

One Pager: Usage Profile Support for Application Server

Table of Contents (Status: Draft, Ready for Review, **Reviewed**, ARC'ed)

[1. Introduction](#)

[1.1 Project/Component Working Name](#)

[1.2 Name\(s\) and e-mail address of Document Author\(s\)/Supplier](#)

[1.3. Date of This Document](#)

[2. Project Summary](#)

[2.1 Project Description](#)

[2.2 Risks and Assumptions](#)

[3. Problem Summary](#)

[3.1 Problem Area](#)

[3.2 Justification](#)

[4. Technical Description](#)

[4.1 Details](#)

[4.2 Bugs/RFE's](#)

[4.3 Scope](#)

[4.4 Out-of-scope](#)

[4.5 Interfaces](#)

[4.6 Documentation Impact](#)

[4.7 Configuration/administration Impact](#)

[4.8 High Availability Impact](#)

[4.9 Internationalization](#)

[4.10 Packaging](#)

[4.11 Security Impact](#)

[4.12 Compatibility](#)

[4.12 Dependencies](#)

[5. References](#)

[6. Schedule](#)

[7. Document History](#)

1. Introduction

1.1. Project/Component Working Name

GlassFish V2/Sun Java System Application Server 9.1.

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

09/11/2006.

2. Project Summary

This project discusses the support for application server usage profiles (*profiles*, for short). It identifies how a particular kind of users use the application server software. Once a profile support for the same planned for this release will take care of relevant *out-of-the box* configuration server runtime such that its user-experience is in sync with the expectation.

2.1. Project Description

Application Server Software is used in a few significantly different ways for significantly different purposes. A *developer* might want to try out the latest technology advancements in and out of the Java EE in a relatively painless manner. In such cases, the general security requirements are not that strict on the server however, can make significant difference to the user-experience. A *deployer* on the other hand, to deploy the state-of-the-art Java EE applications in a production environment that has to be secure. Although the same software can be *manually tuned* to specific requirements using administrative *the-box configuration* (that reasonably meets user expectations) that matters and that is the

Traditionally, this kind of tweaking of the configuration of the runtime was achieved through multiple *editions* of the software. For example, the Platform Edition of Sun's Application Server was for developers and Enterprise Edition was geared towards large enterprises. However, this is not the way to realize the aim to use the software in a particular manner and has following limitations:

- It creates unnecessary barricades in users' perception of capabilities of the application. The problem worsens when virtually all of the source-code is *open*. It becomes complex to manage *distributions* in that case and then limit a particular distribution of the open-source project that you can do with it.

- It unnecessarily associates application server *binaries* and configurable domain *runtime* manner. For example, all the distinct server runtimes (domains) created in a Platform functionally *equivalent*. In other words, in a given distribution of application server configure two runtimes, one of which is tuned for developers and other for deployed demarcation that is not sustainable for long.

What is needed is a single distribution of application server software (that fully implements is capable of behaving according to its expected primary usage, based on its configuration to be a step in that direction. In this release, the foundations for this will be laid. **The exact deliverables of this project with server software distributions is rather unspecified at** for this is the backward compatibility requirements of application server and various environments (Operating System, Java Enterprise System etc.) in which it integrates. It might then be the is available on a particular distribution of application server, but it would make sense to *no* far as possible.

It is important to distinguish the intended scope of profiles in GlassFish V2 and their (that extensions. Going ahead, it is required that application server runtime is configured to a particular *rather than a particular kind of users*. This is expected to optimize the performance (startup etc.) of application server while hosting applications of a given kind. An example would be where the users intend to use the runtime to develop/deploy only the web applications. To demand usage of application server, a more modular structure of application server components clearly *out-of-scope* for this release (but it is in line with the general direction we seem to be taking) points of view should work in concert and should not interfere with each other.

A key enabler of this feature is the careful distinction between binaries and runtime configurations. Application server has been promoting, historically. When the bits are put on the disk, what is needed is of an administrative domain which optionally provides a *pre-configured runtime* for the use of administrators are *already* able to exploit the same set of bits to suit their specific needs. This is a sound foundation.

