

Oracle FLEXCUBE Websphere Server 8.5 Best Practices
Oracle FLEXCUBE Universal Banking
Release 12.87.06.0.0
[May] [2020]



Table of Contents

1. INTRODUCTION	1-1
1.1 BACKGROUND	1-1
1.2 BASICS OF WEBSPHERE	1-1
1.3 PROFILE	1-1
1.4 REFERENCES	1-3
2. JDBC TUNING	2-1
3. JVM TUNING	3-1
3.1 JVM HEAP SIZE	3-1
3.2 THREAD POOL SIZE	3-2
4. LOGGING	4-1
4.1 DIAGNOSTIC TRACE	4-1
4.2 JVM LOGS	4-2
4.3 PROCESS LOGS	4-3
4.4 IBM SERVICE LOGS	4-3
4.5 CHANGE LOG LEVEL DETAILS	4-4
4.6 NCSA ACCESS AND HTTP ERROR LOGGING	4-5
5. SESSION MANAGEMENT	5-1
6. APPENDIX A: FREQUENTLY ENCOUNTERED ERRORS	6-1

1. Introduction

1.1 Background

IBM® WebSphere® Application Server 8.5 supports a range of applications, each with their own unique set of features, requirements, and services. Just as no two applications will use an application server in exactly the same way, no single set of tuning parameters will likely provide the best performance for any two different applications.

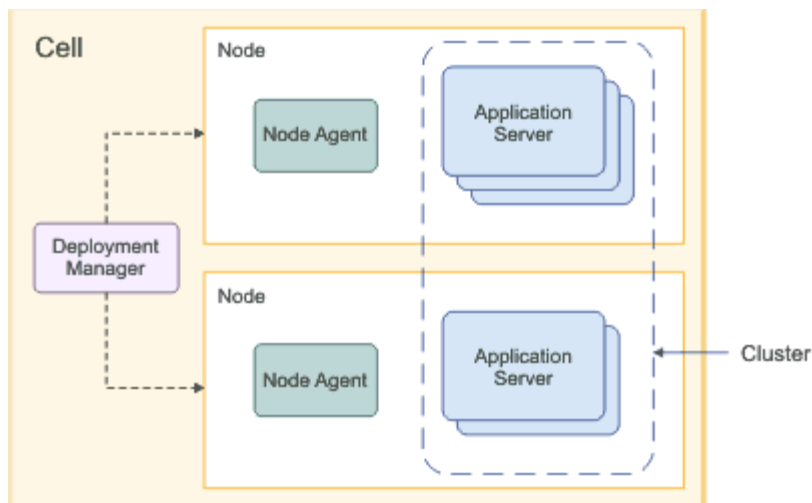
This document explains the best practices followed for Web Sphere application server tuning in the perspective of FLEXCUBE UBS.

1.2 Basics of Websphere

IBM websphere application server cluster deployment contains the below key elements

- Cell
- Nodes
 - Deployment Manager Node- “DMGR”
 - Node- “NodeXX”
 - Node Agent- “NAXX”
- Profiles
- Cluster
- Cluster Members
- Data Sources

1.3 Profile



Following are the details about profile:

Cell

A cell is a grouping of nodes into a single administrative domain. In a Network Deployment environment, a cell can consist of multiple nodes (and node groups), which are all administered from a single point, the deployment manager.

Node

A node is an administrative grouping of application servers for configuration and operational management within one operating system instance

Node Agent

In distributed server configurations, each node has a node agent that works with the deployment manager to manage administration processes. A node agent is created automatically when you add (federate) a stand-alone node to a cell.

Cluster:

A cluster is a logical collection of application server processes that provides workload balancing and high availability. Application servers that belong to a cluster are members of that cluster and must all have identical application components deployed on them.

A profile is a Websphere runtime environment formed by collection of User data and Product files. Product Files are shared application binaries for Websphere. User data is set of user customizations for a specific runtime environment.

Prominent profile types are:

- Stand-alone Application Server
- An application server environment runs Enterprise Application. Application server is managed from its own administrative console and functions independently from other application server.
- Deployment Manager:

A Deployment Manager manages operations for a logical group or cell of other servers. It is the central administration point of a cell that consists of multiple nodes and node groups in a distributed server configuration. The deployment manager uses the node agent to manage the application servers within one node. A deployment manager provides management capability for multiple federated nodes and can manage nodes that span multiple systems and platforms. A node can only be managed by a single deployment manager and must be federated to the cell of that deployment manager.



