Improved Business Continuity with Native Oracle Database Connection Management for On-Premise Solutions

Leverage the capabilities of Oracle Connection Management with JD Edwards EnterpriseOne for Improved Business Continuity

October, 2021 | Version 1.0

Purpose Statement

This document provides an overview of features and enhancements included in EnterpriseOne Tools Release 9.2.6.0. The feature enhancement of using Oracle connection pooling over JD Edwards connection pooling is intended to help customers assess the business benefits of upgrading to Tools Release 9.2.6 and to plan your I.T. projects implemented in an on-premise architecture.

Disclaimer

This document, in any form, software or printed matter, contains proprietary information that is the exclusive property of Oracle. Your access to and use of this confidential material is subject to the terms and conditions of your Oracle software license and service agreement, which has been executed and with which you agree to comply. This document and information contained herein may not be disclosed, copied, reproduced or distributed to anyone outside Oracle without prior written consent of Oracle. This document is not part of your license agreement nor can it be incorporated into any contractual agreement with Oracle or its subsidiaries or affiliates.

This document is for informational purposes only and is intended solely to assist you in planning for the implementation and upgrade of the product features described. 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 in this document remains at the sole discretion of Oracle.

Due to the nature of the product architecture, it may not be possible to safely include all features described in this document without risking significant destabilization of the code.

Table of Contents

Purpose Statement **2**

Disclaimer **2**

Executive Summary **4**

Overview of Connection Pooling Technology **4**

Testing Summary 4

Metrics 6

Configuration of Oracle Connection Pooling **6**

Configuring Oracle Connection Pooling on Enterprise Server 6

Configuring Oracle Connection Poolg on HTML Server 7

Results Summary 7

Testing Summary 8

Conclusion **8**

Appendix A: Testing Information **9**

Pre-requisites 9

Configuring automatic kernel reconnection **Error! Bookmark not defined.**

Testing Environment Setup **Error! Bookmark not defined.**

Performance Characterization of automatic kernel reconnection with JD Edwards **Error! Bookmark not defined.**

Testing Use Cases **Error! Bookmark not defined.**

Testing Results and Analysis **Error! Bookmark not defined.**

Appendix B - Test Cases **10**

Appendix C – Test configuration **12**

List of images

Image 1. Enterprise ServerOracle OCI Configuration 7

Image 2. EnterpriseOne HTML Server UCP Configuration 7

List of tables

Table 1. EnterpriseOne Testing with Oracle UCP 5

Table 3. Kernel Reconnection Test Results **Error! Bookmark not defined.**

Table 4. EnterpriseOne Detail Testing Use Cases 10

Executive Summary

Implementing Oracle database connection pooling management for on-premise in Oracle JD Edwards EnterpriseOne makes database connection management and load balancing much more efficient. This connection management enhancement revamps the connection management layer within EnterpriseOne software to provide the latest of Oracle database connection pooling technology. This provides improved connection handling experience, thereby improving the system availability and business continuity for on-premise EnterpriseOne customers.

Overview of Connection Pooling Technology

A connection pool is a cache of database connection objects. The objects represents physical network database connections that for use by the EnterpriseOne application to communicate to the Oracle database. A session encapsulates user interactions with the database from the moment the user authenticates with the database until the user disconnects or times-out.

There are two types of connection pools that JD Edwards EnterpriseOne implements:

- Oracle database Universal Connection Pooling (UCP)

  UCP is used for JDBC connections to the Oracle database and is utilized by JD Edwards on the EnterpriseOne HTML Server.

- Oracle Call Interface (OCI)

  Oracle OCI is used for Oracle database communication to the Enterprise Server.

For the purpose of this document, both UCP and OCI, are combined as a single term, OCP, to represent the Oracle Connection Pooling technology and to describe the JD Edwards EnterpriseOne connection pooling scheme available in Tools Release 9.2.6.0 to work with the Oracle 19c database.

The Oracle database OCP pool implementation to communicate between the database and EnterpriseOne application is a better suited mechanism for building fault tolerance and high availability solutions than the EnterpriseOne proprietary connection pool for EnterpriseOne. Every release of Oracle Database has many enhancements to OCP.

Without implementing OCP, EnterpriseOne uses a proprietary connection pooling mechanism that is database agnostic to support multiple types of databases. This proprietary connection pool methodology has some limitations in terms of handling failover scenarios, which causes challenges in providing transaction continuity to customers. Due to these limitations, EnterpriseOne customers are required to restart their EnterpriseOne services to re-establish database connections whenever a node failure occurs in a RAC configuration. This is also a limitation for customers looking to adopt Oracle Autonomous Database as the connections during the rolling RAC patching are handled by OCP to ensure zero downtime patching. Additionally, the EnterpriseOne connection pool mechanism uses deprecated Oracle database code, which might cause issues with future database releases when support is ended for that deprecated code.