2.2. Risks and Assumptions

There are no major technical risks with this project. A moderate risk is with respect to making application server implemented as *life cycle modules*, configurable. These are pluggable components of application server which are very closely related with the notion of profiles. In theory, a particular life cycle module should be made pluggable and should be enabled/disabled depending upon its association with respective profiles. However, this is a stretch goal for this release.

A moderate risk (non-technical) is with respect to the distribution of application server software. A loose coupling is desired between usage profiles and application server editions, an unspecified configuration might result in confusion and might result in lack of interest. A detailed discussion of various bundles and their dependencies is however beyond the scope of this document. A recommendation is made below in this regard.

For GlassFish V2, a set of *predefined profiles* will be provided. There is no provision of a release.

3. Problem Summary

3.1. Problem Area

Several questions have been asked and suggestions have been given to improve the development of GlassFish. This specification is a vehicle to accumulate all such suggestions and formalize them. The basic problem that is being attempted to solve is how to improve the response of the application to a specific use.

3.2. Justification

This is the best time to offer such a feature, mainly to drive the GlassFish adoption efforts. When the GlassFish platform is available, it makes sense to provide *usage-profile support* for

4. Technical Description

4.1. Details

4.1.1 Presenting Profiles to Administrators/Developers

The proposed manner in which a user comes across the profiles is using the `asadmin` command with a `--profile` option. An example invocation of this is as follows:

```
asadmin create-domain --user admin --adminport 4848 --profile dev-domain
```

The domain here symbolizes an application server runtime that is suitable for both server and client applications.

The valid values that the option `profile` accepts are: `developer`, `cluster` and `enterprise`. These values are case insensitive. What follows describes in detail the *contract* of this command.

- The value *developer* implies that the domain should be configured (as *best* as it could be) for a development environment. The value *cluster* implies a user should be able to create multiple server instances. The value *enterprise* implies that a user intends to use the given domain for large-scale deployments. The details of these configurations that *define* the domain are given in [4.1.2](#) below.
- Actually, there is no need for a separate *cluster* profile because it is logical for devel

clusters incurring only marginal overhead. However, it is the current state of implement a separate profile which mostly resembles the *developer* profile, except that the cluster instances could be created. The main reason that we are introducing this profile is to the server-side Pluggable FeatureFactory implementation for GlassFish V2 release. two profiles are expected to merge. That will happen when additional capabilities in *life cycle modules*) to support cluster deployments can be easily turned off if needed the *developer* profile is a degenerate case of the *cluster* profile.

In addition to the above, two *classic* profiles are introduced. It might be possible to remove ahead. The need for classic profiles arises because we are introducing profiles for the first [section on installers](#) for details.

4.1.2 Comparison of Various Profiles w.r.t. Configuration Parameters

The two tables (1, 2) below describe the gist of this project. The [first table](#) gives a glimpse a domain for a given profile should look like, whereas the [second table](#) provides the ration configuration parameter.

It is important to note that even if most of domain's configuration is stored in the well-known domain.xml (here is the [schema](#)), there are other configuration files that will have to change profiles.

| Configuration/Other Parameter | Value for Developer Profile | Value for Cluster Profile | V |
|--------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| Security Store | JKS | JKS | N |
| Quick Startup | true (enabled) | false (disabled) | fa |
| Java Platform Security Manager | false (disabled) | false (disabled) | tru |
| JVM | <ul style="list-style-type: none"> • Hotspot Client VM • -Xmx= • -Xms= • | <ul style="list-style-type: none"> • Hotspot Client VM • -Xmx= • -Xms= • | |
| Server Pluggable Feature Factory | PE | EE | E] |
| Default Static Ports | <ul style="list-style-type: none"> • HTTP (8080) • HTTPS(8181) • IIOP (3700) • IIOP/S(3820) | <ul style="list-style-type: none"> • HTTP (8080) • HTTPS(8181) • IIOP (3700) • IIOP/S(3820) | |

