

Functional Specification for Packager & Installer

Author(s): Naman Mehta (naman.mehta@sun.com)

Version: 1.0

1 Introduction

This document provides details on

- a. The packaging structure of the SailFin binaries.
- b. The installation options for installing these SailFin binaries.

1.1 Feature in Packager

Adding new packages for SailFin.

- Add new packages specific to SailFin. Currently there is no planned support for native packages, and hence these packages would be zip based packages, which would not be exposed to the user.
- The existing zip based packages that are created by GlassFish would be picked up by the SailFin packager module and supplied to the installer.

1.2 Features in Installer

Generating different types of installer to setup SailFin.

- Extend existing GUI/CLI installer in GlassFish to install SailFin bundles which would also include the underlying GlassFish bundles.
- Ant based Jar installer.

1.3 Features not supported in current release

- We are not supporting native packages for Solaris (SVR4) and Linux (RPMs). Given that there is no support for native packages.
- Develop a new "overlay installer" which would install SailFin binaries alone on an existing version of GlassFish. This "overlay installer" would be based upon the add-ons installer.

2 Design Overview

2.1 Packager

1. List Of Packages to be delivered

The following packages would be delivered as a part of the SGCS. These packages are planned to be in the .zip packaging format. If there is a need, it would be possible to generate these packages in the native packaging format too, such as RPM/SVR4.

1. *SUNWssa*
2. *SUNWcas*

- Includes all existing GlassFish packages from packager module.

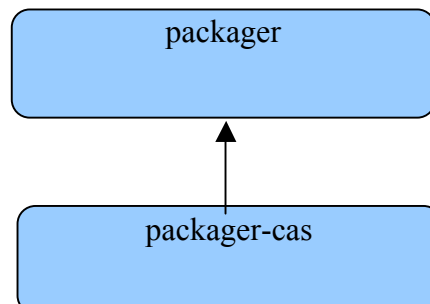
2. Details of Packages to be delivered

Name	Description	Installation Directory	Comments
SUNWssa	Jars to be included: ssa-api.jar	<asinstall_home>/lib	
SUNWcas	Jars to be included: comms-appserv-rt.jar comms-appserv-api.jar comms-appserv-admin-gui.jar comms-appserv-admin-cli.jar	<asinstall_home>/lib	comms-appserv-rt.jar includes all classes form below set of jars: comms-appserv-admin.jar comms-appserv-common.jar comms-appserv-deployment.jar comms-appserv-integration.jar comms-appserv-security.jar sip-stack.jar clb.jar replication.jar http.jar rcm.jar framework.jar http-utils.jar comms-appserv-api.jar includes all API classes form below set of jars: sip-stack.jar comms-appserv-admin.jar
	Samples directory which contains SailFin samples.	<asinstall_home>/samples	

2.2 Design of Packager in SailFin

A new module called “packager-cas” would be created alongside the existing packager directory.

During build, packager-cas module first makes call to packager and build all GlassFish packages and then builds SailFin packages. Detailed design added in Appendix Section 10.



2.3 *Installer*

2.3.1 GUI/CLI based installer

- This installer would be available as a part of the SGCS 1.0 distribution alone.
- This GUI/CLI Based installer would be an extension of the GUI/CLI Based Installer available for SJSAS. This installer would be used to install SGCS as a single bundle. In other words, this installer would install a SJSAS 9.1.1 and then install SGCS binaries in the same location. The whole operation would be a single operation for the user and the installation of GlassFish and SailFin binaries would be transparent to the user.
- This installation would bundle JDK 1.6 which is going to be installed as part of installation.
- This installer creates an installation directory for SGCS (as specified by the user) and install the binaries for SJSAS and SGCS. The default installation directory for SGCS would be the following

OS	Installation Directory
Solaris	/opt/SUNWappserver
Linux	/opt/sun/appserver
Windows	<Root>/Sun/Appserver

- This installer would perform following additional operations specific to the SGCS installation.
 1. Interview user for additional details about the SIP Container like SIP Port and SIPS Port
 2. Type of profile for the domain, the valid values being either cluster or developer. The installer would default the profile to be a “cluster” profile.

SIP Configuration

Please configure the sip settings for the server.

SIP Port:

SIPS Port:

Select Profile to Setup:

With Cluster

Without Cluster

< Back Next > Cancel Help

- User interview would with th

e following details during Installation of GUI/CLI based Installer.

