Functional Specification for SailFin Administration

Author(s): yamini@sun.com
Contributors: Irfan A, Vijay G
Version:1.2

Change Log

Version	Comments	Date	Author
0.1	Early draft	07/12/07	Yamini
0.2	Incorporated review comments from Prasad	07/20/07	Yamini
0.3	Updated sections 1.1, 2.1	07/24/07	Yamini
0.4	Incorporated Eltjo's feedback. Updated sections: 1.1, 2.1.3, 2.2, 3, 4.1.2, 4.1.3, 5, 8	08/08/07	Yamini
0.5	GUI section 2.1.2 updated	08/15/07	Irfan
0.6	Updated sections 2, 4.1.3, 4.1.4, 8	09/05/07	Yamini
0.7	Added Converged Load Balancer commands & configuration	09/19/07	Vijaya Gadhamsetty
0.8	Updated LB commands section, new security commands, session replication related DTD changes, monitoring pluggability	10/11/07	Yamini
0.9	Updated the option names in the convered-lb commands	10/17/07	Vijaya Gadhamsetty
1.0	Added security commands, updated DTD changes for security, stack layer configuration, described all pluggable features	12/15/07	Yamini
1.1	Update the GUI section with latest information.	12/17/07	Irfan
1.2	Incorporated comments from Binod	12/18/07	Yamini

Table of Contents

Introduction

<List proposed feature(s). Introduce the basic vocabulary. Why is this interesting? List capabilities that may be normally expected, but are not being supported. Are there any limitations and caveats that need to be disclosed?>

Session Initiation Protocol (SIP) is an application layer protocol for multimedia conferencing over IP that can be used to establish, maintain, and terminate calls between two or more end points. SIP Servlet API (see JSR 289) which enables SIP applications to be deployed and managed is based on the Servlet model. Combined with Java EE, SIP Servlets can be used to add rich media interactions to enterprise applications. Project SailFin adds SIP Servlets technology extension to GlassFish Application Server.

SailFin administration module extends GlassFish administrative capabilities to configure and manage the SIP container residing within single or multiple (cluster) instances of GlassFish Application Server. To allow for additional administrative features for SailFin, GlassFish administration infrastructure needs to be enhanced.

The focus of this document is on configuration tasks that are unique to SailFin like configuring the SIP container, managing SIP listeners or configuring SailFin load balancer. Configuration changes to the SIP container properties can be applied dynamically to a running server by using the administration console (GUI) or the command line utility *asadmin*.

1.1 Features

<List all requirements and features you are implementing. List those which may be normally expected to be implemented but are not.>

This document talks about providing administrative capabilities for Sun Java System Communication Application Server v1.0 (referred to as just SailFin henceforth) based on the following requirements.

1.1.1 Domain Configuration Enhancement (SWP0902)

Domain configuration needs to be enhanced to support SIP container configuration parameters. It must be possible to augment the administrative functionality of SailFin in a layered way, utilizing interfaces in the infrastructure.

1.1.2 Extending Administration GUI (SWP0904)

Based on the new configuration parameters, new UI screens may have to be introduced or existing screens will require modifications.

1.1.3 Extending Administration CLI (SWP0905)

It should be possible to plug-in new commands over the existing GlassFish commands. The CLI framework needs to be extended for same.

1.1.4 Extending the Event Framework (SWP0906)

The existing event framework of GlassFish does not make it easy to plug-in new event modules. This needs to be enhanced to support dynamic reconfiguration of SIP configuration parameters.

1.1.5 Monitoring (SWP0907)

Providing interfaces to plug-in SIP monitoring.

1.1.6 AMX Support

Expose AMX interfaces for all the new elements.

2 Design Overview

<Discuss the core concepts of the design. Provide diagrams. Show how this sub-system interacts with other features and sub-systems. You may write as much as you consider as useful for other developers to understand the basic design. Insert scenarios, use cases that will help the reader understand how the system will behave under various conditions. You can also look at this section as a map to navigate documented code! >

The design for administration as whole can be organized into 2 sections:

2.1 Extensions/Additions to GlassFish

The high level architecture is based on GlassFish administration infrastructure. The idea is to use and extend the GlassFish administrative capabilities to manage and support the SIP container.

2.1.1 CLI

2.1.1.1 Framework

The proposal is to use the extensibility feature of the existing CLI framework in GlassFish. The CLI framework stores all commands and options in a descriptor file called *CLIDescriptor.xml* The extensibility feature allows for loading of commands from different descriptor files. However, this framework has a limitation that though it can load multiple descriptors found in class path, the descriptor file name itself is fixed (default being *CLIDescriptor.xml*) It should be possible to provide different descriptor files. The base descriptor will have all the common/shared options defined, there is no need to define these options in new descriptor.