(All Profiles get a IPDA

| | | | |
|----------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------|
| Debugger Port(9009/disabled by default)) | <ul style="list-style-type: none"> • IIOP/S/MA(3920) • JMX (8686) • Admin/HTTP(4848) • JMS Broker (7676) | <ul style="list-style-type: none"> • IIOP/S/MA(3920) • JMX (8686) • Admin/HTTPS(4848) • JMS Broker (7676) | |
| Admin URL | http://host:admin-port | http://host:admin-port | ht |
| HTTP Access Logging | false (disabled) | false (disabled) | tru |
| Heart-beat (GMS) | false (disabled) | false (disabled) true (enabled) | tru |
| JMS Implementation Type | EMBEDDED | EMBEDDED for DAS, LOCAL for instances | EM L |
| Default Startup of Domain (Background/Foreground) | Foreground Background | Foreground Background | B F |
| Ability to Create Cluster/Standalone Java EE Engines | Not available | Available | A |
| Session Replication Mechanism | Not available | Available, in memory. | A |

Table 1: Distinction within Profiles

| Parameter | Few details about what it means | Reasons to make <i>profile-defining</i> |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Security Store | Server uses at least a key-store and trust-store each to deal with its certificates and keys. There are two main types of stores: JKS and NSS . These differ in terms of formats and tools that know how to configure the stores. | A particular orga requirements are have processes to configure their security store (e.g. Java EE) or security store configuration. |
| | | Traditionally, the Startup gives an |

| | | |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------|
| Quick Startup | This is an Java-NIO based implementation that is part of on-demand services <i>framework</i> . | that perceived state than its actual state Traditionally, the startup of admin in the non-administered <i>enterprise</i> domain way. |
| Java Platform Security Manager | A security manager controls the privileges of a particular piece of Java code in accordance with an administrator-defined security policy. | Developers are not <i>CodeSources</i> and the security manager developers are liable behavior of their (seemingly) unneeded platform security all over). |
| JVM | By default, we use Sun's JVM's. As such, it is obvious to leverage the optimizations available in the JVM itself, especially the Hotspot VM. | JVM configuration significantly different certain situations VM parameters (making users read documentation). |
| Server Pluggable Feature Factory | | |
| Default Static Ports | | |
| Admin URL | | |
| HTTP Access Logging | | |
| Heartbeat enablement for a Cluster (GMs) | | |
| JMS Implementation (SJS MQ Broker) Type | | |
| Default Startup of Domain (Background/Foreground) | As of now, the only way to start a domain is <code>asadmin start-domain</code> . By default, the domain is started in the background letting the user know the location of the file where the server log is redirected. See a dependency here. | Competing products have this capability to see the server rather than in a file In an enterprise setting can only be started such it is always |

| | | |
|------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Ability to Create Cluster/Standalone Java EE Engines | Determines whether there is a runtime support to create additional clustered or not Java EE Engines. | A pivotal difference are cluster-aware |
| Session Replication Mechanism | A mechanism to replicate the <i>session state</i> across various <i>clustered</i> server instances to provide reasonable availability. | It is needed that applications are 1 TBD: Need more |

Table 2: Why a particular parameter is a *profile-defining-parameter*

4.1.3 Distinction Among Profiles

This specification does *not* define any specific rules regarding defining a profile. It (defines) the outcome of *need*. There are various items to consider while defining a profile and hence it is subjective around this. In general, it is subjective whether to create a new profile or to make a new profile a part of an existing profile.

This brings up an interesting point: Why have 2 profiles like `developer` and `enterprise` and `cluster`? Can these be subsumed under one profile?

A reasonable explanation can be provided along the following lines:

- At the center of it, a profile is almost completely defined by a set of configuration parameters *defining parameters* of a profile. Two different profiles have different values for *defining parameters*.
- A profile should result in optimized performance, expected user experience *right of the moment* creation configuration changes to achieve its promise. A few changes here and there can change a profile. Thus a profile must represent a *class of users*.

Food for thought:

Do developers Need to Create Application Server Clusters? (Do we need yet another runtime support for clusters)?

4.1.4 Profiles and Administrative Clients

Profiles is a server-side phenomenon. It is not expected that admin *clients* know the profiles and behave differently. Here is how various admin clients will be impacted by profiles:

- `asadmin` client: The *asadmin* client is designed in such a way that it can help an admin domain runtimes (with unknown usage profiles) from a single shell. In other words, it is *agnostic* and as such, it cannot (and should not) present a command set that is applicable to all profiles.

profile. For example, if there is asadmin is directed to a domain with developer profile asadmin to hide a command like `create-cluster`. The command set is thus no

- AMX client: AMX is the programmatic API. We have exactly one API that caters for server administration. This is true for previous releases. Thus, by definition, AMX is For this release, as said elsewhere, there will be no AMX support to realize a particular be used after the domain has been created and started.
- Admin GUI client: This is implemented as a web application deployed to a particular administered. The current plan is to query the administrative MBeans about certain differences when certain support is not present in the MBeans. This, however is independent profiles.

