Test Design Specification Document

Introduction

Document Scope:
The scope of our project consists on the essential functionality of the GlassfishV3 Application Server's Administration GUI.

Document Structure
We will of course validate that the application can be installed and deployed successfully within our testing environments. During this phase, we will validate the results of installation under various conditions by partitioning the installation into functionality partitions:

- Successful Silent Installation
- Successful Interactive Installation
- Failed Installation/Invalid Environment
- Failed Installation/Invalid Silent Mode Parameter File

In addition to this, we perform a simple test of the provided uninstaller script to confirm that it removes glassfish from its target directory.

After our round of installation and deployment tests have passed, we will continue and test the various modules within the Administration GUI. We will again partition these modules as follows:

- Registration Module
- Enterprise Server Module
- Applications Module
- Lifecycle Module
- Resources Module
- Configuration Module

We will further partition the subcomponents (if any) of the modules, such that our coverage will encompass the core functionality of each module. For simpler subcomponents, such as an administrator password-change module, we will employ a decision table to encompass all possible use case scenarios for the feature. In some of the more complex subcomponents, we will test the functionality of the system in a branch based techniques. In this case, we will treat GUI page steps as predicate nodes. This will allow us to map out potential paths of functional execution and ensure coverage for all reachable paths within the given module.

Test Design

Registration Component

Goals and Coverage:
Confirm that the registration module appropriately registers the Glassfish application. Coverage should be for all combinations of valid/invalid usernames and passwords.

Test Case Information
The tests are partitioned into two: Valid input and Invalid input. Next, using a simple decision table, the presence/validity of username/password configurations can be verified.

Enterprise Server Component

Goals and Coverage
Confirm that the subcomponents comprising the Enterprise Server module perform their respective tasks via validation that the Admin Console GUI retains the information it is configured with. In other cases, we will expect it to reject certain configurations due to faulty input values.

Test Case Information

- **General Subcomponent**
- Stop and restart the domain by following prompts. View Log files
  - **Admin Subcomponent**
    - **Application Configuration**
      - Modify configuration settings and confirm that state is retained. Confirm that the reset feature correctly restores the system to its default state.
    - **Domain Attributes**
      - Record and modify the Log Root locations. Validate log file location via terminal. Restore original settings and repeat validation.
- **Admin Password Subcomponent**
  - Using simple decision table, confirm that the password can be modified if and only if the correct combination of parameters are specified.
- **System Properties Subcomponent**
  - Add a system property, and confirm persistence of state by navigating to the advanced subcomponent and confirming presence of additional property. Delete this property and repeat procedure to confirm that state restoration.
- **Monitor Subcomponent**
  - **Application Monitoring**
    - confirm that the user can configure system monitoring. Confirm monitoring levels can be changed for application/component combination.

- **Server Monitoring**
  - Confirm that user can change monitoring levels for different servers.
- **Resource Monitoring**
  - Confirm that user can change monitoring levels for different resources

Applications Component

Goals and Coverage
Confirm that the Admin Console can successfully perform the deployment of the file types for which it claims to support; therefore coverage is complete for the core deployment of the supported deployable file types.

Test Case Information
Using simple partitioning, and acquisition of simple test archive files from the internet, we can divide the tests into a single partition for every file type. We can then confirm that the file has been deployed, can be disabled, and undeployed.

Lifecycle Modules Component

Goals and Coverage:
Confirm that Lifecycle modules can be created.

Test Case Information:
Verify that the module can be successfully created and removed.

Resources Component

Goals and Coverage
Validate the functionality of the component by partitioning based on this module's subcomponents. Coverage encompasses the base functionality of each subcomponent.

Test Case Information

- **JDBC Resource Subcomponent**
  - Create a JDBC resource and verify that appropriate errors are reported due to lack of database configurations**.**
- **Connection Pool Subcomponent**
  - Create connection pool and verity that the test ping fails due to lack of database configurations
- **Connectors Subcomponent**

    - Verify that the connector resources can be created. Determine whether a non functional connection pool (created in previous step) is still visible. Verify an Admin Object Resource, and a Work Security Map can be created and deleted.
- **Resource Adapter Configurations Subcomponent**
    - Verify an Adapter Configuration can be created
- **JMS Resources Subcomponent**
    - Verify that connection factories can be configured, and that errors are appropriately thrown due to faulty data. Analogously test destination resource configuration.
- **Java Mail Session Subcomponent**
    - Verify a new JMS can be created and errors thrown for faulty parameters
- **JNDI Subcomponent**
    - Verify a new JDNI (customer and external) resource can be created and errors thrown for faulty parameters

Configuration Component

Goals and Coverage
Validate the functionality of the component by partitioning based on this module's subcomponents. Coverage encompasses the base functionality of each subcomponent.

Test Case Information

- **JVM Settings Subcomponent**
    - Add a JVM property and confirm that it can be subsequently deleted with simple partition testing.
        - **Path Settings**
        - **JVM Options**
        - **Profiler**
- **Logger Settings Subcomponent**
    - Modify the location of the system log file and confirm this setting persists. Restore default data and repeat via simple partition testing
        - **Log Levels**
            - confirm that log levels can be changed for different components
- **Web Container Subcomponent**
    - Add and delete a system property underneath each of the following tabs via a two level partition testing method.
        - General
        - Session
        - Manager
        - Store
- **EJB Container Subcomponent**
    - Add and delete a system property underneath each of the following tabs via a two level partition testing method.

        - EJB Settings
        - MDB Settings
- **Ruby Container Subcomponent**
    - Specify a home for jRuby, confirm persistence and remove the value.
- **JMS Subcomponent**
    - **JMS Hosts Subcomponent**
        - Add a new JMS Host, confirm persistence, and remove via partition testing.
    - **Physical Destinations Subcomponent**
        - Add and delete a JMS physical destination. Flush the locations and confirm successful system response with partition testing.
- **Security Subcomponent**
    - **Realms Subcomponent**
        - Add and delete a realm via partition testing.
    - **Audit Modules Subcomponent**
        - Add and delete an audit module via partition testing.
    - **JACC Providers Subcomponent**
        - Add and delete a property via partition testing.
    - **Message Security Subcomponent**
        - Successfully configure the system with all available configurations with simple decision table testing.
- **Transaction Service Subcomponent**
    - Add and delete a property via partition testing.
- **HTTP Service Subcomponent**
    - Enable/disable the logging, and add and delete a property, both via partition testing.
- **Virtual Servers Subcomponent**
    - Add and delete a Virtual Server via partition testing.
- **Network Configuration Subcomponent**
    - **Network Listeners**
        - Add and delete via partition testing
    - **Protocols**
        - Add and delete via partition testing
    - **Transports**
        - Add and delete via partition testing
- **Thread Pools Subcomponent**
    - Add and delete via partition testing
- **Object Request Broker Subcomponent**
    - Add and delete a property. Modify the thread pool ID via decision table testing. Confirm the defaults can be restored.
- **Admin Service Subcomponent**
    - Add and delete via partition testing
- **Connector Service Subcomponent**
    - Modify the Connector Service configurations via decision table testing, confirming persistence and that defaults can be restored.
- **Monitoring Subcomponent**

- Use decision table to test that all combinations of enabled/disabled can be configured for the following settings:
  - Monitoring Service
  - Monitoring MBeans
  - DTrace
- Due to the magnitude of combinations, use simple partition testing for setting some of installed components (individually) to:
  - Low
  - High
  - Off (Restoration to default state)