for the EnterpriseOne Logic Server, Exadata Database Machine, and WebLogic HTML Servers using the EnterpriseOne In-Memory applications on the Oracle Exalogic and Exadata platform. The guide was developed on the Exadata and Exalogic X3-2 engineered systems.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
<code>monospace</code>	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Understanding JD Edwards EnterpriseOne In-Memory Applications

This chapter contains the following topics:

- [Section 1.1, "Understanding This Guide"](#)
- [Section 1.2, "Certifications \(Formerly Known as Minimum Technical Requirements\)"](#)
- [Section 1.3, "JD Edwards EnterpriseOne Architecture"](#)

1.1 Understanding This Guide

This guide is designed to provide configuration suggestions which were derived from test environments for JD Edwards EnterpriseOne systems built for Oracle engineered systems. These recommendations are based on the recommended system configuration where the Enterprise Server and HTML Server are running in Oracle VM Templates for JD Edwards EnterpriseOne and that database server is an Exadata machine. These settings have not been tested in production environments, and are provided as a basis of determining your own values.

For details on installing JD Edwards EnterpriseOne using Oracle VM Templates on Exalogic and Exadata, refer to the [Oracle VM Templates Express Installation Guide Release 9.1 for Oracle Exalogic Platforms](#).

Sections of this guide are devoted to the major components of the EnterpriseOne architecture, namely the EnterpriseOne Logic Server, WebLogic HTML Server, and Oracle Database Machine components.

1.2 Certifications (Formerly Known as Minimum Technical Requirements)

Customers must conform to the supported platforms for the release as detailed in the Oracle Certifications (formerly known as Minimum Technical Requirements) for JD Edwards EnterpriseOne In-Memory Applications. In addition, JD Edwards EnterpriseOne may integrate, interface, or work in conjunction with other Oracle products. Refer to the following link for cross-reference material in the Program Documentation for Program prerequisites and version cross-reference documents to assure compatibility of various Oracle products.

<http://www.oracle.com/corporate/contracts/index.html>

Refer to the Certifications tab on My Oracle Support and search for JD Edwards EnterpriseOne certifications with In-Memory in the title.

Within the Certification Search, navigate to these JD Edwards EnterpriseOne In-Memory products:

- JD Edwards Tools for In-Memory Applications
- JD Edwards Enterprise Server for In-Memory Applications
- JD Edwards Database Server for In-Memory Applications
- JD Edwards HTML Server for In-Memory Applications

Note: This Certification also includes the WebLogic Application Server on which the JD Edwards HTML server runs.

1.3 JD Edwards EnterpriseOne Architecture

This section describes these topics:

- [Section 1.3.1, "VM Configuration"](#)
- [Section 1.3.2, "Separation of Interactive Applications from Batch Jobs"](#)

1.3.1 VM Configuration

The JD Edwards EnterpriseOne installation allows the architecture to be configured using Oracle VM templates for JD Edwards, which are available on the Oracle e-delivery website. The use of Oracle VM templates allows rapid deployment of most of the JD Edwards EnterpriseOne components including the EnterpriseOne logic, batch, HTML (JAS), database, and deployment servers.

In an environment that uses Oracle VM templates, the EnterpriseOne templates allows a more granular configuration and consumption of hardware resources. This partitioning of hardware through virtualization is a powerful tool in controlling the resources available.

1.3.2 Separation of Interactive Applications from Batch Jobs

The flexibility of the JD Edwards EnterpriseOne architecture enables interactive and batch processing to be configured on separate servers, providing a scalable and configurable platform.

For the purpose of discussing the JD Edwards EnterpriseOne in-memory applications, it is suggested that the in-memory applications processing load resides on separate servers for best interactive performance. That is, it is recommended that a separate JD Edwards EnterpriseOne logic server be dedicated for the in-memory applications while the JD Edwards EnterpriseOne batch processing is recommended to be executed on a different logic server. In addition to using physical machines, this logical server separation may be implemented using Oracle VM templates.

In-Memory Application Specific Tuning Configurations

This section provides specific tuning configuration advice for these JD Edwards EnterpriseOne in-memory applications:

- [Section 2.1, "Project Portfolio Management"](#)
- [Section 2.2, "Sales Advisor"](#)

2.1 Project Portfolio Management

Project Portfolio Management is a JD Edwards EnterpriseOne in-memory application that utilizes the greater processing and storage capacity of the Oracle engineered systems. More details on the Project Portfolio Management application can be found in the following link:

<http://www.oracle.com/us/products/applications/jd-edwards-enterpriseone/in-memory-ppm-1943157.pdf>

This section describes these tasks:

- [Section 2.1.1, "Clearing Work Tables for Project Portfolio Management"](#)
- [Section 2.1.2, "Tuning Recommendations for Project Portfolio Management"](#)

