Oracle Banking Websphere Server 8.5 Best Practices
Oracle Banking Enterprise Limits and Collateral
Management
Release 14.5.0.0.0
[May] [2021]

ORACLE
FINANCIAL SERVICES

Table of Contents

ORACLE

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
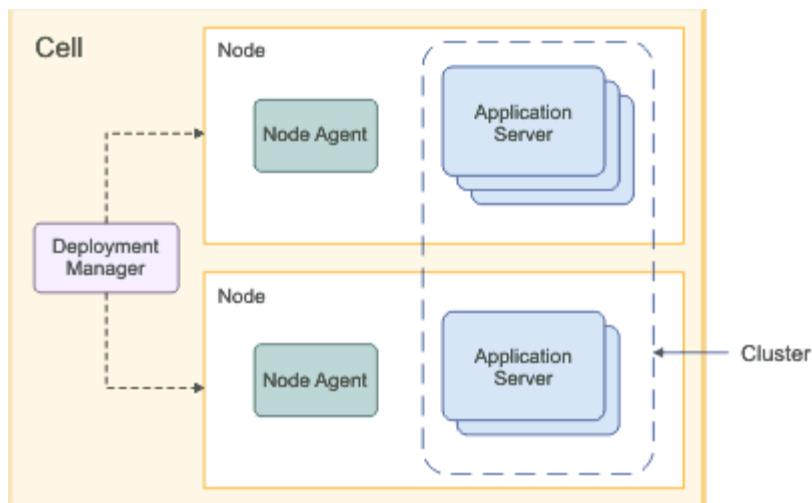
Following are the details about profile:

ORACLE

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.

ORACLE®

1.4 <u>References</u>

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

ORACLE

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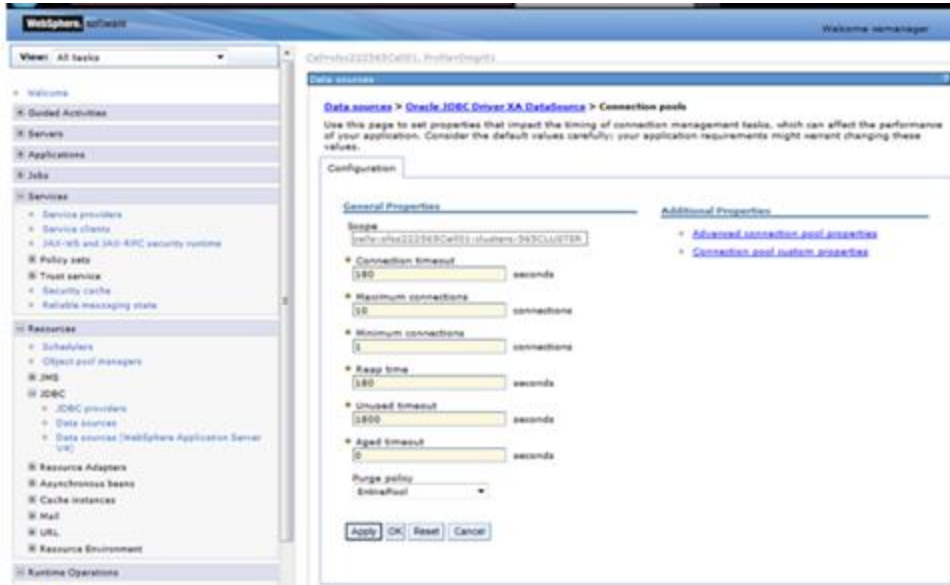
Parameter	Value
Connection Time out	180
Maximum Connection	10
Minimum Connections	1
Reap Time	180
Unused Time	180
Aged Timeout	0
Purge Policy	Entire Pool