1. Details about the DAS configuration like admin user name, admin password, admin port, HTTP Port , HTTPS port. This interaction is similar to SJSAS.
2. Details about the SIP Configuration like SIP Port and SIPS Port. Type of profile for creation of the domain. This interaction is newly added as part of SailFin Installation as mentioned above.
3. Provides different Installation options like 'Upgrade from Previous Version' and 'Enable Update center client'. Default Enable Update center client would be selected. This interaction is similar to SJSAS.
4. Asks for the Product Registration. It would ask for user details for Sun on-line Account (SOA). If user doesn't have SOA id they can register at the same time. User have option to 'Skip Registration' also there. This interaction is similar to SJSAS.

2.3.2 Ant based Jar Installer

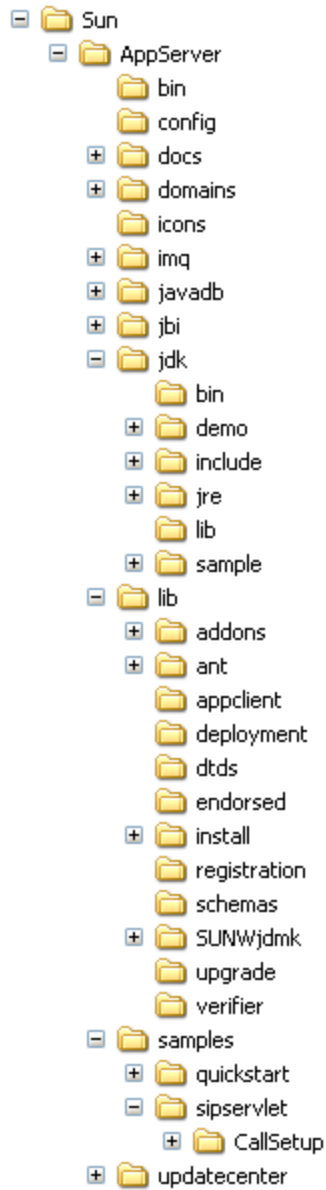
This installer is the same as the jar installer available with GlassFish. This would be the standard installer for SailFin distributions.

- The installer be available as jar file. The user would have to use the command `Java -Xmx256M -jar xxxx-installer.jar` to install the binaries.
- The installer would install the binaries of SailFin which would also include the underlying GlassFish v2.1 binaries.
- Two ant based build files would be provided to configure the domain.
 - Setup.xml would allow the users to specify the configuration for the domain with a developer profile and also create a domain.
 - Setup-cluster.xml would allow the users to specify the configuration for the domain with a cluster profile and also create a domain.

2.4 Layout

The SailFin(SGCS 1.0) directory layout would closely resemble the layout of GlassFish (SJSAS). The diagram below indicates the layout. The root directory would be `<install-root>/SailFin`.

- SailFin specific jars would be the part of `<install_home>/SailFin/lib` directory.
- SailFin Samples would be the part of `<install_home>/SailFin/samples/sipservlet` directory.



3 Performance

NA

4 Management

Imported Interfaces

Name	Specified in	Stability
Asadmin CLI commands	http://wiki.GlassFish.java.net/attach/GlassFishV2AdminRelatedOnePagers/cli-one-pager.html	Stable
SUNtls	WSARC/2002/366 NSS: Network Security Services	Stable

Name	Specified in	Stability
SUNtisu	WSARC/2002/511 NSS: Network Security Services Command Line Utilities	Stable
Update Center	http://wiki.GlassFish.java.net/attach/OnePagersOrFunctionalSpecs/update_center_onepager.txt	Stable

5 Packaging, Files, and Location

5.1 Details of Packages to be delivered from SailFin

This has been mentioned in detail in Section 2.1

6 Quality

Both the installers should be tested for :

- a. Installation of binaries, configuration of domains and also uninstallation of binaries.
- b. Aborting of the installation process should abort cleanly and allow for a re-install.

7 Documentation Requirements

An installation guide should be created for zip based installer. These guides should list down all the scenarios and have screen shots wherever necessary.

8 Open Issues

9 References

9.1 Details of Packages to be delivered from GlassFish.

<http://iecwweb.india/j2eexml/pub/Main/JES5IFRWork/SJSAs911-pkg-list.html>

10 Appendix

10.1 Details about packager-cas directory

For adding new packages on SailFin side added new directory under m/jws called 'packager-cas'. Directory structure for the same described as below.