2.1.1 Clearing Work Tables for Project Portfolio Management

The Project Portfolio Management in-memory application uses a set of work tables for optimal efficiency and performance of the applications. Generally these work tables automatically clear themselves as part of normal processing. However in some instances, the work tables used to store temporary application data may require manual clearing to ensure optimal performance. It is recommended that you periodically truncate the following Sales Advisor work tables, from an Oracle SQLPlus session or through the Oracle Enterprise Manager:

- F42X010W
- F42X020W
- F42X040W
- F42X041W
- F42X050W
- F42X060W

2.1.2 Tuning Recommendations for Project Portfolio Management

This section describes the tuning of these aspects:

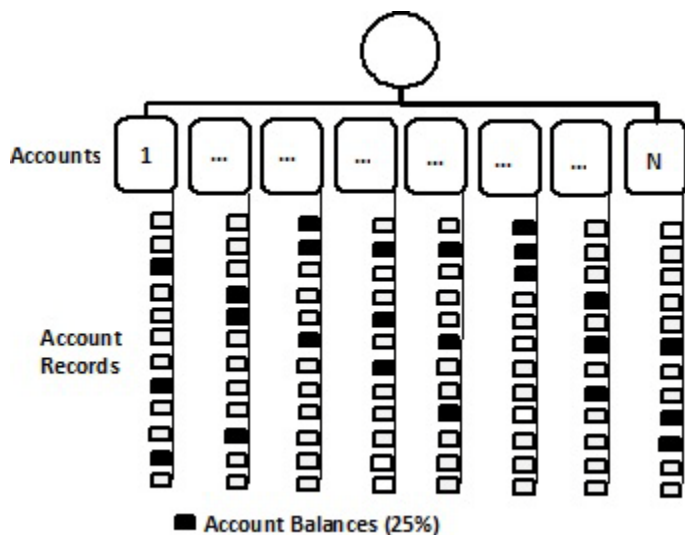
- [Section 2.1.2.1, "Number of Users and Accounts \(Jobs\)"](#)
- [Section 2.1.2.2, "Billing vs Non-Billing"](#)
- [Section 2.1.2.3, "Use of 'View Later'"](#)

2.1.2.1 Number of Users and Accounts (Jobs)

These are the main factors that affect performance in the Project Portfolio Management application:

- The number of concurrent users
- The number of accounts that users will attempt to access

Normally there is a consistent end-user experience with the Project Portfolio Management application, however the number of users and the number of accounts can affect performance. The number of accounts is directly related to the advanced query in the Portfolio Management application where multiple job numbers can be specified in the Project Portfolio Management application grid. Each of these jobs will have a number of accounts associated with them. The following graphic outlines the relationship of accounts to account balances.



In the above example, each user through the job advance query will be accessing a number of accounts and each of these accounts will have a number of account records. The more dense the account records are, the more processing occurs in the Project Portfolio Management application. In this example a 25% profile of account records per account accessed is shown.

2.1.2.2 Billing vs Non-Billing

Specifying billing in the Project Portfolio Management application or the use of 'BILL' or 'REC' as a category specification will increase the amount of work the Project Portfolio Management application performs. In general, the inclusion of billing roughly doubles the processing performed within Project Portfolio Management.

2.1.2.3 Use of 'View Later'

Project Portfolio Management has a flexible interface providing many options for viewing projects such as those discussed in the preceding sections (such as users, jobs, and enabling billing categories). The more information that is displayed, the longer and more process intensive the application becomes. Process time outs can occur when too much information is requested or load is encountered by the application. In these circumstances, it is suggested that the 'View Later' functionality be utilized to off load these more data intensive requests.

2.2 Sales Advisor

The JD Edwards EnterpriseOne in-memory Sales Advisor adds additional functionality to the sales order application by using the enhanced processing and storage capacity of the Oracle Engineered systems. More details on the Sales Advisor application can be obtained through the following link:

<http://www.oracle.com/us/products/applications/jd-edwards-enterpriseone/jde-in-memory-sales-advisor-1931628.pdf>

This section describes these tasks:

- [Section 2.2.1, "Clearing Work Tables for Sales Advisor"](#)
- [Section 2.2.2, "Tuning Recommendations for Sales Advisor"](#)

2.2.1 Clearing Work Tables for Sales Advisor

The Sales Advisor in-memory application uses a set of work tables for optimal efficiency and performance of the applications. Generally these work tables automatically clear themselves as part of normal processing. However in some instances, the work tables used to store temporary application data may require manual clearing to ensure optimal performance. It is recommended that you periodically truncate the following Sales Advisor work tables, from an Oracle SQLPlus session or through the Oracle Enterprise Manager:

- F51X006
- F51X0902
- F51X0911
- F51X10
- F51X21
- F51X23

2.2.2 Tuning Recommendations for Sales Advisor

This section describes these tuning configuration considerations for the in-memory Sales Advisor application:

- [Section 2.2.2.1, "P42X00 configuration considerations - Configuring ON/OFF"](#)
- [Section 2.2.2.2, "Sales Advisor - Time Range for Cross-Sell"](#)
- [Section 2.2.2.2, "Sales Advisor - Time Range for Cross-Sell"](#)

2.2.2.1 P42X00 configuration considerations - Configuring ON/OFF

The Sales Advisor configuration parameters that are configurable can be controlled by the single application of P42X00. A number of features can be specified that will affect

the performance of Sales Advisor. The following is a list of features that can be toggled ON/OFF for the Sales Advisor in-memory application:

- Cross-Sell
- Frequency
- Order Fulfillment
- Order Profit
- Order UpSell
- Supply and Demand

When ENABLED, these features have an effect to performance. The parameter specifications for the cross-sell time range and the order advisor customer versus customer group are two particular tuning considerations that are described in the following sections of this guide.

2.2.2.2 Sales Advisor - Time Range for Cross-Sell

In general, the time range for cross-sell, or the time frame which can be specified to look for cross-sell opportunities typically range from 1 to 3 months up to a 6-month period. The larger the time-frame, the more data that much be analyzed to find the cross-sell opportunities.

For performance considerations, the recommended setting for the cross-sell time range is between 1 to 3 months.

2.2.2.3 Sales Advisor - Order Advisor Customer versus Customer Group

Order Advisor may be more efficient when specifying a customer instead of a customer group in the P42X00 application. Customer groups are more process intensive because they process data across the customer group as opposed to the specific customer.

Configuring the JDE.INI File for the JD Edwards EnterpriseOne Enterprise (Logic) Server

This section provides initial recommended configuration parameters stored in the JDE.INI file for the JD Edwards EnterpriseOne Enterprise (logic) Server. These settings represent values to consider for configuring EnterpriseOne In-Memory Applications running on the Oracle engineered systems. These settings are based on a load of 2,000 concurrent interactive users. You should use the basic configuration rules provided in the comments for scaling above the 2,000 user limit and you should refrain from configuring any of the settings below the default values.

Note: You should always use Server Manager to set the values in the JDE.INI file for the JD Edwards Enterprise logic server. The settings provided in this guide are a good starting point for properly configuring EnterpriseOne running on the Oracle Exalogic platform.

The following general rules may not always apply to all industry verticals and EnterpriseOne implementations. They have found to be provide good performance characteristics during lab testing of EnterpriseOne on the Oracle Exalogic engineered systems. The In-memory applications require a slight deviation from the default best tuning practices.

Note: For simplicity, changes to specific sections of the JDE.INI for performance are provided below.

- **JDENET network processes**

The best practice to determine the number of network process remains the same, 100 users per netprocess + 1.

- **DEF4 Security Kernels**

Similarly, the best practice for the number of security kernels, 100 users per security_kernel + 1.

- **DEF6 Call Object Kernel**

The DEF6 process for Call Object Kernels (COK) is an area in the EnterpriseOne logic server that requires a deviation from the standard tuning recommendations. The JD Edwards EnterpriseOne tools code has been optimized for the JD Edwards EnterpriseOne In-memory applications. Starting with Tools code X9.1.3.1 for Oracle engineered systems, a change in the number of database connections the COK process can establish has been increased. For JD Edwards EnterpriseOne

In-memory applications, child threads can establish their own independent database connections. Prior to this change, only the master business function was allowed to establish database connections. This change increases the consumption of resources that COKs require. A ratio of five users per COK is recommended.

A second tuning value is the number of threads per session. Testing indicates that the default value of 10 is the proper setting for both In-memory applications. If Project Portfolio Management is the predominate application used, then a setting of five may provide a slight boost to performance.

Following is a snapshot of a JDE.INI file with the above recommended setting changes.

```
[JDENET]
## The maxNetProcesses is derived by assuming 100 users per netProcess + 1
## (main jdenet process)
## For example 2000 users would be 20 + 1 or 21 total processes
maxNetProcesses=21
maxNetProcesses=6
maxNetConnections=2000
maxNumSocketMsgQueue=800
maxIPCQueueMsgs=400
maxLenInlineData=4096
maxLenFixedData=16384
maxFixedDataPackets=2000
```

