

GlassFish V2— Usage Profiles

1. Introduction

1.1. Project/Component Working Name

Usage Profiles for GlassFish

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

Kedar Mhaswade (Kedar.Mhaswade@Sun.Com)

1.3. Date of This Document

Feb. 16, 2007

2. Project Summary

See [section 3](#).

2.1. Project Description

Provision of custom and predefined usage profiles.

2.2. Risks and Assumptions

3. Problem Summary

3.1. Problem Area

An application server distribution consists of two main components -- a set of binaries and a set of templates. These templates are processed in a particular manner to create certain configurations out-of-the-box. A configuration is then tweaked after it is created in such a way that it can support (or not support) certain capabilities. It is just quite an administration-intensive task in that the administrator has to execute several configuration operations. It is also important that we impart a certain shape and form to the configuration right from the beginning.

Customers have also expressed interest in being able to *customize* the configuration creation process in a predefined manner. This is because the configuration that they want to start with denotes their typical application server runtime. Most of the configuration is stored in an XML document named "domain.xml".

The advent of GlassFish made it mandatory to plug-in different implementations for a particular functionality rather than rigidly bundling the components in a particular distribution. This bundling was something that had to happen in earlier releases. An example would be if the user downloaded the Enterprise Edition product, the session persistence had to be based on the HADB (third-party) software. This is a bit of a constraining.

3.2. Justification

GlassFish requires clustering capability and a way to configure various components of application facilitate the typical use of the software.

4. Technical Description

4.1. Details

The general approach is to be able to process the configuration files in a customizable manner such that a particular configuration is created. It is expected that administrators augment the process using the techniques like XML style-sheets. A number of predefined profiles are bundled along with the distribution so that existing users continue to configure the product in a way they used to.

Administrator runs a single command "asadmin create-domain" specifying the name of the profile for which configuration should be created. A profile is a well-defined set of files according to interface table below and are stored in a known location. When the domain.xml is to be created during the create-domain process, a properties file that contains various customization hooks is read and processing is according to that file. In short, it provides ways and means to control how the initial domain.xml will look like. This approach allows for the customization because more often than not, administrators are interested to build upon existing templates and tweak them using transformation techniques.

In order to fully support the created configuration at runtime, all the binary components of the server are available in all the distributions. This is another deviation from the previous releases, where the installation dictated the name and number of product packages installed on the system.

This project helps us to reach a transition stage where a particular installation of software packages give co-existence of domains with different capabilities rather than rigidly binding the type of configuration at runtime with the type of installation. This was a limitation with earlier releases.

This project also gives us a vehicle to optimize predefined profiles for their intended use. For example, the *developer* profile by default configures the server such that web server's access logs are disabled, for better responsiveness. Several such improvements were suggested by the users of the GlassFish application server, on the discussion forum.

4.2. Bug/RFE Number(s)

4.3. In Scope

This project will provide an interface to customize the application server domain creation process. This provides such a capability to only the domain.xml file. The project guarantees that backward compatibility is preserved.

4.4. Out of Scope

It is not the aim of this project to provide customization support for all the configuration files (such as startup/shutdown scripts, security policy files) that are required for proper functioning of the server runtime.

4.5. Interfaces

4.5.1 Exported Interfaces

| Interface | Stability (Former Stability) | Comments |
|--|------------------------------------|---|
| asadmin create-domain command accepting a new option named "-- profile" | Committed (Committed) | --profile accepts any arbitrary v that denotes the name of the fold Predefined values: "developer", "cluster", "enterprise", each spe a well-defined profile. |
| location of folder specifying profile | Committed (N/A) | This is the place where profile artifacts are searched for. This install-folder/lib/install/templa |
| The default domain.xml.template | Committed (N/A) | A template that generates the def configuration. This should be use the base XML template that is transformed by the style-sheets. |
| A properties file named profile.properties found in profile folder | Committed (N/A) | Simple property file with one nam pair per line. The name-value pai '='-delimited. Following properti exposed interfaces as well: * domain.xml.style-sheets -- a co separated list of XML style-sheet * domain.xml.token-values -- a co separated list of name-value pai can be additionally. The format c token in the documents is of the %%NAME%%. |
| A file of defaults for command line interface, asadmin | Committed (N/A) | Following is true about this file * Name: asadminenv.conf * Location: install-dir/config. * Format: simple property file, c allowed, if line starts with a '#' Names of properties start with "AS_ADMIN_". Users can edit this file for addi options that they commonly use, e AS_ADMIN_USER=admin. |
| A list of predefined tokens and their values | Committed (N/A) | These are predefined tokens that would always want to be replaced their documents. Examples of such and their values: * HTTP_PORT -- user specified HT * SERVER_ID -- name of the defaul |

```
virtual server ("server").  
* ADMIN_PORT -- user specified ac  
port.
```

4.5.2 Imported interfaces

| Interface | Stability | Exporting Project | Comments |
|-------------|-----------|-------------------|-------------------------------|
| Java SE 5.0 | Stable | JDK (open source) | This is a project-dependency. |

4.6. Doc Impact

Admin Guide, primarily, which explains how to create custom configurations, man pages.

4.7. Admin/Config Impact

This projects defines default configurations for three basic profiles that are bundled along with the product.

4.8. HA Impact

There are no new requirements.

4.9. I18N/L10N Impact

None.

4.10. Packaging & Delivery

Bundled "developer" profile: Optimized for developers

SUNWasu/lib/install/templates/developer

SUNWasu/lib/install/templates/developer/domain.xml (predefined)

SUNWasu/lib/install/templates/developer/domain.xml.xsl (for customization)

SUNWasu/lib/install/templates/developer/profile.properties (for customization)

Bundled "cluster" profile: Provides clustering capability, uses in-memory session persistence and security.

SUNWasu/lib/install/templates/cluster

SUNWasu/lib/install/templates/cluster/domain.xml (predefined)

SUNWasu/lib/install/templates/cluster/domain.xml.xsl (for customization)

SUNWasu/lib/install/templates/cluster/profile.properties (for customization)

Bundled "enterprise" profile: Provides clustering capability and preserves backward compatibility availability solution (HADB) and uses NSS for security.

SUNWasu/lib/install/templates/enterprise

SUNWasu/lib/install/templates/enterprise/domain.xml (predefined)

SUNWasu/lib/install/templates/enterprise/domain.xml.xsl (for customization)

SUNWasu/lib/install/templates/enterprise/profile.properties (for customization)

Bundled example for a custom profile: Explains the custom domain creation process.

SUNWasu/lib/install/templates/custom

SUNWasu/lib/install/templates/custom/domain.xml (predefined)

SUNWasu/lib/install/templates/custom/domain.xml.xsl (for customization)

SUNWasu/lib/install/templates/custom/profile.properties (for customization)

SUNWasu/lib/install/templates/custom/README

4.11. Security Impact

None anticipated.

4.12. Compatibility Impact

None anticipated. Care has been taken to preserve the previous distributions.

4.13. Dependencies

None.