Implementing Oracle OCP effectively enhances the connection management layer within EnterpriseOne by adopting the latest Oracle database technology. Oracle recommends OCP to make EnterpriseOne ready to manage connections with future Oracle database releases. This is a key building block for all subsequent feature adoption for Oracle Autonomous Database and for using the ADB-Shared technologies.

Testing Summary

The main objective is the testing of interactive and batch processes in EnterpriseOne with Oracle OCP technology enabled. The test process included batch and interactive testing as shown in Table 1.

Table 1 refers the two test case categories – batch and interactive. Batch test cases is the submission of a set of nine short running UBEs. Interactive testing involves 40 simultaneous EnterpriseOne users performing actions in various interactive EnterpriseOne application modules. Test results performed with these use cases referred in Table 1 are covered in a subsequent section of this document. A detail listing of the more specific interactive and batch job description of use cases is specified in Appendix B.

TEST CASE	TESTCASE TYPE	DESCRIPTION
1	Batch	Submission of set of nine short running UBEs • Submission initiated using runube utility on the Enterprise Server • Approximately 300 batch jobs are executed for each test
2	Interactive	Interactive user applications • JMeter utility used to generation interactive user load • Load consisted of 40 interactive users running the different EnterpriseOne module application for a period of an hour

Table 1. EnterpriseOne Testing with Oracle UCP

For a comparison of the performance of the EnterpriseOne application, two test scenarios, listed in Table 2 are provided:

TEST SCENARIO	TESTCASE TYPE	SCENARIO DESCRIPTION
1	Connection Pooling	Test cases with JD Edwards connection and Oracle connection pooling configurations enabled
2	Oracle OCP with Oracle RAC failover	Test cases with JD Edwards and Oracle OCP connection pooling enabled during Oracle RAC failover event

Table 2. EnterpriseOne Test Scenarios

The test scenarios presented in Table 2 perform testing with connection pooling for the two different implementations, which are with and without OCP. The test scenarios extend to examine EnterpriseOne behaviour in an Oracle RAC failover situation.

The test use cases and scenarios presented in this document for the interactive and batch processes tested exercise a limited number of EnterpriseOne kernel processes on the Enterprise Server along with those database requests coming from the EnterpriseOne HTML Server. The kernel processes that are the focus of this analysis are presented below in Table 3.

Although the major focus of the testing is on the EnterpriseOne kernel processes listed in Table 3, a small number of requests are designed by the use cases to exercise the Oracle UCP connection pool from the EnterpriseOne HTML Server.

KERNEL DEFINITION	KERNEL NAME	DESCRIPTION
[JDENET_KERNEL_DEF2]	UBE	Controls the submission of batch process requests from clients and schedules batch jobs in the job queue.
[JDENET_KERNEL_DEF4]	Security	Provides the credentials and application tokens for security access to the EnterpriseOne application. Additionally it also handles all user profile and security related table updates.
[JDENET_KERNEL_DEF6]	Call Object (COK)	Responsible for executing the business function logic through requests received from JDENET. If the user has already established a request to the COK process, JDENET will route subsequent requests to the same COK process. The only time a business function will run on a different COK process is when the BSFN is OCM mapped to a different server. Only COK kernels support multi-thread processes. All other kernels are meant to handle data/cache one at a time.
[JDENET_KERNEL_DEF30]	Metadata Kernel	The metadata kernel runs in a JVM and processes the XML specification access requests readable by the call object kernel.

Table 3. EnterpriseOne Kernel Processes

Metrics

Metrics provide an indication whether the application is performing within the range of expected. For testing EnterpriseOne for use with Oracle's OCP technology, the criteria for success was the metrics surrounding the success or failure of the interactive and batch process using Oracle OCP. The test and scenario cases are a means to exercise the JD Edwards and Oracle connection pool configuration evaluated in this testing.

The tool used to measure the performance metrics for interactive user performance is Apache JMeter. This tool is an open source software developed using Java and is designed to load test multi-user load functional behaviour. The metrics collected in JMeter represent a server-side picture of performance. JMeter measures the time gap for EnterpriseOne HTML interactive users where the browser performs HTTP requests to the WebLogic HTML Server. Batch testing metrics compared the runtime of the batch for each of the configuration parameters tested.