The following are configuration settings for the Security kernels:

```
[JDENET_KERNEL_DEF4]
## The maxNumberOfProcesses is derived assuming 100 users per security
## kernel +1.
## For example 21 security servers will handle 2000 users
krnlName=SECURITY KERNEL
dispatchDLLName=libjdeknet.so
dispatchDLLFunction=JDEK_DispatchSecurity
maxNumberOfProcesses=21
numberOfAutoStartProcesses=0
```

The following are configuration settings for the Call Object kernels:

```
[JDENET_KERNEL_DEF6]
krnlName=CALL OBJECT KERNEL
dispatchDLLName=libxmlcallobj.so
dispatchDLLFunction=XMLCallObjectDispatch
## Processes setting for this kernel assumes 2000 users where the ratio is 10:1
## for NON-In-memory applications
## For In-memory applications, set this to a 5:1 ratio
maxNumberOfProcesses=400
numberOfAutoStartProcesses=0
singleThreadedMode=N
ThreadPoolSize=30
ThreadPoolSizeIncrement=5
## In-memory tuning value, default is 10. Change to 5 if Project Portfolio
## Management is the predominate application.
```

Summary

The combined configuration of the JDENET section of the JDE.INI for the network processes, the EnterpriseOne kernel settings defining the Security Server and the Call Object kernels, provided good scaling to 2,000 concurrent users with the in-memory

applications with slightly adjusted changes to the JD Edwards EnterpriseOne configuration

Oracle WebLogic Application Server and the JD Edwards EnterpriseOne HTML Server

This section provides some recommended parameters for the Oracle WebLogic Application Server and the JD Edwards EnterpriseOne HTML Server which supports it. These settings are representative of the changes to consider. The recommended values are based on evaluating the performance and scalability of loads up to 2,000 concurrent users.

This section discusses the recommendations for changing the WebLogic parameters for use with the in-memory applications of Project Portfolio Management and Sales Advisor:

- [Section 4.1, "Configuring Oracle WebLogic"](#)
- [Section 4.2, "WebLogic Server Startup Parameters for Managed Instances"](#)

4.1 Configuring Oracle WebLogic

To further configure the Oracle WebLogic and JD Edwards HTML Servers for better performance:

1. From the WebLogic Admin Console, reset the JVM arguments used for this managed server instance with the following startup parameters:

```
-Djava.net.preferIPv4Stack=true -Doracle.net.SDP=true -Xms4096m -Xmx4096m  
-XX:MaxPermSize=128m -DcloneId=C1_VC1
```

Note: Refer to subsequent notes in this section for additional Server Startup parameter considerations.

2. From the WebLogic Admin console, turn off HTTP logging:

```
<managed_instance> > Logging > HTTP > uncheck HTTP access log file enabled
```

3. From the WebLogic Admin console, enable the Exalogic Optimization feature:

```
<domain_name> > General > enable checkbox for Enable Exalogic Optimizations
```

4. From the WebLogic Admin Console and the EnterpriseOne Server Manager Console, increase the number of Managed Server Instances.

Test results indicated a good starting point is to assume 200 users per JVM. Given that assumption, for a 2,000 user load, create ten (10) JVMs for each pathcode (for example, DV). To do so, add the first node, configure the EnterpriseOne tools for

that clustered node and then add additional managed server instances after the first node has been configured.

Note: A single pathcode cannot point to a mixture of cluster managed instances and stand-alone managed instances concurrently. For the greatest flexibility, and for scalability, configure all managed instances in a single cluster defined within the WebLogic Administrative Console.

5. From the EnterpriseOne Server Manager Console, set the TimeWaitBeforeAutoResume setting to 0 to improve EnterpriseOne performance:

Web Runtime Settings > Web Runtime > Auto Resume Interval (Value: 0)

Note: After adding a large number of nodes to a cluster, it may be faster to restart the VM servers using the WebLogic Admin Console rather than through Server Manager.

6. From the EnterpriseOne Server Manager Console, consider changing these other HTML Server (JAS) and JDBJ setting changes:

- **Max Users**

Assuming 2,000 users, the recommended configured value for Exalogic is 500 per managed instance. The default value is 100.

Web Runtime > Web Runtime > Max Users

- **Maximum Connections**

Assuming 2,000 users, the recommended configured value for Exalogic is 500. The default value is 50.

JDBJ Database Configuration > JDBJ Connection Pools > Maximum Connections

- **Pool Growth Size**

The configured value for Exalogic assuming 2,000 users is 10.

JDBJ Database Configuration -> JDBJ Connection Pools

- **Initial Connections**

The configured value for Exalogic assuming 2,000 users is 10.

JDBJ Database Configuration -> JDBJ Connection Pools

- **Minimum Connections**

The configured value for Exalogic assuming 2,000 users is 50.

JDBJ Database Configuration -> JDBJ Connection Pools -> Minimum Connections

- **BSFN Timeout Threshold**

The configured value for Exalogic assuming 2,000 users is 270000 milliseconds. The default setting is 90000 (90 seconds).