Deployment Manager is part of Network Deployment Edition of Websphere.

1.4 **References**

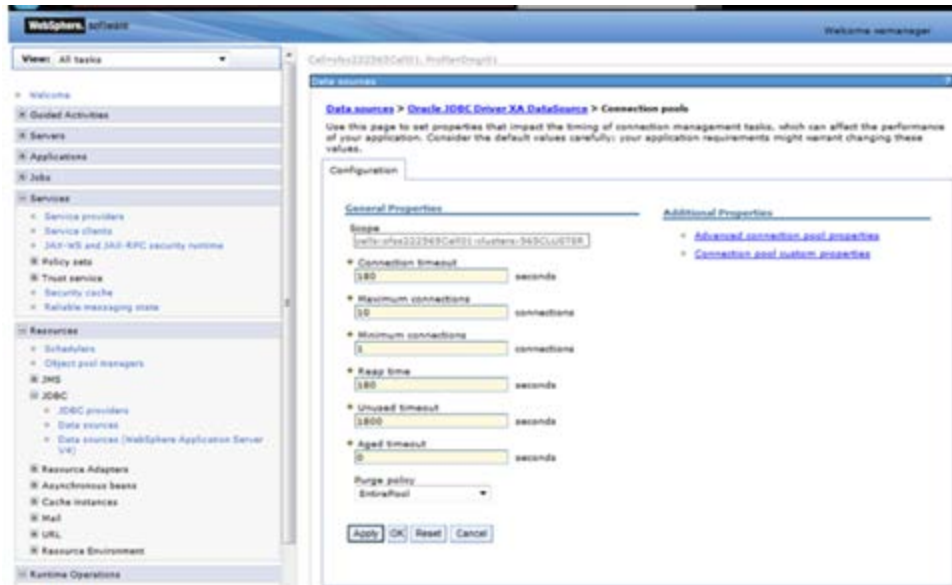
Before proceeding with the document, ensure the below documents are followed for setup

- For resource creation(JDBC Provider, Datasource, Queue connection factory, JMS queue), refer to Resources_To_Be_Created.docx
- For cluster configuration, refer to FCUBS_WAS_Cluster_Configuration.docx
- For application deployment, refer to FCUBS_Manual_Deployment_WAS.docx
- For deployment of Gateway applications, refer to Gateway_Application_Deployment_Websphere.docx
- For SSL configuration in Websphere, refer to SSL_Configuration_WAS.doc

2. JDBC Tuning

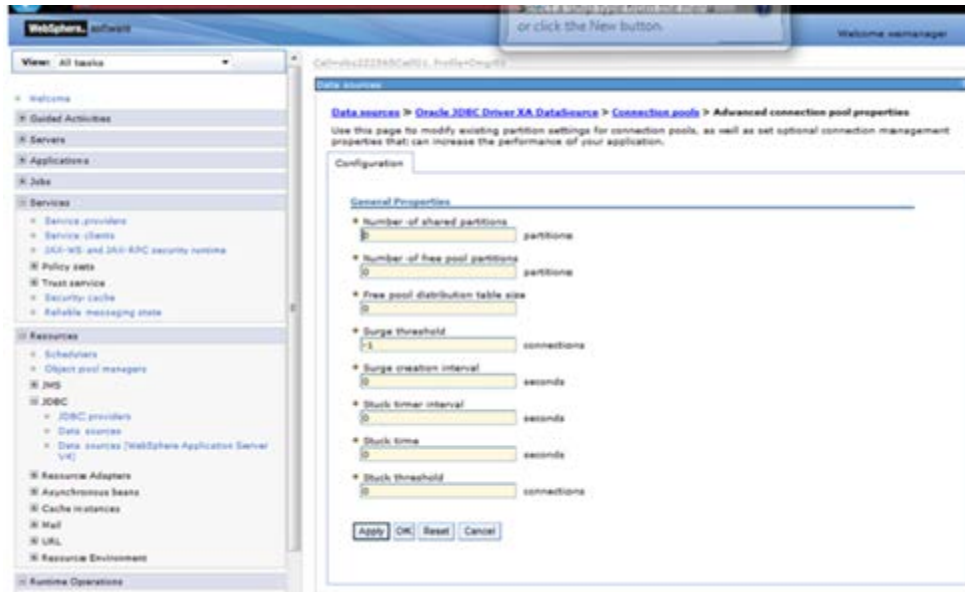
The JDBC provider object encapsulates the specific JDBC driver implementation class for access to the specific vendor database of your environment. The datasource object supplies your application with connections for accessing the database. In a typical production environment database and application needs to use the perfect combination of parameters to achieve higher throughput.