The new commands and options can be stored in a separate descriptor file, new command classes can be added if required. New command classes will be added in those cases where extra validation of options is required. All the serialized commands, descriptor file and man pages can be bundled as a separate jar and can be included in the class path of the CLI startup script '*asadmin'*

2.1.1.2 Commands

Interactive or automated configuration operations can be performed using the command line utility. New commands that will be introduced for SIP container management as well as the SailFin load balancer administration will be exposed as new MBeans in the administration server. New commands fall under following categories:

- λ SIP service/listener configuration
- λ Converged load balancer configuration
- λ Identity assertion trust configuration

See section 4.1.4 for new commands and options descriptions.

2.1.2 GUI

2.1.2.1 Description

The Sailfin GUI will enhance the current existing GlassFish administration GUI to take care of the new features and properties in Sailfin. The proposal is to enhance the GlassFish administration GUI to have certain hooks in the main components trees so that when it finds a SailFin specific administration module, the SailFin screens are displayed in the GUI.

2.1.2.2 GUI Pluggability

The GlassFish administration GUI uses a set of frameworks to render the various screens. The GUI uses the JSF Templating framework on top of Woodstock to display the various properties and pages for administering the GlassFish application server. The GUI uses a tree navigation system where a tree of nodes is displayed on the left pane and a corresponding page is displayed on the right pane.

The GUI also has two modes of display: one is the normal single configuration mode, called the **Developer Profile**, where only a single default configuration exists. The other is the **Cluster Profile** where the administration GUI can display and configure several configurations which can be then deployed to a domain.

There are therefore two component trees that are displayed depending on whether the administration GUI is running in a developer profile or a cluster profile. The two component trees will be modified to display the SailFin node components when a SailFin administration GUI specific JAR archive is found in the class path.

The Sailfin pluggability is achieved through a call to a special class which is initialized as a bean in the Glassfish code. The class have a function called isEnabled that is checked to see if the Sialfin components are available. Depending on isEnabled the Sailfin components will be displayed in the GUI.

Another important feature to note is the availability of the **<!include>** tag. This is available as of JSFTemplating 1.0 which is used in Sailfin. The tag allows the user to specify a file containing code to display certain nodes. The file can reside in any JAR file in the class path for the web application. We will be using the include tag to add code into the Glassfish repository.

2.1.2.3 GUI Navigation

There are several new features that will be available in SailFin and hence these will require new nodes in the GUI. The SailFin administration GUI will add the following nodes to the present GlassFish administration GUI. (Note that the bold components are the ones that are already present in the GlassFish administration GUI)

- **Applications**
  - SIP Modules
  - Converged SIP Modules
- **Configurations**
  - SIP Container
  - SIP Service
    - SIP Listeners
  - Converged Load Balancer
  - **Security**
    - Trust Configurations

2.1.2.3.1 Sip Modules and Converged SIP Modules

The SIP modules section will display pure SIP web applications as well as converged SIP web applications. A pure SIP web application is the one which only has a sip.xml and the corresponding SIP files. A converged SIP web application will contain a sip.xml as well as a web.xml along with other files. The normal extension for such applications would be SAR (SIP Archive).

Sailfin application server also has the ability to support a combined SIP and J2EE enterprise application archive. This would be an EAR application which contains a SAR module within it. Such modules will be shown within Enterprise Applications node of the Applications node in the administration GUI.

2.1.2.3.2 SIP Container

The SIP Container node will list various tabs such as Session Properties, Manager Properties, Store Properties, Layer Configurations and so on. There will also be a General tab. The general tab is used for general SIP container properties and attributes.

2.1.2.3.3 SIP Service and SIP Listeners

The SIP Service node will be used to configure the element sip-service and its child elements. The SIP service node will display tabs to configure the SIP service settings, access log settings for the configuration, Request processing, Keep Alive, Connection Pool and SIP Protocol.

There will be a single child node for the SIP service called the SIP Listeners. The node will display the SIP listeners that have been created for the configuration. The users can create new listeners, delete listeners and configure listener properties such as ports, IP address and SSL Properties.