Network Settings -> JDENET Configuration -> Timeout Threshold

- **Async BSFN Thread Pool Size**

The default configured value for this setting is 50. Increasing this value to 500 avoids any bottlenecks in thread pooling for the increased load anticipated with the in-memory applications.

Web Runtime -> Web Runtime -> Initial BSFN Thread Pool Size

You should also check **Asynchronous Business Function** checkbox to take advantage of this above setting. This is the default behavior and setting, but should be validated to ensure the relevance of this tuning value.

4.2 WebLogic Server Startup Parameters for Managed Instances

This section describes the Oracle WebLogic Server startup parameters for managed instances. An example of the startup parameters is shown below:

```
XX:MaxPermSize=256m -d64
-Djavax.xml.rpc.ServiceFactory=oracle.j2ee.ws.client.ServiceFactoryImpl -Xms8g
-Xmx8g -Xmn6g -XX:ParallelGCThreads=32 -XX:+AggressiveOpts -verbose:gc
-XX:+PrintGCDetails -XX:+PrintGCTimeStamps -DcloneId=VC1 -Djava.awt.headless=true
```

The list below further describes the functionality of each startup parameter and their impact on JD Edwards EnterpriseOne in-memory applications.

- **MaxPermSize**

If you are received error messages indicating "out of perm space", it is recommended that you change this value from 128 to 256.

- **-Xmn6g**

It is generally desirable for the YoungGeneration size larger than the OldGeneration to enable the JVM to recover more allocations during the minor GC. Consequently, with an 8GB heap size, it is preferable to have the YoungGeneration at 6 GB (-Xmn6g) which would result in a setting for the OldGeneration at 2 GB.

- **ParallelGCThreads=32**

You may have too many ParallelGC threads running as the default is to start the same number of ParallelGC threads as the number of CPUs. Too many threads can waste CPU cycles. A value not to exceed 32 is recommended.

- **AggressiveOpts**

You can accept the default value of -verbose:gc.

Tuning the Oracle Database Machine

This chapter discusses these topics:

- [Section 5.1, "Primary Indexes"](#)
- [Section 5.2, "Table Pinning in Flash Memory on the Oracle Database Machine"](#)
- [Section 5.3, "Configuring the Oracle Exadata Database Server"](#)

5.1 Primary Indexes

Specific in-memory applications for JD Edwards EnterpriseOne were evaluated to determine if placing critical indexes in memory on the Oracle database machine yielded performance benefits. This section documents specific indexes to consider manually placing in memory. However, other impacts to the overall Oracle Exadata database performance may be introduced and the overall performance of the database should be considered and evaluated. Additionally, the customer DBA must determine whether it is feasible to place the recommended indexes in memory given constraints such as the size of database tables and available memory.

This section discusses the recommendations for creating primary indexes in memory for the following JD Edwards EnterpriseOne In-Memory applications:

- [Section 5.1.1, "Project Portfolio Management"](#)
- [Section 5.1.2, "Sales Advisor"](#)

5.1.1 Project Portfolio Management

For the Project Profile Management application, performance can be improved by creating the primary indexes in memory for the following tables:

- F4812H
- F0911
- F4812
- F0902

When specifying indexes to be created in memory, the schema, table name, and the index name must be specified. For example, in a DV910 environment, the JD Edwards default schema is testdta. Therefore, assuming DV910 the following create index segment specifies these values in the correct order and syntax:

```
create index testdta.f4812h_exa
```

Caution: For supportability purposes, it is recommended that the index name specified in this document is used.

The second segment of the SQL statement to create the index in memory must include the schema, table name, and each table column name. For example, the following segment completes the create index in memory SQL statement:

```
on testdta.f4812h(wdmcu,wdoobj,wdsuub,wdlssq,wdvoid,wdelgc,wdidgj);
```

The following are the SQL statements that are recommended to create the primary indexes in memory (each create index should be specified as one contiguous line with no line returns) for the DV910 pathcode and testdta datasource:

```
create index testdta.f4812h_exa on
testdta.f4812h(wdmcu,wdoobj,wdsuub,wdlssq,wdvoid,wdelgc, wdidgj);
```

```
create index testdta.f0911_exa on
testdta.f0911(glmcu,glaid,glit,globj,glsub,gldgj,glpost,glaa,glu);
```

```
create index testdta.f4812_exa on
testdta.f4812(wdaid5,wdmcu,wdoobj,wdsuub,wddgj,wdelgc,wdcrcd,wdaid,wdccd,wdbtol,wtdcls,wdjrst,dicuj);
```

```
create index testdta.f0902_exa on testdta.f0902
("gbctry","gblt","gbfq","gbfy","gbmcu","gbaid");
```

5.1.2 Sales Advisor

Unlike the In-Memory application of Project Portfolio Management, the indexes for the Sales Advisor application are pre-built and included in the deliverable of the code base. Therefore no further index creation requirements are needed.

