

# 1. Introduction

## 1.1. Project/Component Working Name:

Connectors, Connection Pool, JDBC features in GlassFish V3

## 1.2. Name and e-mail address of Document Author(s)/Supplier:

Smitha Prabhu: Smitha.Prabhu@sun.com

## 1.3. Date of This Document:

10/2/09

## 1.4. Revision History

Date	Change	Revision
10/2/09	Draft Created	1

# 2. Project Summary

## 2.1. Project Description:

This document describes the testing scenarios for the functionality changes and enhancements to Connectors, Connection Pooling, JDBC-RA

## 2.2. Risks and Assumptions:

- Risks: None.
- Assumptions: None.

## 3.1. Problem Area:

Following enhancements need to be introduced in GlassFish v3 to overcome or to provide :

- Absence of statement caching support for jdbc applications
- Aid to call a user defined validation mechanism
- No mechanism to execute a user provided SQL for initialization
- No mechanism to flush a connection pool and re-initialize the connections
- No mechanism to have application specific (self-contained) resources
- Ease of use and configurability features
- Application Server vendor specific support for JSR 322

## 4. Technical Description:

### 4.1. Details:

The Connectors, Pooling, JDBC-RA would provide the some enhancements in this release details of which can be got from the engineering one page @ <http://wiki.glassfish.java.net/Wiki.jsp?page=GFV3JDBCOnePager>

### 4.2 Test Scenarios

#### 4.2.1 Statement Caching

- a) Set statement-cache-size to -ve number (negative case)
- b) Statement caching implies that the statement is cached only for the particular connection. So, execute a query (to be cached) with all the connections. Store the object id of the query for each connection. Make sure that the same object is got for that particular connection.
- c) Loop it for more than the max. no. of connections and check for LRU (the object id should be different)

#### 4.1.2 Custom Validation

Implementation of the public interface [org.glassfish.api.jdbc.ConnectionValidation](#) to do custom validation for different DBs

- org.glassfish.api.jdbc.DerbyConnectionValidation
- org.glassfish.api.jdbc.MySQLConnectionValidation
- org.glassfish.api.jdbc.PostgresConnectionValidation
- org.glassfish.api.jdbc.OracleConnectionValidation

#### 4.1.3 Init Sql

Executes SQL query during pool creation

- a) Grant permissions to particular user on table and try queries

#### 4.1.4 Tracing SQL Statements

- a) Logging using the inbuild implementation
- b) implementation the public interface [org.glassfish.api.jdbc.SQLTraceListener](#)

#### 4.1.5 Introspection of JDBC drivers

Manual test to add jdbc-drivers in the GF\_INSTALL/lib directory and “**detect installed drivers**” using GUI.

#### 4.1.6 Flush Connection Pool

a) Use all the connections and then flush the connection pool to recreating the connections in the pool and bringing the pool to the steady pool size. This can be checked with making sure that new connection objects are created.

b) Can be tested manually thro' the GUI.

#### 4.1.7 Built in factories for Custom Resources

Resources to be used for providing data to the applications using:

a) **JavaBeanFactory**

b) **PrimitivesAndStringFactory**

c) **URLFactory**

d) **PropertiesFactory**

#### 4.1.8 Associating multiple connections with a thread

a) Set the no. of connections for a thread .

#### 4.1.9 Switch off Pooling

a) set Associate-with-thread to true and check if an exception will be thrown

b) Flush connection pool, when pooling is switched off results in an exception

c) A warning will be logged when the following attributes are active in a switched off pool.

- connection-validation
- validate-atmost-once-period
- match-connections
- max-connection-usage
- idle-timeout

#### 4.1.10 Default values of JDBC Connection Pool Attributes

a) connection-validation-method "**table**"

b) wrap-jdbc-objects "**true**"

#### 4.1.11 Ping Pool during creation

a) ping the pool to validate the pool attributes and make sure that the pool creation was successful.

b) Will be tested thro' the GUI as well.

#### 4.1.12 java.sql.Driver based pooling support

a) **datasource-classname** attribute will be an optional attribute when res-type is set to java.sql.Driver, whereas the **driver-classname** attribute will be mandatory.

## 5. Reference Documents:

1. [JSR 322 - Java EE Connector Architecture 1.6 Specification](#) ↗
2. [JSR 322 - Reference Implementation - Half Pager](#) ↗

## 6. Schedule:

### 6.1. Projected Availability:

Aligned to overall GlassFish V3 Final release schedules.