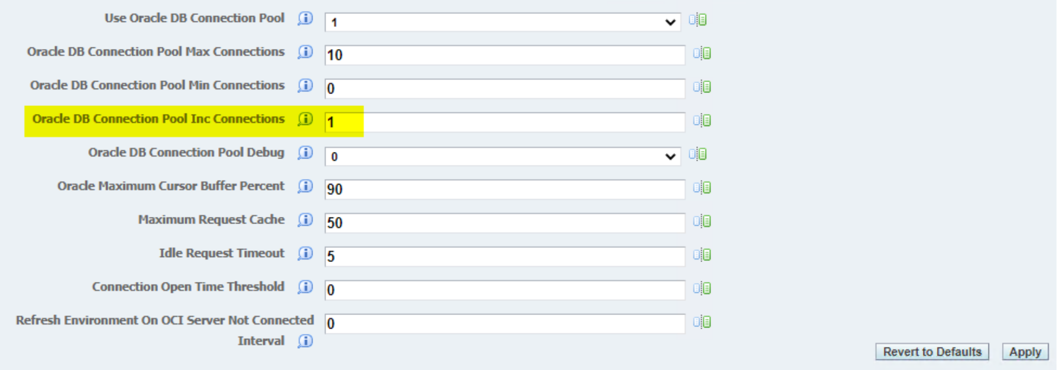
Configuration of Oracle Connection Pooling

Oracle OCP can be configured directly from the Server Manager Console in EnterpriseOne. It can be configured separately for the OCI and UCP configuration. Images 1 and 2 below illustrate enabling Oracle OCP in an EnterpriseOne architecture. When these values are not set, EnterpriseOne uses the default JD Edwards legacy connection pooling configuration.

Configuring Oracle Connection Pooling on Enterprise Server

The configuration of Oracle OCP on the Enterprise Server is performed using the following procedure:

Server Manager Console -> <Enterprise Server> -> Configuration -> Advanced -> Bootstrap Datasource



Use Oracle DB Connection Pool: 1

Oracle DB Connection Pool Max Connections: 10

Oracle DB Connection Pool Min Connections: 0

Oracle DB Connection Pool Inc Connections: 1

Oracle DB Connection Pool Debug: 0

Oracle Maximum Cursor Buffer Percent: 90

Maximum Request Cache: 50

Idle Request Timeout: 5

Connection Open Time Threshold: 0

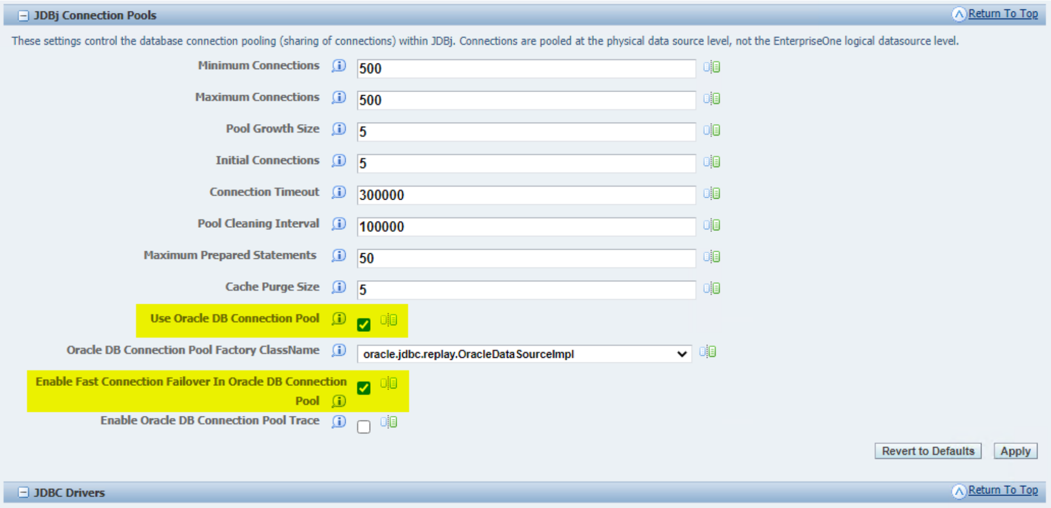
Refresh Environment On OCI Server Not Connected Interval: 0

Image 1. Enterprise ServerOracle OCI Configuration

The highlighted section in Image 1 shows that Oracle OCP is enabled on the Enterprise Server. This is a toggle of 0 (OFF) and 1 (ON) for Oracle OCP. The variables below this setting are tuning configurations for Oracle OCP. Additional performance tuning recommendations for Oracle OCP and EnterpriseOne are covered in the appendix.