5.2 Table Pinning in Flash Memory on the Oracle Database Machine

The Oracle database machine allows tables to be pinned to fast access memory areas. This feature is unique in how it is managed and handled by the Oracle software. The Oracle database allows the database administrators to specify certain tables to be forced into cache apart from the automatic caching algorithm. By using this feature when faster memory can be used in lieu of storage, the end user response time in lab environments was decreased up to 33% for 2000 and 3000 user loads.

This chapter discusses these topics:

- [Section 5.2.1, "Discovering Suitable Candidates for Table Pinning"](#)
- [Section 5.2.2, "Manually Pinning Table into Flash Cache Memory"](#)

5.2.1 Discovering Suitable Candidates for Table Pinning

The method to determine which tables to pin into flash memory included querying the Oracle metric 'buffer gets' and 'full table scans'. A full table scan is one of the more costly disk I/O database processes. Table scans involve many requests for data stored on the disk and requires a substantial number of 'buffer gets'. The performance of table scans is significantly improved when the targeted table is in solid state flash memory.

The following Oracle SQL Query was used to determine high 'buffer gets' and 'full table scans'. As a result, for EnterpriseOne the serialized object tables, DV900.F989999

and DV900.F989998, were identified as good candidates for pinning into the solid state flash memory.

```
%> cat get FullScan.sql
set pages 1000
set feedback off
set lines 150
spool fts.lst
column sql_text Format A50 Heading 'Full Table Scans' wrap
break on sql_text skip 1

Select Executions, Sorts, Disk_Reads, Buffer_Gets, CPU_Time, Elapsed_Time, sql_
text from v$sqlarea
where (address, hash_value) in (Select address, hash_value from v$sql_plan where
options like '%FULL%' and operation like '%TABLE%') Order by Elapsed_Time

spool off
set feedback on
```

The resulting output file was inspected and the following line was identified:

```
EXECUTIONS SORTS   DISK READS BUFFER GETS CPU TIME ELAPSED TIME Full Table Scans

32284          0      2271          266744503   447974714          452738447

SQL_TEXT

SELECT  T0.WBJPO,T0.WBUID,T0.WBOID,T0.WBLNGPREF,T0.WBJVER FROM DV900.F989999 T0,
DV900.F989998 T1 WHERE ((T1.WBJOBID = :1 AND T0.WBUID = :2 )) AND (T0.WBOID =
T1.WBOID)
```

Relative to the results identified above, the large number of disk reads, executions and buffer gets indicated these tables are excellent candidates for caching.

5.2.2 Manually Pinning Table into Flash Cache Memory

To manually pin tables into flash memory the Oracle databases uses a simple SQL query. For example, to pin the two EnterpriseOne tables that were identified as likely candidates for improved performance in the previous section of this chapter. In this example, the cell_flash_cache is the name of the faster storage. The following example shows the SQL commands to pin specific JD Edwards EnterpriseOne tables in a RAM solid state disk cache on the Exadata Database Machine:

```
alter table dv910.f989999 storage (cell_flash_cache keep);
alter table dv910.f989998 storage (cell_flash_cache keep);
```

5.3 Configuring the Oracle Exadata Database Server

This section provides some recommended configuration parameters for JD Edwards EnterpriseOne Database Server. These settings are representative of the changes you should consider to configure your own system, depending on such factors as available machine resources and expected user load. These settings are based on a load of 2,000 users.

Note: These recommendations assume user competency with applicable tools on the referenced systems.

The example 'alter' commands listed in this chapter are representative of those that your Oracle DBA might use to configure the database on your Exadata machine. These commands assume a load of 2,000 users on a two-node database on the Exadata machine.

This section discusses these topics:

- [Section 5.3.1, "Example - Alter Commands"](#)
- [Section 5.3.2, "Example - Database Settings After 'alter' Commands"](#)
- [Section 5.3.3, "Oracle Shared and Dedicated Connections"](#)
- [Section 5.3.4, "Scaling Project Portfolio Management and Sales Advisor Using Oracle Database Tuning Settings"](#)
- [Section 5.3.5, "Table and Index Partitioning for the Project Portfolio Management In-Memory Application"](#)