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

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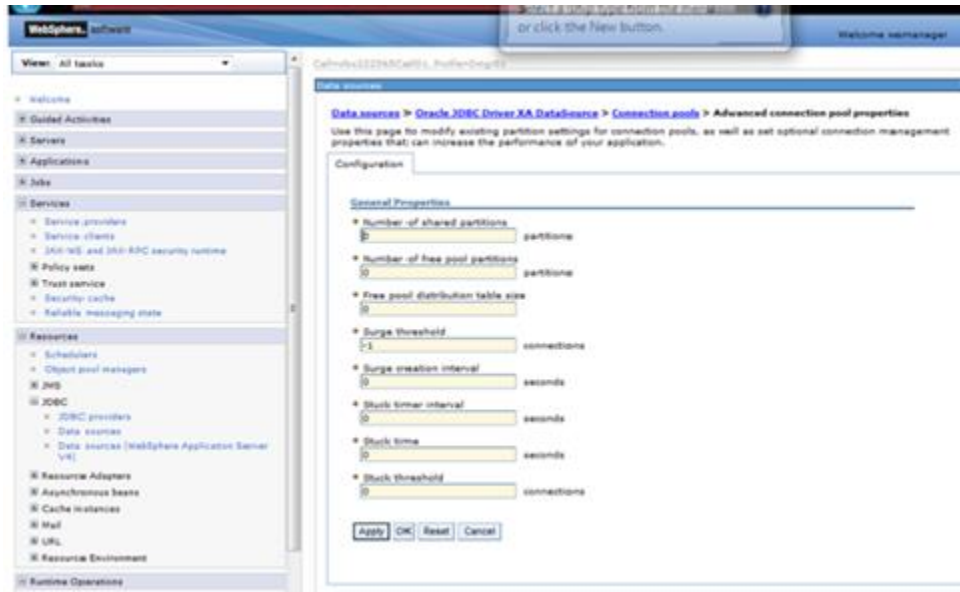
Parameter	Value
Number of shared pool partitions	0
Number of free pool partitions	0
Free pool distribution table size	0
Surge threshold	-1
Surge creation interval	0
Stuck timer interval	0
Stuck time interval	0
Stuck threshold	0

ORACLE

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
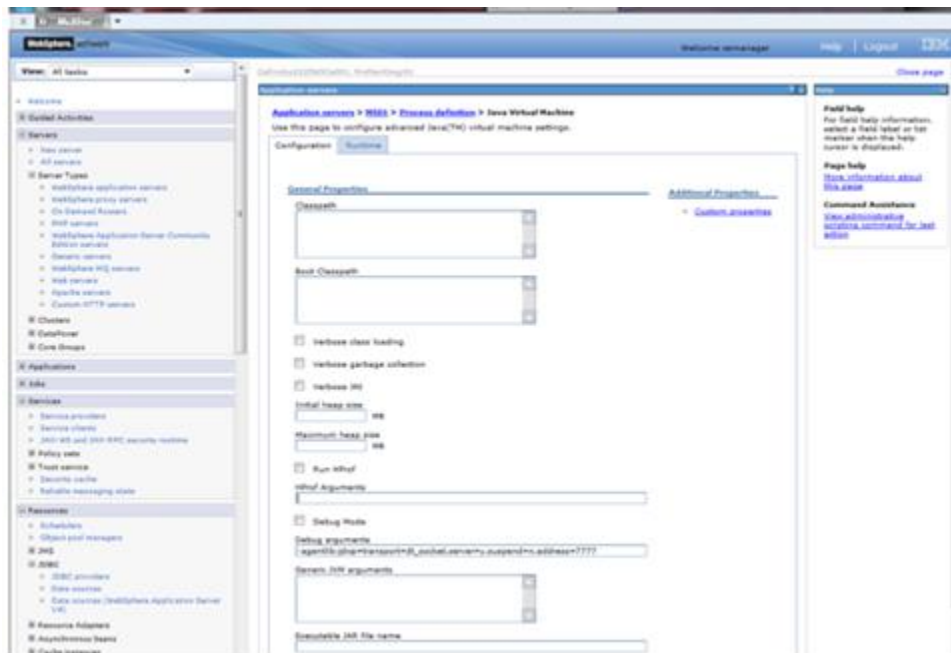
Parameter	Value
Initial Heap Size	1024
Maximum Heap Size	4096