4.1.5 Profiles and Application Server Installers and GlassFish Bundles

As of now, we have following basic distribution mechanisms (**this excludes the Java EE variations thereof. The Java EE SDK distributions are based on PE distribution**).

Following has been decided at the architecture forum in this regard:

- PE user and Java EE SDK user should be able to create clusters. In other words, in available.

| Id | Bundle/Distribution | Comments | Default Profile |
|----|------------------------------------------|---------------------------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 1 | Standalone Application Server PE Bundle. | Base Application Server. | developer |
| 2 | Standalone Application Server EE Bundle. | Base Application Server + Enterprise capable portion + Load balancer plugin + HADB software. | classic-ee enterprise |
| 3 | GlassFish Download Zip | Similar to 1, with single XML (setup.xml) for configuration. | developer (User has to run ant setup.xml). |
| 4 | GlassFish Download Zip | Similar to 3 with additional support for clusters/instances. Single XML called setup-cluster.xml for configuration. | cluster (User has to run ant setup-cluster.xml). |

| | | | |
|---|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|
| 5 | Platform Specific Packages and Java ES Installer. | These is a mode where the application server software is distributed in terms of defined set of packages and post installation scripts. The other software that integrates into Java ES is leveraged, rather than bundling it ourselves unlike 2 . | enterprise |
|---|---------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------|

To take care of these differences in the way we bundle the application server software, the [asaenv.conf\[bat\]](#) has been made. Ideally, a pluggable architecture that lets users download is a model we should embrace.

One of the goals for GlassFish V2 is provision of one bundle that has capabilities of cluster will be one bundle by merging bundles [3](#) and [4](#). The setup.xml will set up the developer profile, cluster.xml will set up the cluster profile.

Following should be noted in this regard:

- Only a single set of templates will be used for all types of installations. These templates are located in `dir/lib/install/templates/ee`. The portion "ee" in the name of this path is restored to match with various environments in which application server is integrated. This should not mean that a particular jar (referred to in a particular template) is not available in given distribution. If a particular functionality will not be available.
- In order to retain the user expectation of domain creation process, two additional profiles which will be the default profiles for respective installations. These are *classic-pe* and *classic-ee* which will be the default profile for PE installation, whereas *classic-ee* will be the default profile for EE installation.
- The values for various [profile-defining parameters](#) in the case of *classic-pe* and *classic-ee* will be the same because they are defined by the classic templates that are available in PE and EE installers.
- ~~It is desirable that the standalone installers for PE and EE provide a choice to the user to select the profile for the default domain. Note that the default domain is created by the installer. Thus, users installing the product get to choose the profile for the default domain. It is expected that the default profiles for each installer should be. Thus,~~
 - ~~PE installer has a choice among developer, cluster and classic-pe.~~
 - ~~EE installer has a choice among developer, cluster, enterprise and classic-ee.~~

4.1.6 Profiles: Noteworthy

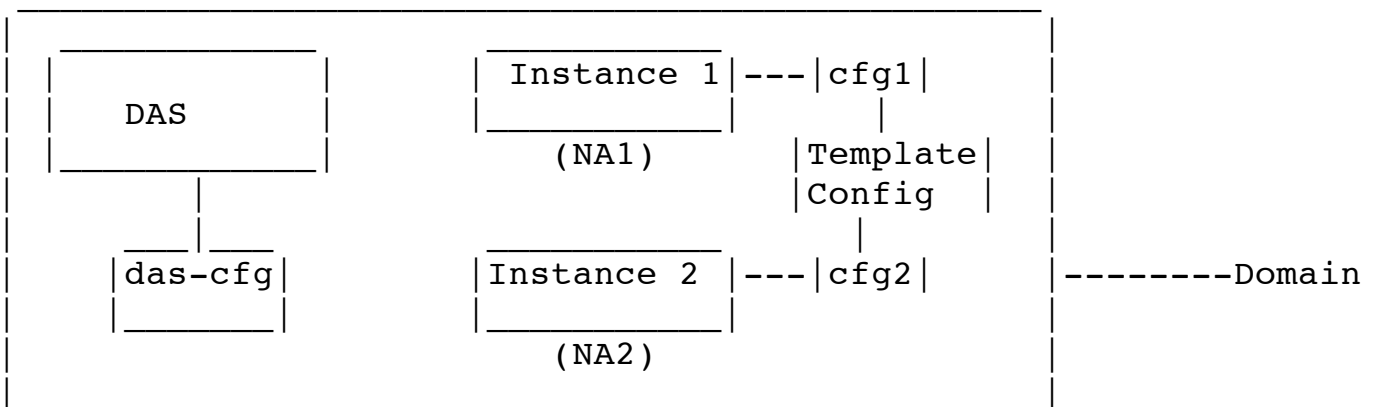
Here is something that you should know, about this implementation of profiles:

- A running domain does not know anything about its own profile. For example, a dc

asadmin create-domain --profile enterprise ... does *not* runtime about specifics of this profile. Coupling it loosely like this has several benefits making migration within profiles easy. If a user has forgotten about what profile a p and Admin GUI will have ways to identify possible profile of a domain but it is *not* domain. The shortcoming of doing it this way is that the startup of the server cannot behavior, since configuration does not have an explicit indication of the profile of the ~~[TBD: Discussion Required]~~.

- In general, a user is supposed to *know* the profile of a domain. A profile is a user's v configuration.
- It is quite possible that user reconfigures an existing domain to behave in a profile d profile. From this standpoint, a profile is a fluid property of a domain.
- An important outcome of this discussion is that the so-called ClientPluggableFeature The way the asadmin client deals with the domain creation will be determined comp

4.1.7 More about Enterprise and Cluster Profiles



The enterprise architecture (schematic) of the application server is shown [above](#). **DAS** is t that manages other Java EE Engines (in that domain) that host the user applications. As an additional costs that are arguably unnecessary) DAS is a fully functional, compliant Java E can choose (though they rarely, if at all, do so) to deploy applications to the DAS The esse enterprise profile is the provision of standalone and clustered server instances. As shown i has a *configuration* (shared or not) that is based on a *Template Configuration*. Note that T applies not only to the existing server instances, but also to the instances that might be crea its own configuration (shown as *das-cfg*) that is independent of the Template Configuratio

DAS:das-cfg

Instances:Template Configuration

It is quite important then to know what the profile-specific configurations apply to -- DAS

Configuration. It applies to **both** unless specified otherwise. A significant outcome of this

- When an administrator chooses to create a domain with Cluster/Enterprise Profile, a instances is also reasonably determined. This further reduces the administrative over those server instances when enterprise deployments occur.

4.2. Bug/RFE Number(s)

These are indicated throughout the document where applicable. This feature however, is a discussion threads on GlassFish Discussion Forum.

4.3. In Scope

All that has been said about realizing usage profiles in Section [4.1](#) is in scope for this project. The composition of any defined profile is based on experiential knowledge. It is a learning process to learn *over time* that a particular configuration setting is suitable for a particular usage.

4.4. Out of Scope

Since this entire topic is of general interest and naturally has various interpretations, it is in this project is *not*. Following things are not in scope for this project, at least for this release

- There is no programmatic (AMX) configuration support or Admin GUI (Console) support for this release. The only means to manifest profiles shall remain the `asadmin create-domain`.
- There is no migration planned for an existing domain that's created (or not) with a profile. It is planned that the profile of a domain can be changed after it is created. Thus, this feature conflict with *the upgrade story for application server*. It is to be noted however that we will provide a reasonable upgrade path, in following releases.
- Although the concept of profiles can easily be extended to take care of custom needs, we will not provide such a support for this release. GlassFish V2 will only provide a profile for profiles integrated into the product.

4.5. Interfaces

- <http://www.opensolaris.org/os/community/arc/policies/interface-taxonomy/> describe interface taxonomy.