Navigation Path >> Resources->JDBC->Data Sources ->DATA_SOURCE_NAME->Connection Pools ->



Parameter	Value
Connection Time out	180
Maximum Connection	10
Minimum Connections	1
Reap Time	180
Unused Time	180
Aged Timeout	0
Purge Policy	Entire Pool

Navigation Path >> Resources-> JDBC->Data Sources ->DATA_SOURCE_NAME->Connection Pools ->Advanced Connection Pool Properties



Parameter	Value
Number of shared pool partitions	0
Number of free pool partitions	0
Free pool distribution table size	0
Surge threshold	-1
Surge creation interval	0
Stuck timer interval	0
Stuck time interval	0
Stuck threshold	0

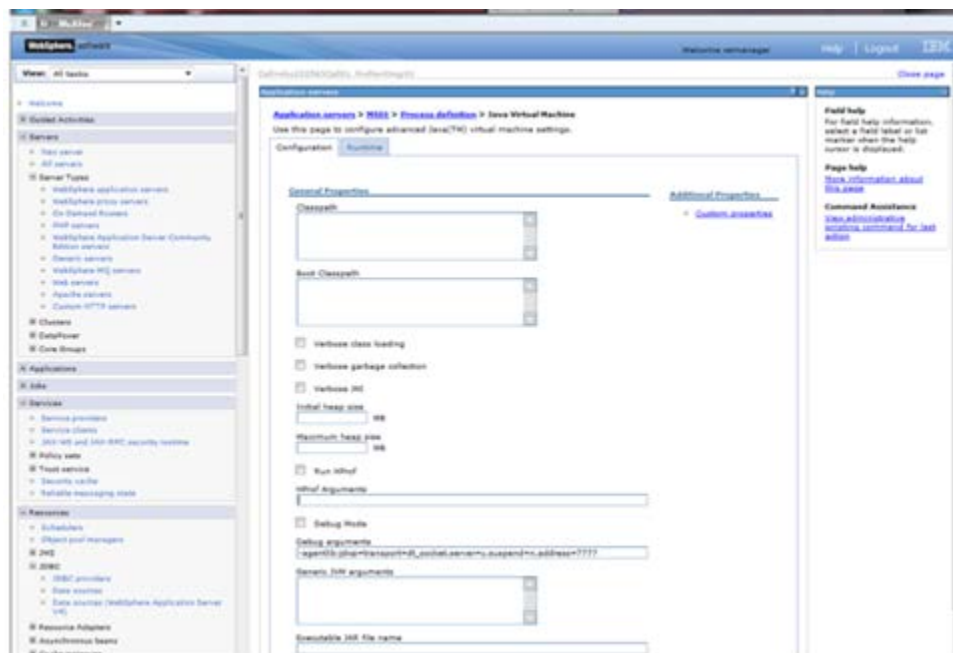
3. JVM Tuning

3.1 JVM Heap Size

When a Java Virtual Machine (JVM) is started, it obtains a large area of memory from the underlying operating system. This area is called the heap, and Java performs its own memory management by allocating areas of the heap as memory is needed by the process. Performance tuning generally starts with the Java Virtual Machine (JVM), which serves as the foundation for the application server. From that point forward, tuning is primarily driven by the application server components that are used by the application.

Every 75 concurrent FLEXCUBE users require one websphere application server of size 4GB. I.e. for 300 concurrent FLEXCUBE users, it is recommended to have 4 websphere application servers.

Navigation Path >> Application Server->APPLICATION_SERVER_NAME-> Server Infrastructure->Java And Process Management->Process Definition-> Java Virtual machine