ORACLE

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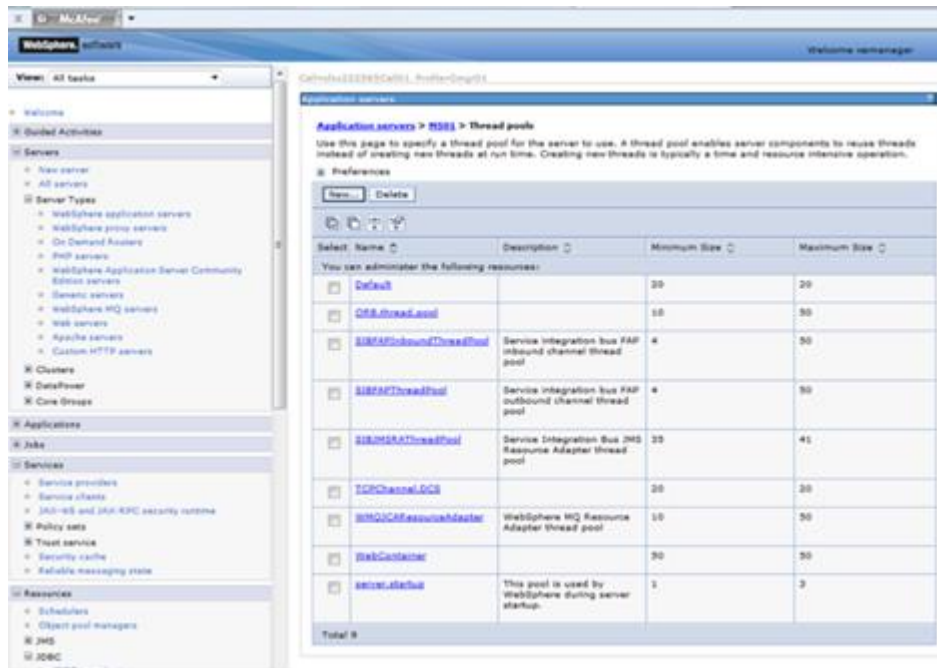
Parameter	Minimum Size	Maximum Size	Thread Inactive Timeout
Default	20	20	5000
ORB.thread.pool	10	50	3500
Web Container	50	50	60000

ORACLE

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

Additional 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

▪ Custom properties

3-3

ORACLE®

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

Parameter	Value
Trace Output	File

ORACLE

4.2   JVM Logs

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

ORACLE®

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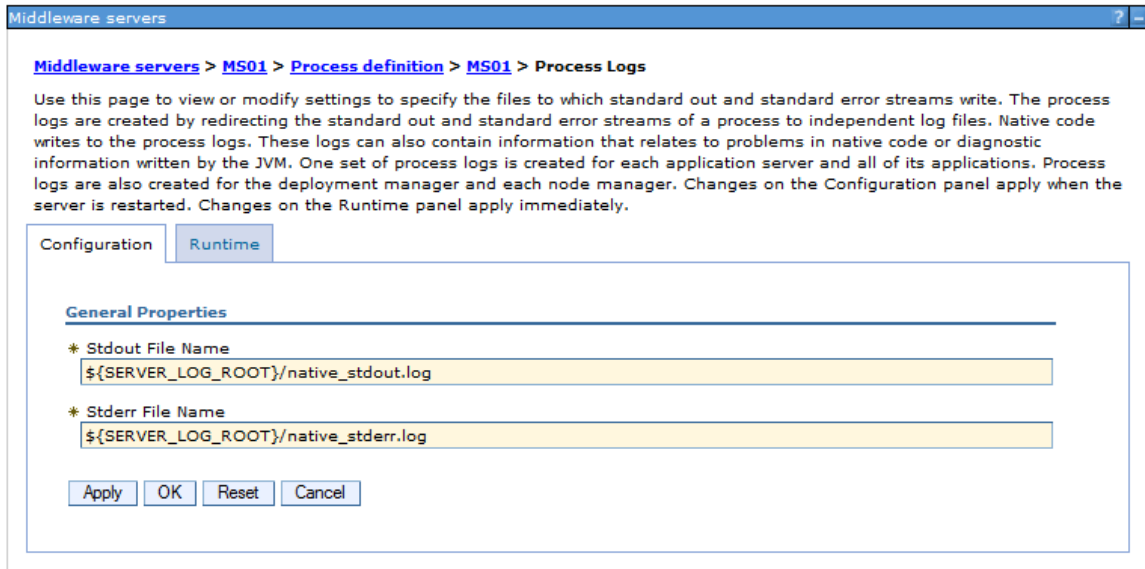
Parameter	Value
Stdout File Name	${SERVER_LOG_ROOT}/native_stdout.log
Stderr file name	${SERVER_LOG_ROOT}/native_stderr.log