4.5.1 Exported Interfaces

| Interface | Stability | Former Stability (if | Comments |
|-----------|-----------|----------------------|----------|
|-----------|-----------|----------------------|----------|

| | | changing) | |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>asadmin create-domain [--profile (enterprise cluster developer)]</p> <p>A new option named profile that takes one of the following values: "developer", "cluster", "enterprise".</p> | EVOLVING | NA | Exact changes that configuration for a change without not |
| <p>A text-file named: asadminenv.conf on all platforms</p> | EVOLVING | NA | <p>This file is used to care of the interplay application server in script will <i>use this f</i> is expected that use as it affects the insta</p> <p>Note that the conten to define the default asadmin commands</p> |
| <p>The default location of asadminenv.conf</p> | EVOLVING | NA | <p>Standalone Installer</p> <p>[install-dir]/config</p> <p>Java ES Installers (/opt/SUNWappserv</p> |
| <p>The format of asadminenv.conf</p> | EVOLVING | NA | <p>The format is that o is supposed to cont: '=' separated name- beginning with '#' i</p> |
| | | | <p>These can be enhar For this release,</p> <hr/> <p>A Vanilla PE Sola</p> <p># These are install- AS_ADMIN_POR AS_ADMIN_PRO AS_ADMIN_SEC</p> <hr/> <p>A Vanilla EE Sola</p> |

| | | | |
|-----------------------------------------------------|-----------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>The contents of <code>asadminenv.conf</code></p> | <p>EVOLVING</p> | <p>NA</p> | <pre># These are install- AS_ADMIN_POR AS_ADMIN_PRO AS_ADMIN_SEC</pre> <hr/> <p>A Vanilla GlassFish</p> <pre># These are install- AS_ADMIN_POR AS_ADMIN_PRO AS_ADMIN_SEC</pre> <p>Note that the default users would get is domain.</p> <p>Respective installer this file appropriate.</p> |
|-----------------------------------------------------|-----------------|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

4.5.2 Imported interfaces

| Interface | Stability | Exporting Project: Name, Specification or other Link. | Comments |
|-----------|-----------|-------------------------------------------------------|----------|
| | | | |

4.5.3 Other interfaces (Optional)

| Interface | Stability | Exporting Project: Name, Specification or other Link. | Comments |
|-----------|-----------|-------------------------------------------------------|----------|
| | | | |

4.6. Doc Impact

There will be considerable impact of this project on documentation. The following will be

- Admin Guide.
- asadmin create-domain manpage.

4.7. Admin/Config Impact

This has been dealt with elsewhere ([1](#), [2](#)). If this [dependency](#) is resolved to have an indication configuration, the configuration schema might change (although we can leverage a <prop to accommodate that.

4.8. HA Impact

Not applicable.

4.9. I18N/L10N Impact

No considerable impact above and beyond product impact.

4.10. Packaging & Delivery

Here is the packaging impact:

| Package | Changes | Comments |
|-------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|
| SUNWasu sun-asu-9.1.rpm SUNWasu.zip | \$installdir/lib/install/templates/developer/domain.xml \$installdir/lib/install/templates/cluster/domain.xml \$installdir/lib/install/templates/enterprise/domain.xml | These used doma |
| | | |

For installation impact, see [here \(section 4.1.5\)](#).

4.11. Security Impact

Not applicable.

4.12. Compatibility Impact

// Incompatible changes to interfaces that others expect
// to be stable may cause other parts of application server or
// other dependent products to break.

// Discuss changes to the imported or exported interfaces.

// Describe how an older version of the interface would

// be handled.

// List any requirements on upgrade tool and migration tool.

4.13. Dependencies

No dependencies on other projects. Some subtle dependencies with respect to *positioning*.

5. Reference Documents

Refer to Product Documentation for Sun's Application Servers, especially the admin guide

6. Schedule

6.1. Projected Availability

Fully functional implementation by Milestone 3 - GlassFish V2. Refer to overall [milestone](#)

7. Document History

| Version | Date | Author, Comment |
|---------|-------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1.00 | 06 January 2007 | Kedar Mhaswade. Removed classic profiles, cleaned asadminenv.conf related entries. Also, updated GlassFish distribution information. Some changes to fit the current implementation direction. |
| 0.97 | 01 October 2006 | Kedar Mhaswade. Added classic profiles, clarified templates. |
| 0.95 | 14 September 2006 | Kedar Mhaswade. Added various material for profile-defining parameters, installers, asadminenv.conf interface. |
| 0.8 | 11 September 2006 | Kedar Mhaswade, created. |