Parameter	Value
Initial Heap Size	1024
Maximum Heap Size	4096

3.2 Thread Pool Size

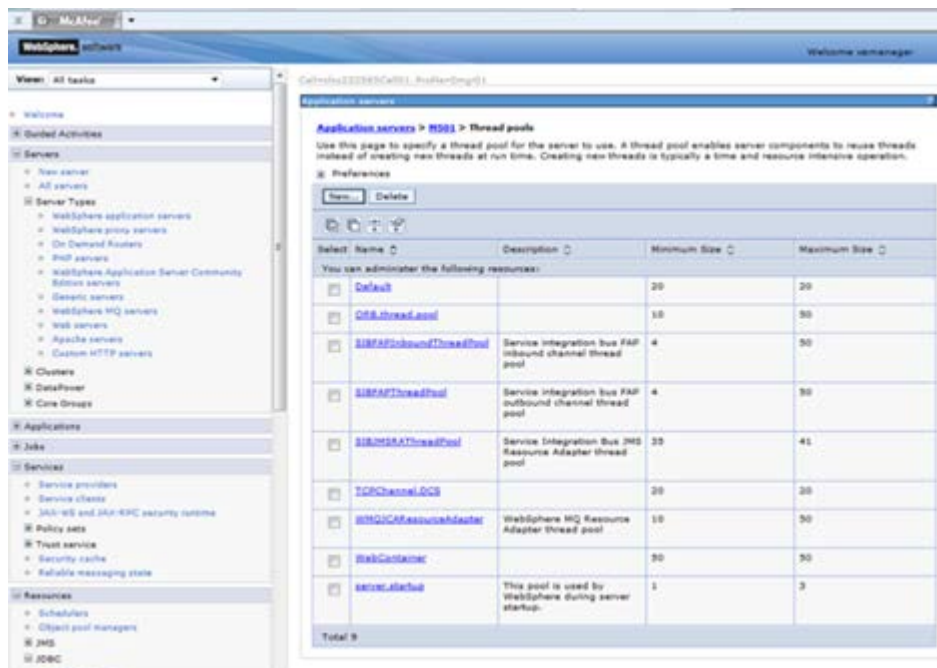
Each task performed by the server runs on a thread obtained from one of WebSphere Application Server's many thread pools. A thread pool enables components of the server to reuse threads, eliminating the need to create new threads at run time to service each new request. Three of the most commonly used (and tuned) thread pools within the application server are:

Web container: Used when requests come in over HTTP.

Default: Used when requests come in for a message driven bean or if a particular transport chain has not been defined to a specific thread pool.

ORB: Used when remote requests come in over RMI/IIOP for an enterprise bean from an EJB application client, remote EJB interface, or another application server.

Navigation Path >> Application Server->APPLICATION_SERVER_NAME->Additional Properties->Thread Pools->



Parameter	Minimum Size	Maximum Size	Thread Inactive Timeout
Default	20	20	5000
ORB.thread.pool	10	50	3500
Web Container	50	50	60000

Middleware servers

Middleware servers > MS01 > Thread pools > Default

Use this page to specify a thread pool for the server to use. A thread pool enables server components to reuse threads instead of creating new threads at run time. Creating new threads is typically a time and resource intensive operation.

Configuration

General Properties

* Name
Default

Description

* Minimum Size
20 threads

* Maximum Size
20 threads

* Thread inactivity timeout
5000 milliseconds

☐ Allow thread allocation beyond maximum thread size

Apply OK Reset Cancel

Additional Properties

☐ Custom properties

4. Logging

When working with IBM to debug request processing problems, there might be occasions where low-level tracing components must be enabled to capture details for how the request is processed. These low-level Application server trace components do not have knowledge of the request intent or the potential data within. Therefore, when enabled, it is possible that these tracing components might potentially include sensitive information, in plain text, in the trace file.

It is recommended whenever possible to not enable these types of tracing components on a production system and attempt to simulate the problem on a quality assurance environment to capture the appropriate information.

Navigation Path >> Application Server->APPLICATION_SERVER_NAME->Process Definition-> Logging And Tracing

4.1 Diagnostic Trace

The screenshot shows the 'Diagnostic trace service' configuration page in the IBM WebSphere Administrative Console. The breadcrumb trail is 'Middleware servers > MS01 > Process definition > MS01 > Diagnostic trace service'. Below the breadcrumb, there is a description: 'Use this page to view and modify the properties of the diagnostic trace service. Diagnostic trace provides detailed information about how the application server components run within this managed process. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.' There are two tabs: 'Configuration' (selected) and 'Runtime'. The 'General Properties' section is expanded, showing 'Trace Output' options: 'None', 'Memory Buffer', and 'File' (selected). Under 'File', there are fields for 'Maximum Buffer Size' (8 thousand entries), 'Maximum File Size' (20 MB), 'Maximum Number of Historical Files' (5), and 'File Name' (\$SERVER_LOG_ROOT/trace.log). There is also a 'Trace Output Format' dropdown set to 'Basic (Compatible)'. At the bottom are 'Apply', 'OK', 'Reset', and 'Cancel' buttons. An 'Additional Properties' section on the right has a link 'Change log detail levels'.