4.4   IBM Service Logs

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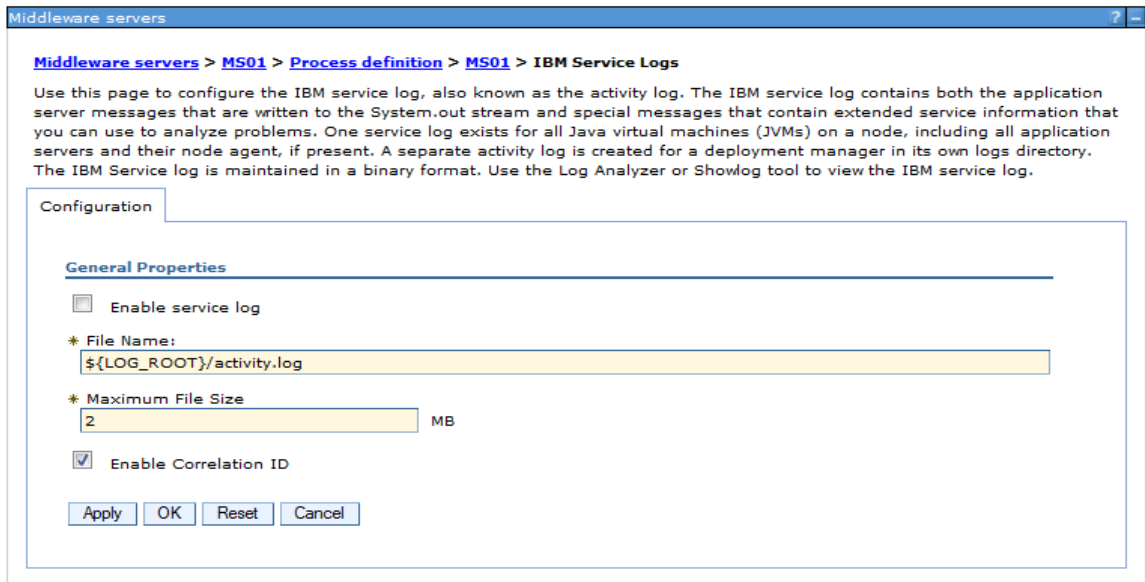
Parameter	Value
Enable Service Log	False
Maximum File Size	2
File Name	${LOG_ROOT}/activity.log
Enable Correlation Id	True

4.5 Change Log Level Details

Parameter	Value
Disable logging	False
Enable Log and Trace Correlation	False

ORACLE®

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
${SERVER_LOG_ROOT}/http_access.log

Access log maximum size
500   MB

Maximum number of historical files
1

✱ NCSA access log format
Common ▾

Error logging

☑ Enable error logging

✱ Error log file path
${SERVER_LOG_ROOT}/http_error.log

Error log maximum size
500   MB

Maximum number of historical files
1

✱ Error logging level
Warning ▾

Parameter	Value
NCSA Access logging Enable	True
Error logging Enable	True

ORACLE

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

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

ORACLE®

Parameter	Value
Serialization Session access	True

ORACLE

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 Webpshere 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 mentiond steps:

1.   Click New.

2.   Enter webSphereDefaultIsolationLevel for the name field.

3.   Enter 2 for the value field.

ORACLE®

ORACLE®

Websphere Server 8.5 Best Practices
[May] [2021]
Version 14.5.0.0.0

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