Configuring Oracle Connection Poolg on HTML Server

Server Manager Console -> <HTML Server> -> Configuration -> Advanced -> Database -> JDBj Connection Pool



JDBj Connection Pools

These settings control the database connection pooling (sharing of connections) within JDBj. Connections are pooled at the physical data source level, not the EnterpriseOne logical datasource level.

Minimum Connections: 500

Maximum Connections: 500

Pool Growth Size: 5

Initial Connections: 5

Connection Timeout: 300000

Pool Cleaning Interval: 100000

Maximum Prepared Statements: 50

Cache Purge Size: 5

Use Oracle DB Connection Pool: ☑

Oracle DB Connection Pool Factory ClassName: oracle.jdbc.replay.OracleDataSourceImpl

Enable Fast Connection Failover In Oracle DB Connection Pool: ☑

Enable Oracle DB Connection Pool Trace: ☐

Image 2. EnterpriseOne HTML Server UCP Configuration

To enable Oracle OCP for the HTML Server, mark the check box in the Server Management Console. There are also two additional Oracle OCP configurations for the HTML Server that are described in the appendix.

Testing Results Summary

This section provides a comparison of the interactive user testing, batch user test comparison, and a testing summary of the results of use case and scenario of EnterpriseOne connection pooling and Oracle OCP connection pooling.

Interactive User Testing Comparison

JMeter metrics formed the basis of this comparison between EnterpriseOne connection pooling and Oracle OCP connection pooling. JMeter can only measure differences in any great accuracy in performance that is less than 250 ms. The overall comparison yielded an overall average of 130 ms difference between legacy EnterpriseOne connection pooling and the Oracle OCP connection pooling. Given the performance accuracy of JMeter, this difference is considered as indistinguishable between the two different connection pooling configurations.

In theory, the Oracle OCP connection pooling schema should have slightly more overhead that that of the legacy EnterpriseOne connection pooling because of the additional caching of information that takes place on the Oracle connection pooling configuration. To provide more fault tolerance and high availability functionality, Oracle connection pooling must cache more information to retain the state of the process using the Oracle connection pool in the case of database connection disruptions. This level of performance difference is hard to measure using the accuracy of JMeter.

With regard to interactive user testing during an Oracle failover event on an Oracle 2-node RAC configuration, similar numbers of users failed to complete successfully with both database connection configurations. With regard to the JD Edwards connection pooling, approximately 9% of the users failed. In the configuration with Oracle connection pooling, 6% of the users failed during a forced Oracle RAC failover event. In both circumstance, EnterpriseOne services had to be manually recycled to return the application to fully functional operations.

EnterpriseOne application failures for interactive users are problematic in providing high availability and fault tolerance in the EnterpriseOne architecture. In fact, many software applications suffer from the same inherent issues with regard to keeping application process state during a failover. There are two main contributing factors to these failures

1. EnterpriseOne application loses state information when the database connection is lost. The JD Edwards EnterpriseOne application is designed to retry connection to the database in short term network and communication outages, but can fail if those attempts do not resolve in a timely fashion.
2. EnterpriseOne interactive applications are constrained by a number of timeouts controlling their communication to the database. If these timeouts occur before connections are re-established, than the application process will fail.

The goal of implementing Oracle connection pooling is to provide better a better foundation of technology for allowing greater business continuity between the communication of the EnterpriseOne application and the database. A decrease of 3% failure rate was a good indication that Oracle connection pooling technology benefits EnterpriseOne in terms of business continuity.

Batch User Testing Comparison

Batch testing showed similar results in the comparison between runtimes and experience during Oracle RAC failover events. In the case of runtimes, an average runtime of 2% difference was measured in favor of the JD Edwards connection pooling. With regard to Oracle RAC testing, the same number of failures were measured for both connection pooling configurations.

Testing Summary

The purpose of the testing was to determine if there were significant performance differences between the JD Edwards connection pooling configuration and Oracle connection pooling. Only relatively minor differences were observed in the testing.

Conclusion

Oracle connection pooling (OCP) demonstrated all the same qualities and functionality as the legacy EnterpriseOne connection pooling. Overall, the legacy EnterpriseOne connection pooling performed slightly better, likely because of its lightweight design and streamlined processes. Oracle connection pooling, while exhibiting a less than 3% overhead, can provide better business continuity and stability moving forward for the EnterpriseOne application and provide a better strategy for implementation in architectures such as Autonomous Database and other high availability and fault tolerant solution for Oracle JD Edwards customers.