Parameter	Value
Trace Output	File

4.2 JVM Logs

Middleware servers

[Middleware servers](#) > [MS01](#) > [Process definition](#) > [MS01](#) > [JVM Logs](#)

Use this page to view and modify the settings for the Java virtual machine (JVM) System.out and System.err logs for a managed process. The JVM logs are created by redirecting the System.out and System.err streams of the JVM to independent log files. The System.out log is used to monitor the health of the running application server. The System.err log contains exception stack trace information that is used to perform problem analysis. One set of JVM logs exists for each application server and all of its applications. JVM logs are also created for the deployment manager and each node manager. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.

Configuration **Runtime**

General Properties

System.out

* File Name:

File Formatting

Log File Rotation

☒ File Size ☐ Time

Maximum Size MB

Start Time

Repeat Time hours

Maximum Number of Historical Log Files. Number in range 1 through 200.

Installed Application Output

☒ Show application print statements

☒ Format print statements

System.err

* File Name:

Log File Rotation

☒ File Size ☐ Time

Maximum Size MB

Start Time

Repeat Time hours

Parameter	Value	Remarks
System. out	{SERVER_LOG_ROOT}/SystemOut.log	File Name and File Size
System.err	{SERVER_LOG_ROOT}/SystemErr.log	File Name and File Size

4.3 Process Logs

Middleware servers > MS01 > Process definition > MS01 > Process Logs

Use this page to view or modify settings to specify the files to which standard out and standard error streams write. The process logs are created by redirecting the standard out and standard error streams of a process to independent log files. Native code writes to the process logs. These logs can also contain information that relates to problems in native code or diagnostic information written by the JVM. One set of process logs is created for each application server and all of its applications. Process logs are also created for the deployment manager and each node manager. Changes on the Configuration panel apply when the server is restarted. Changes on the Runtime panel apply immediately.

Configuration Runtime

General Properties

* Stdout File Name
\${SERVER_LOG_ROOT}/native_stdout.log

* Stderr File Name
\${SERVER_LOG_ROOT}/native_stderr.log

Apply OK Reset Cancel

Parameter	Value
Stdout File Name	\${SERVER_LOG_ROOT}/native_stdout.log
Stderr file name	\${SERVER_LOG_ROOT}/native_stderr.log

4.4 IBM Service Logs

Middleware servers > MS01 > Process definition > MS01 > IBM Service Logs

Use this page to configure the IBM service log, also known as the activity log. The IBM service log contains both the application server messages that are written to the System.out stream and special messages that contain extended service information that you can use to analyze problems. One service log exists for all Java virtual machines (JVMs) on a node, including all application servers and their node agent, if present. A separate activity log is created for a deployment manager in its own logs directory. The IBM Service log is maintained in a binary format. Use the Log Analyzer or Showlog tool to view the IBM service log.

Configuration Runtime

General Properties

☐ Enable service log

* File Name:
\${LOG_ROOT}/activity.log

* Maximum File Size
2 MB

☒ Enable Correlation ID

Apply OK Reset Cancel

Parameter	Value
Enable Service Log	False
Maximum File Size	2
File Name	\${LOG_ROOT}/activity.log
Enable Correlation Id	True

4.5 Change Log Level Details

Middleware servers

Middleware servers > MS01 > Process definition > MS01 > Change log detail levels

Use log levels to control which events are processed by Java logging. Click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative; a level near the top of the list includes all the subsequent levels.

Configuration Runtime

General Properties

Change log detail levels

☐ Disable logging and tracing of potentially sensitive data (WARNING: This might cause the log detail level setting to be modified when it is applied on the server.)

Select components and specify a log detail level. Log detail levels specified here will apply to the entire server. Expand Components and Groups and click Components to specify a log detail level for individual components, or click Groups to specify a log detail level for a predefined group of components. Click a component or group name to select a log detail level. Log detail levels are cumulative.

*=info

☐ Components and Groups

Correlation

Enable log and trace correlation so entries that are serviced by more than one thread, process, or server will be identified as belonging to the same unit of work.