- *m_jws*
 - *packager-cas*
 - *build.xml*
 - *maven.xml*
 - *package.properties*
 - *project.properties*
 - *project.xml*

- *package-cas.xml*
- *internal*
 - *install*
 - *bundled_pkg.txt*
 - *bundled_rpm.txt*
 - *copy_bundled_rpm.txt*
- *packages*
 - *bld_awk_pkginfo.ksh*
 - *common.properties*
 - *common.xml*
 - *mkepkg*
 - *makerpm*
 - *pkginfovars*
 - *common_files*
 - *SUNWcas*
 - *build.properties*
 - *build.xml*
 - *depend*
 - *pkginfo.tmpl*
 - *prototype_com*
 - *SUNWcas.spec.tmpl*
 - *SVR4PackageDefToFileList.xml*
 - *All other new packages for SailFin...*

- packager-cas module contains build script to build SailFin packages, which is called by installer during build or user can call 'maven package' to build SailFin packages.
- packages directory contains different SailFin specific packages like SUNWcas (as mention in section 2.2 above) and some common script to build each package which is called by packager-cas build script.
- During build, packager-cas module first makes call to packager and build all GlassFish packages lying under packager module and then makes call to packager-cas module to build SailFin packages.



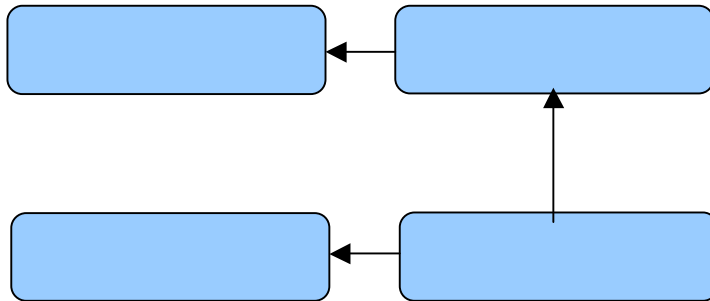


10.2 Details about installer-cas directory

For supporting new packages on SailFin side, planning to add new directory under m/jws 'installer-cas'. Directory structure for the same described as below.

- *m_jws*
 - *installer-cas*
 - *build.xml*
 - *dependencies.mk*
 - *Makefile*
 - *installer-common.mk*
 - *maven.xml*
 - *project.properties*
 - *project.xml*
 - *src/java/com/sun/enterprise/installer/windows*
 - *j2ee_app_lin.xml*
 - *j2ee_app_mac.xml*
 - *j2ee_app_sol.xml*
 - *j2ee_app_win.xml*
 - *j2ee_sdk_lin_nojdk.xml*
 - *j2ee_sdk_lin.xml*
 - *j2ee_sdk_sol_nojdk.xml*
 - *j2ee_sdk_sol.xml*
 - *j2ee_sdk_win_nojdk.xml*
 - *j2ee_sdk_win.xml*
- installer-cas module contains build script to build SailFin packages and GlassFish packages. When user calls build on installer-cas, it builds installer and then packager-cas module. installer module calls packager module to build GlassFish specific packages.
 - Each j2ee_*.xml file under installer-cas module contains packages need to be installed. It includes both GlassFish and SailFin packages.

- To execute the build user can make call 'maven bootstrap build create-self-extract' from installer-cas module.
- bootstrap from installer-cas calls to bootstrap in installer and installer makes call to bootstrap in packager module.
- build from installer-cas depends upon the dependencies.mk file. It makes call to installer module and then packager-cas module. During build it would copy j2ee_*.xml files to HOST_OBJ folder which contains which packages need to be add as part of SailFin build.
- create-self-extract from SailFin makes call to installer create-self-extract and which is going to create executable file based on the contents specified under j2ee_*.xml files.



10.3 Script based overlay installer

Note: This installer would not be part of this release.

- This installer would be available as a part of SailFin down loadable binaries.
- This installer would enable users to install the binaries that add the communications components of SailFin (as mentioned in Section 2.1 of this specification) on an existing installation of GlassFish. The version of the GlassFish installation on which this installer would install SailFin would SJSAS 9.1.1.
- This installer would be based on the add-ons installer available in GlassFish V2.
- This installer do the following operations:
 1. Check for the version of GlassFish at this location. If the version is less that the recommended version then it would abort. If not, then it would install the SailFin components mentioned in Section 2.1 on the installation of the GlassFish.
 2. If there is no domain that exists , the installer would create a domain with a developer profile.
 3. If there is a domain exists, the installer would update the domain with the configuration information needed for SailFin.