5.3.1 Example - Alter Commands

The following are an example of the 'alter' commands that an Oracle DBA might consider to configure the Oracle Database:

```
alter system set "_enable_NUMA_support"=FALSE scope=spfile;

alter system set "_file_size_increase_increment"=2044M scope=spfile;

alter system set compatible=11.2.0.3.0 scope=spfile;

alter system set db_lost_write_protect=TYPICAL scope=spfile;

alter system set filesystemio_options=setall scope=spfile;

alter system set global_names=TRUE scope=spfile;

alter system set open_cursors=1000 scope=spfile;

alter system set parallel_adaptive_multi_user=FALSE scope=spfile;

alter system set processes=10000 scope=spfile;

alter system set sql92_security=TRUE scope=spfile;

alter system set dispatchers = '' scope=spfile;

alter system set pga_aggregate_target=0 scope=spfile;

alter system set shared_servers=0 scope=spfile;
```

5.3.2 Example - Database Settings After 'alter' Commands

The following shows an example of the database settings after the alter commands in the preceding section have been executed to configure the Oracle database used with Oracle VM Templates on Exalogic. This example assumes the Exadata Oracle IDs are ovsorc1 and ovsorc2 for each of the two nodes on the Exadata Database Machine.

An ASM Disk Group of DA_SLCM61 is used in this example. The Exadata target hostname is 'slcm61-scan1' on the default port of 1521 for Oracle connection configuration. The Exadata is configured to use dedicated connections and automated memory management.


```

ovsorcl2.__db_cache_size=25232932864
ovsorcl1.__db_cache_size=20937965568
ovsorcl2.__java_pool_size=134217728
ovsorcl1.__java_pool_size=134217728
ovsorcl2.__large_pool_size=13421772 8
ovsorcl1.__large_pool_size=13421772 8
ovsorcl2.__oracle_base='/u01/app/oracle'
ovsorcl2.__pga_aggregate_target=21743271936
ovsorcl1.__pga_aggregate_target=26172456960
ovsorcl2.__sga_target=32480690176
ovsorcl1.__sga_target=28051505152
ovsorcl2.__shared_io_pool_size=0
ovsorcl1.__shared_io_pool_size=0
ovsorcl2.__shared_pool_size=6710886400
ovsorcl1.__shared_pool_size=6308233216
ovsorcl2.__streams_pool_size=0
ovsorcl1.__streams_pool_size=268435456
*.enable_NUMA_support=FALSE
*.file_size_increase_increment=2143289344
*.audit_file_dest='/u01/app/oracle/admin/ovsorcl/adump'
*.audit_trail='db'
*.cluster_database=true
*.cluster_interconnects=''
*.compatible='11.2.0.2.0'
*.control_files='+DA_SLCM61/ovsorcl/controlfile/current.286.807801199','+DBFS_
DG/ovsorcl/controlfile/current.2      56.807801199'
*.db_block_size=8192
*.db_create_file_dest='+DA_SLCM61'
*.db_domain=''
*.db_lost_write_protect='TYPICAL'
*.db_name='ovsorcl'
*.db_recovery_file_dest='+DBFS_DG'
*.db_recovery_file_dest_size=4558159872
*.diagnostic_dest='/u01/app/oracle'
*.dispatchers=' '
* . filesystemio_options='SETALL'
*. global_names=TRUE
ovsorcl2.instance_number=2
ovsorcl1.instance_number=1
*.memory_target=54129590272
*.open_cursors=1000
*.parallel_adaptive_multi_user=FALSE
*.pga_aggregate_target=0
*.processes=10000
*.remote_listener='slcm61-scan1:1521'
*.remote_login_passwordfile='exclusive'
*.sec_case_sensitive_logon=FALSE
*.sessions=2755
*.shared_servers=0
*.sql92_security=TRUE ovsorcl2.thread=2 ovsorcl1.thread=1
ovsorcl1.undo_tablespace='UNDOTBS1' ovsorcl2.undo_tablespace='UNDOTBS2' *.use_
large_pages='TRUE'

```

5.3.3 Oracle Shared and Dedicated Connections

The Oracle database can be configured in either a shared or dedicated connection mode, where:

- Dedicated connection mode is the default for the Oracle database and works best on systems that have ample memory and require persistent connections.

- Shared connection mode is ideal when processor and memory resources need to be constrained.

For the JD Edwards EnterpriseOne in-memory applications, the number of connections to the database is greatly increased relative to applications that are not in-memory. In some instances the number of connections could be as high as 10 times greater. As a result of this increase in connections, the Oracle database requires a larger footprint in processor and memory resources. Therefore, while shared connections can be used to place a limit on processor and memory consumption, it is likely that this connection mode will result in decreased performance.

5.3.4 Scaling Project Portfolio Management and Sales Advisor Using Oracle Database Tuning Settings

There are a number of direct Oracle database tuning settings at the startup of the database that are not dynamic and which must be considered with the JD Edwards EnterpriseOne in-memory applications. As described below, these settings are the memory sizing of the PGA/SGA, processes, open cursors and the configuration of the redo and temp areas of the database.