☐ Enable log and trace correlation

- ☒ Include request IDs in log and trace records
- ☐ Include request IDs in log and trace records and create correlation log records
- ☐ Include request IDs in log and trace records, create correlation log records, and capture data snapshots

Parameter	Value
Disable logging	False
Enable Log and Trace Correlation	False

4.6 NCSA Access and HTTP Error Logging

Middleware servers

[Middleware servers](#) > [MS01](#) > [Process definition](#) > [MS01](#) > [NCSA access and HTTP error logging](#)

Use this page to configure HTTP error logs and National Center for Supercomputing Applications (NCSA) access logs.

Configuration

General Properties

☐ Enable logging service at server start-up

NCSA Access logging

☒ Enable access logging

* Access log file path

Access log maximum size
 MB

Maximum number of historical files

* NCSA access log format

Error logging

☒ Enable error logging

* Error log file path

Error log maximum size
 MB

Maximum number of historical files

* Error logging level

Parameter	Value
NCSA Access logging Enable	True
Error logging Enable	True

5. Session Management

Web browsers and applications use HTTP to communicate. Since HTTP is a stateless protocol (meaning that each command is executed independently without any knowledge of the commands that came before it), there must be a way to manage sessions between the browser side and the server side. Session management is used to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support.

Navigation Path >> Enterprise Application->APPLICATION_NAME->Web Module Properties->Session Management

All Applications > FCJWSApp > Session management

Use this page to configure session manager properties to control the behavior of Hypertext Transfer Protocol (HTTP) session support. These settings apply to both the SIP container and the web container.

Configuration

General Properties

☒ Override session management

Session tracking mechanism:

☐ Enable SSL ID tracking

☒ [Enable cookies](#)

☐ Enable URL rewriting

☐ Enable protocol switch rewriting

Maximum in-memory session count:
1000 sessions

☒ Allow overflow

Session timeout:

☐ No timeout

☒ Set timeout
30 minutes

☐ Security integration

Serialize session access:

☐ Allow serial access
Maximum wait time
0 seconds

☒ Allow access on timeout

[Custom properties](#)
[Distributed environment settings](#)

Apply OK Reset Cancel

Parameter	Value
Enable SSL ID Tracking	False
Enable Cookies	True
Enable URL rewriting	False
Maximum in-memory session count	1000
Session timeout	30 Minutes
Security Integration	False

Parameter	Value
Serialization Session access	True

6. Appendix A: Frequently Encountered Errors

Error 1: Websphere is causing too many database locks.

Solution: In order to avoid problems with database locks issue, change the custom property "webSphereDefaultIsolationLevel=2". By default Webpsphere uses Repeatable Read isolation level(4) for transactions. Before proceeding with the isolation level change make sure these locks are not caused by the application.

Possible Values	JDBC Isolation Level	Isolation Level
8	TRANSACTION_SERIALIZABLE	Repeatable Read (RR)
4 (default)	TRANSACTION_REPEATABLE_READ	Read Stability (RS)
2	TRANSACTION_READ_COMMITTED	Cursor Stability (CS)
1	TRANSACTION_READ_UNCOMMITTED	Uncommitted Read (UR)
0	TRANSACTION_NONE	No Commit (NC)

Navigation Path >> Resources-> JDBC->Data Sources ->DATA_SOURCE_NAME-> Custom Properties

If this property is found then update the value to 4 and Save, else create a new property with below mentioned steps:

1. Click New.
2. Enter webSphereDefaultIsolationLevel for the name field.
3. Enter 2 for the value field.

Data sources > Default DataSource > Custom properties > New...

Use this page to specify custom properties that your enterprise information system (EIS) requires for the resource providers and resource factories that you configure. For example, most database vendors require additional custom properties for data sources that access the database.

Configuration

General Properties

Scope
cells:ofss222565Node01Cell:nodes:ofss222565Node01:servers:server1

Name
webSphereDefaultIsolationLevel

Value
2

Description

Type
java.lang.String

Apply OK Reset Cancel



Websphere Server 8.5 Best Practices
[May] [2020]
Version 12.87.06.0.0

Oracle Financial Services Software Limited
Oracle Park
Off Western Express Highway
Goregaon (East)
Mumbai, Maharashtra 400 063
India

Worldwide Inquiries:
Phone: +91 22 6718 3000
Fax: +91 22 6718 3001
<https://www.oracle.com/industries/financial-services/index.html>

Copyright © [2007], [2020], Oracle and/or its affiliates. All rights reserved.

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

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 failsafe, 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.

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.

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.