Appendix A: Testing Information

Pre-requisites

In order to configure your JD Edwards to leverage automatic kernel reconnection, the release levels required are:

- JD Edwards EnterpriseOne Applications 9.2
- JD Edwards EnterpriseOne Tools Release 9.2.5.3 or above
- Supported Database Versions:
    - Oracle Database 19c (19.11 or above)
    - Oracle Autonomous Database 19c running on Dedicated Exadata Infrastructure, Shared Exadata Infrastructure or Exadata Cloud@Customer
    - Microsoft SQL Server 2019
    - IBM DB2 LUW 11.5.4 on IBM AIX 7.2
    - IBM DB2400 on IBM i 7.3 and 7.4
- Testing was performed with JD Edwards and Oracle connection pool enabled

Appendix B - Test Cases

The test cases for interactive and batch processes are described in the tables below.

USE CASE	TESTCASE TYPE	UNIT TESTING	LOAD TESTING
1	Batch	Submission of long running UBEs -R31410, R3483, R42565 and R43500	Submission of 9 short running UBEs and long running UBEs mentioned in the unit testing
2	Interactive	Execution of below use case script in JMeter for 10 User 10 Iterations • Supplier Ledger Inquiry (P0411) • Apply Receipts (P03B102) • Work Order Completion (P31114) • Sales Order Entry (P42101)	All the Interactive scripts mentioned in unit test executed together as a 40 User test with 10 Iterations each

Table 4. EnterpriseOne Detail Testing Use Cases

APPLICATION NAME	DESCRIPTION
P03B102	Receipt Entry
P0411	Supplier Ledger
P051141	Daily Timecard Entry
P17500	Case Management
P31114	Work Order Completion
P3411	MRP/MPS Detail Message Revisions
P4113	Inventory Transfer
P42101	Sales Order
P4310	Purchase Order
P4312	PO Receipts
P4314	Voucher Match

Table 5. Interactive Applications

UBE NAME	DESCRIPTION
R0004P	User Define Codes Record Types Print
R00067	Business Unit Translation Report
R0006P	Business Unit Report

R0008P	Date Patterns Report
R0010P	Company Constants
R0012P1	AAI Report – One Line per AAI
R0014	Payment Terms Report
R0018P	Tax Detail Report
R00425	Organization Structure Report
R01402W	One Line per Who's Who Report
R42565	Print Invoices

Table 6. UBEs

Appendix C – Test configuration

The JD Edwards EnterpriseOne components configured in the architecture for the testing process discussed in this document are listed below:

The JD Edwards EnterpriseOne components were implemented on Oracle Cloud Infrastructure as a standard VM.

JD Edwards EnterpriseOne Enterprise Server:

- Oracle Enterprise Linux 7.8
- Oracle Database 19c (19.12.0.0.0) client

Database Server:

- Oracle Enterprise Linux 7.8
- Oracle Database 19c

  Enterprise Edition Release 19.12.0.0.0 - Production

HTML Server

- Oracle Enterprise Linux 7.8
- WebLogic Server 12c (12.2.1.4.0); Java JDK (1.8.0_272)

Deployment Server:

- Windows Server 2016 Standard
- VM Standard 2.4 with 4 OCPUs
- 4 OCPUs x Intel Xeon Platinum 8167M CPU @ 2.00 GHz
- 60 GB RAM

Server Manager Console:

- Oracle Enterprise Linux 7.8

Test Controller:

- Windows Server 2016 Standard
- VM Standard 2.4 with 4 OCPUs
- 4 OCPUs x Intel Xeon Platinum 8167M CPU @ 2.00 GHz
- 60 GB RAM
- Apache JMeter 5.1.1.r1855137

Software: JD Edwards EnterpriseOne Application 9.2 Update 4 with Tools Release 9.2.6.0

Connect with us

Call +**1.800.ORACLENTERPRISEONE** or visit **oracle.com**. Outside North America, find your local office at: **oracle.com/contact**.

blogs.oracle.com   facebook.com/oracle   twitter.com/oracle

This device has not been authorized as required by the rules of the Federal Communications Commission. This device is not, and may not be, offered for sale or lease, or sold or leased, until authorization is obtained.

Disclaimer: If you are unsure whether your data sheet needs a disclaimer, read the revenue recognition policy. If you have further questions about your content and the disclaimer requirements, e-mail REVREC_US@oracle.com.