1.0 Summary of Model Change Request

In the current version of JTLS, if a user is working on more than one scenario at a time, a separate machine must be assigned for each scenario to execute Glassfish. The desire is to run several Glassfish instances on a single machine, if the user needs to do so. In addition, some improvements are desired for the Database Development System (DDS) Configuration Program (DCP). The DCP fulfills two functions:

- Allows the user to configure the DDS for a specific scenario by indicating the machine on which the scenario’s Glassfish server executes and the names and passwords of the DDS users that are allowed to have access to the scenario.

- Manages the execution of the Glassfish server. From the DCP, a user starts and stops the scenario’s Glassfish server. Currently to accomplish this second function, the DCP must be executing on the machine assigned to the Glassfish Server. A user cannot start and stop the Glassfish Server from a different machine. This ECP will also solve this issue.

2.0 Design Summary

To solve these problems, JTLS must move to a newer version of Glassfish. Glassfish Version 3.1.2 will be delivered as part of JTLS 4.1. Because a new version of Glassfish is being delivered with JTLS, there also need to be a few minor changes to the method used to manage the AAR version of Glassfish. Using this new version of Glassfish, Technical Control will be able to configure Glassfish in anyway that they want.

3.0 Detailed Design

Pictorially, the current situation, for a user that wants to build and run three different scenarios simultaneously can be viewed as shown in Figure 1. Only one instance of Glassfish can execute on a machine at once, and to start and stop Glassfish, a user has to be on the machine. The user does not need to be physically sitting at the machine, but the DCP, which is used to start the DDS Glassfish instance and the Web Services Manager (WSM), which is used to start the AAR Glassfish instance must be executing on the machine on which Glassfish is designated to execute.
Figure 1 shows just one possible configuration that Technical Control can setup and distribute the instances of Glassfish that are needed for our sample situation of building and executing three scenarios simultaneously.

Figure 2 shows just one possible configuration that Technical Control can setup and distribute the instances of Glassfish that are needed for our sample situation of building and executing three scenarios simultaneously.
In Figure 2, note that JTLS must still run a separate instance of Glassfish for each of the three scenario building tasks and each of the three scenario execution tasks. The difference is that each Glassfish instance can be placed on any computer on the network and the DCP and WSM used to start each instance can be executing anywhere on the network.

If the DCP and WSM are not on the same machine as the Glassfish instance that is being started, the process will be started by opening a Secure Shell (ssh) connection to the appropriate machine. In Figure 2 this is shown by the “Red” line. When opening the ssh connection, the user will need to enter the password for the JTLS account. This is already a requirement when starting other services using the WSM.

If Glassfish is to be started on the same machine as the DCP or WSM that is starting the process, then the Glassfish instance is simply started as a background process. There is once such example of this situation in Figure 2. The WSM for Scenario C is executing on the same machine that the AAR Glassfish instance is running; therefore, no ssh connection is required.

If the user decides to change the machine or the port on which Glassfish should execute, the DCP or ICP will check whether Glassfish is currently executing with the old values. If it is, the program will not make the change and will warn the user that Glassfish needs to be manually stopped. We decide to make this a manual stop vice and automatic stop because, Technical Control needs to make sure that everyone is logged off prior to making this change.

Finally the previous version of JTLS did not make it clear that a minimum of 12 ports are needed to properly execute Glassfish. Each Glassfish instance has two ports, an administrative port and an http port. The new versions of the DCP and the ICP will include a check that these two specified ports have at least 12 free port numbers in between the two numbers. For example, an administrative port of 18000 and an http port of 18020 would be legal but an administrative port of 18000 and an http port 18005 would not be legal because it has only four free ports between the two values.

In addition, the ICP will check no other processes specified in the file, use any of the reserved Glassfish ports.

Although there is a significant effort required to make this all work in an automated manner, from the user’s perspective, the following tasks must be accomplish to implement this design:

- The DCP needs to be changed so the user, as part of the setup procedure, enters the port numbers that Glassfish will use.
- The ICP needs to be changed so the user, as part of the setup procedure, enters the port numbers that Glassfish will use.
- The DCP needs to be changed to query the user for the needed ssh password when attempting to start the DDS scenario’s Glassfish instance on a different machine.
• The WSM needs to be changed to query the user for the needed ssh password when attempting to start the AAR scenario’s Glassfish instance on a different machine. Note that the WSM already requests this password once when any of the web services are started on a different machine. This will also be true for the Glassfish. The portion of the WSM logic will refer to the single password entered by the user as is done for all of the other web services.

Finally, this ECP must also consider the changes required by JTLS-2012-11205 Web Services Manager DIACAP Requirement. When the DCP or WSM is executing on a different machine that the Glassfish instance for which it is responsible, all of the health information concerning the instances will be accomplished through the ssh tunneling capability fully described in that design.

4.0 Data Changes

No database parameter or structure changes are required to implement this design.

5.0 Order Changes

No order parameter or structure changes are required to implement this design.

6.0 JODA Changes

No JODA Data System parameter, structure, or protocol changes are required to implement this design.

7.0 Test Plan

To properly test this ECP, the user should have at least three JTLS 4.1 scenarios loaded into Oracle.

7.1 Test 1 Title

Purpose: [Describe the specific feature, function, or behavior to be tested or measured.]

Step 1: Text

Step 2: Text
Expected Results: [Describe the specific model behavior to be observed.]

7.2 Test 2 Title

Purpose: Text

Step 1: Text

Step 2: Text

Step 3: Text

Expected Results: Text