Test One Pager

Glassfish V3 Admin GUI

Version	Comments	Date	Autho
1	First Draft	April 15th, 2009	Yifeng Luo
2	Second Draft(modify based on Anissa's comments)	June 8th, 2009	Yifeng Luo
3	Third Draft(modify based on Ken's comments)	June 9th, 2009	Yifeng Luo

Table of Contents

1.0 Introduction

1.1 Overview
This document specifies the plans to test Glassfish V3 Admin GUI. I will update this document once I have more details about the features to be implemented.

Details for test cases will be documented in the Test Specification

1.2 Objective
To provide the best quality for Admin GUI component in a permissible time

2.0 Developer one pager

Please click this link

3.0 Features to be Tested

3.1 New Functionality:

3.1.1 Scripting (more details to come)

Test deployment for JRuby and Jython/Django applications
Test configuration of JRuby and Jython/Django applications.
Test monitoring of the JRuby container

3.1.2 Non-war deployments

Test server side and client side deployment of war, ear, jar and rar files
Test redeployment of war, ear, jar and rar files
Test directory deployment, enable and disable applications
Test correct engines are displayed for deployed applications, correct subcomponents are listed

3.1.3 Grizzly Configuration

Configuration --> Transaction Service

Test modification of transaction service settings, load defaults

Configuration --> Thread Pools

Create, delete and edit thread pools

Configuration --> Network Config

create, delete and edit network listeners
create, delete and edit protocols

3.1.4 Logging support

Test GUI displays correct logging values from logging.properties file (under domains\domain1\config\logging.properties)
Test logging properties and values can be written to logging.properties file according to user input

3.1.5 MQ Administration (more details to come)

test the following four tree nodes: connection factories, destination resources, JMS hosts, physical destinations

3.1.6 Nice to have features (more details to come)

3.1.6.1 dashboard:

some UI mock up page can be found under here

Test main dashboard functionality.
Test user can add contents to user dashboard page.
Test user can change layout of dashboard gadgets. For example: move gadgets up/down within a column or move gadgets
between columns.
Test individual gadgets, test GUI can display relavant information according to user preferences, test GUI can edit user

preference for the gadgets.
More details to come....

3.2 Existing Features:

3.2.1 Common Task (this will likely be replaced by dashboard feature under section 3.1.6.1)

3,2,2 Product registration

register product using new user or existing users

3.2.3 Application Server

- General
- JVM Settings
- Advanced
- Monitor

3.2.4 Login and Logout of Console

by default, no authentication is required

when new user created with asadmin group, test login and logout of console

3.2.5 Applications

deployment, redeployment, undeployment, directory deployment
enable, disable applications
list application sub components
edit applications. launch

3.2.6 Resources

create, edit, delete, enable and disable of jdbc resource
create, edit, delete of jdbc connection pool
ping jdbc connection pool

create, edit, delete of connector resource
ping connector connection pool

3.2.7 Configuration

- Web Container
  - General/Session/Manager/Store Property
- Transaction Service (covered under new functionality)
- HTTP Service
  Configure HTTP Service Access Logging
- Visual Servers

- List/Create/Delete/Modify Virtual Services

* Thread Pools (covered under new functionality)

* Network config (covered under new functionality)

* Monitoring
Test set different logging level (HIGH, LOW, OFF)

* Security

Test creation of different realm, edit realm
Test creation of audit modules, edit audit modules

3,2,8 Update tool

install available add-ons
uninstall installed components
configure download directly or through a proxy server

3.3 508 compliance test

We are still under discussion how to do 508 testing, Will update this session when I have more info.

3.4 Online Help Testing

Invoke help page under a certain GUI page, make sure the help content matches with current GUI content.

3.5 O.S. and Browsers Supported

For javaone release, refer to the following link for test matrix
http://wikihome.sfbay.sun.com/CentralSWIQA/Wiki.jsp?page=GFv3JAVAONESQETestMatrix

For v3 release, refer to the following link for supported O.S. and browsers
http://appserver.sfbay.sun.com/Wiki.jsp?page=SupportedPlatforms

3.6 Database support

For javaone release,

Derby
MySQL 5.1/MySQL Connector/J Driver 5.1

4.0 Features Not Tested

- Any components not listed above

5.0 Deliverables

- Test one pager - this document
- Test specification document for instructions on running test cases -
- Test reports and logs - output from test execution runs, posted on web page
- Test suites - manual and automated test cases
  - All manual test cases will be documented - target 100%
  - Automated test cases will be automated, the plan is to have 85% by the FCS time frame
- Bugs filed on IssueTracker

6.0 Test Methodology

- A set of manual test cases will be delivered/executed for verifying the supported features listed in 3.0
- Automation Tools (SilkTest and Selenium)
- All manual test will be automated and run at weekly promoted build for adhoc and regression verification
- Other comprehensive test and test tools will be conducted to enhance the quality.
- Will provide test status, recent issues related to promoted build to FishCAT members once there is a request from the community. Will provide support to students program regarding GUI testing methodology and scenarios.

D