- **PGA/SGA and Processes**

Enabling the feature of Automatic Memory Management (AMM) requires the SGA and PGA together. It should be configured initially depending on the options of the Oracle installation and the amount of memory on the database server. Additionally, shared connections will add to the requirements necessary for the SGA, so following the recommendations in this section for Oracle database tuning in a shared connection environment in this regard is important.

- **Open_Cursors**

The following script can be initiated to determine the amount of open cursor consumption by the JD Edwards EnterpriseOne in-memory applications:

```
-- Script to calculate open_cursor utilization
column highest_open_cur format 999;
column max_open_cur format a15;
select max(a.value) as highest_open_cur, p.value as max_open_cur
from v$sesstat a, v$statname b, v$parameter p where a.statistic# = b.statistic#
and b.name = 'opened cursors current' and p.name= 'open_cursors' group by
p.value;
```

In lab environments, relative to the normal JD Edwards EnterpriseOne applications, by design the in-memory applications consumed a much larger number of open cursors. It would not be uncommon for this number to be in the range of 10 to 20K.

- **Oracle Redo Logs/Temp**

The JD Edwards EnterpriseOne in-memory applications demonstrate extensive use of insert, update, and delete activity since it relies heavily on work tables in the JD Edwards EnterpriseOne schema. As a result these large numbers of data manipulation language (DML) actions require larger and greater numbers of Redo logs. The redo logs store changes to the database for instance recovery and integrity of the database in case of failure.

The default number and size of the redo logs is small. Coupled with applications that require a greater size for these logs, the use of the defaults will likely cause the database server to checkpoint continuously. This puts a high load on the buffer cache and disk I/O subsystem. If there are too few redo logs, then the archive

cannot keep up, and the database will must wait, degrading the overall performance of the JD Edwards EnterpriseOne application.

Running the EnterpriseOne in-memory applications may require a larger Redo logs. Also consider that the JD Edwards EnterpriseOne in-memory applications require larger temporary tablespace sizes to help improve sort performance.

5.3.5 Table and Index Partitioning for the Project Portfolio Management In-Memory Application

Table and index partitioning was evaluated in lab environment for the Project Portfolio Management work tables as a way to increase the performance. The range partitioning of both the table and index was performed for the Project Portfolio Management work tables.

Range partitioning is useful when you have distinct ranges of data you want to store together, which is the case with the Project Portfolio Management application. Although implementing this practice would appear to be of great benefit, in practice the additional cost of this feature likely does not warrant the small benefit observed when table and index partitioning was enabled.

Managing Queries

This appendix contains the following topics:

- [Appendix A.1, "Selecting Additional Fields"](#)
- [Appendix A.2, "Clearing the Form"](#)

You use Query Management mode to create, modify, and save queries. Click the icon to the right of the Query field to enter Query Management mode. Depending on the value in the Query field when you enter Query Management, the options for a new query or an existing query appear in the Query Management side panel. When you enter Query Management mode, a plus sign on the form designates each field and QBE column that you can select for a query.

See Also

The chapter entitled: [Creating Queries to Search for Data](#) in the *JD Edwards EnterpriseOne Tools Foundation Guide*.

http://docs.oracle.com/cd/E24705_01/doc.91/e24251/jde_e1_web_user_interface.htm#EOTF0345

A.1 Selecting Additional Fields

You can select a comparison type in the Query Management panel for each field or query-by-example (QBE) column and specify the comparison value in the query. To do so, click the form fields or QBE columns that you want use as search criteria. You can select only the form fields and QBE columns that have a plus sign.

Alternatively, you can use the Additional Fields Selection section to select the form fields or QBE columns. This section displays only the business view (BSVW) columns that are included on the form including the hidden columns. For example, if BSVW includes the columns A, B and C; A is displayed on the form, B is hidden and C is not included, then the Additional Fields Selection section displays A and B columns.

See Also

The chapter entitled: [Creating Queries to Search for Data](#) in the *JD Edwards EnterpriseOne Tools Foundation Guide*.

http://docs.oracle.com/cd/E24705_01/doc.91/e24251/jde_e1_web_user_interface.htm#EOTF0345

A.2 Clearing the Form

You can clear the values that you have entered on the form. For example, the values entered in the header or QBE column.

To clear the values on the form:

1. Access a form with the Query control.
2. Enter any search criteria on the header or the QBE and then, click Find.
3. Click the Clear Form icon located to the right of the Add, Edit Queries icon. The system deletes the values in the header or the QBE.

Alternatively, you can select Clear form when selected option in the Query Management panel to clear the values on the form automatically when the query is selected.