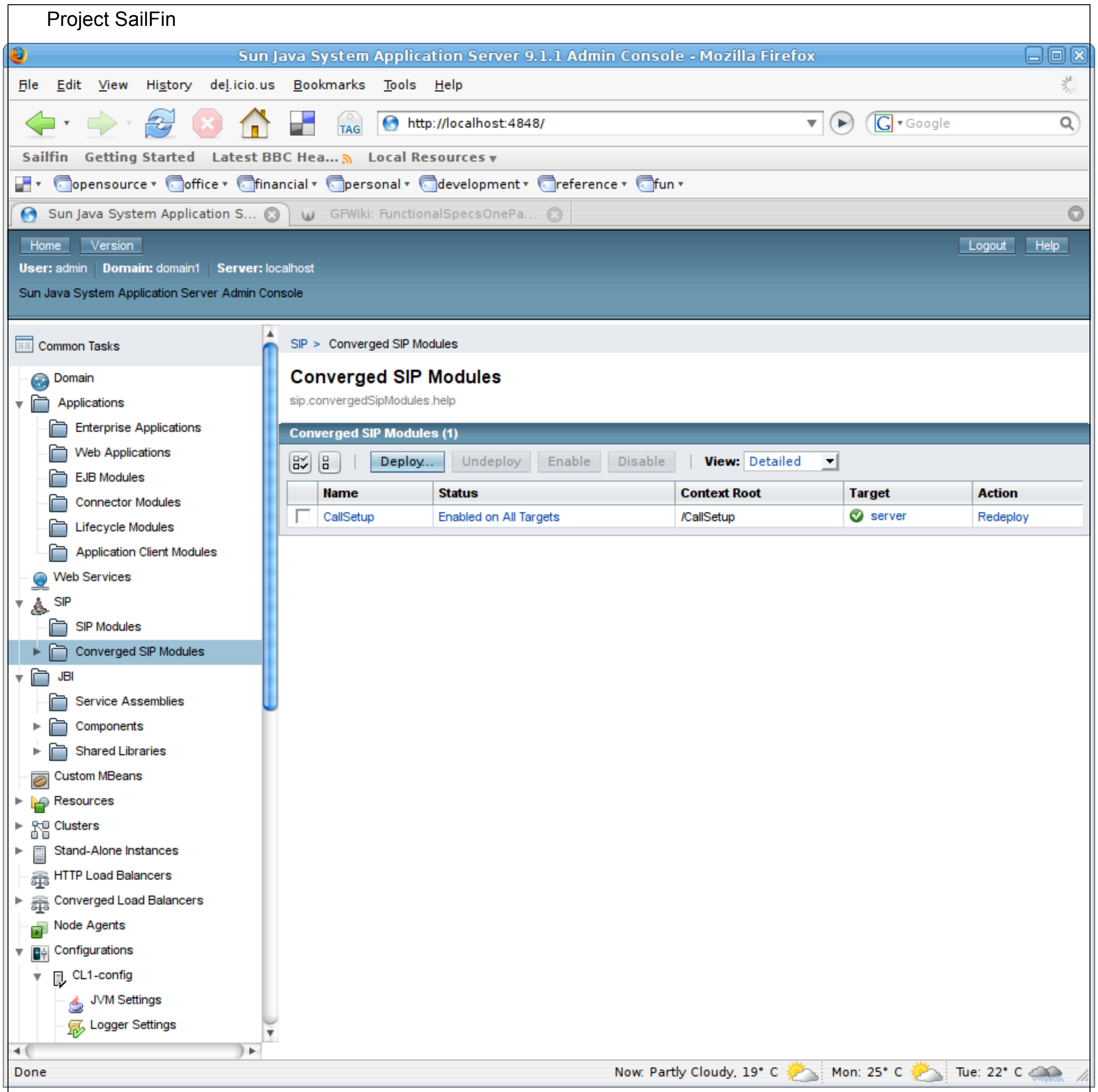
2.1.2.3.4 Converged Load Balancer (CLB)

The CLB is a new component in Sailfin. The CLB is used to configure and setup load balancing across clusters. The users can create a new CLB for a given target and configure settings for the CLB itself, the CLB configuration that is automatically created and the hosting targets as well as the load balancing targets. The HTTP Load balancer is not used in Sailfin and hence the node shall be hidden in the Sailfin administration GUI

2.1.2.3.5 Trust Configurations

The trust configurations are node for handling the identity-assertion-trust element which is new in the Sailfin DTD. The users can use the GUI to create a new trust configuration that can be used to trust hosts and domains or a specific user defined trust handler. The GUI can be used to list, create, and delete trust configurations as well as edit trust configurations.

The following pictures depict the new Sailfin administration GUI.



Sun Java System Application Server 9.1.1 Admin Console - Mozilla Firefox
File Edit View History del.icio.us Bookmarks Tools Help
http://localhost:4848/
Google
Sailfin Getting Started Latest BBC Hea... Local Resources
opensource office financial personal development reference fun
Sun Java System Application S...
GFWiki: FunctionalSpecsOnePa...
Home Version
Logout Help
Common Tasks
Domain
Applications
Enterprise Applications
Web Applications
EJB Modules
Connector Modules
Lifecycle Modules
Application Client Modules
Web Services
SIP
SIP Modules
Converged SIP Modules
JBI
Service Assemblies
Components
Shared Libraries
Custom MBeans
Resources
Clusters
Stand-Alone Instances
HTTP Load Balancers
Converged Load Balancers
Node Agents
Configurations
CL1-config
JVM Settings
Logger Settings
SIP > Converged SIP Modules
Deploy... Undeploy Enable Disable
Done
Now: Partly Cloudy, 19° C Mon: 25° C Tue: 22° C

User: admin | Domain: domain1 | Server: localhost

Sun Java System Application Server Admin Console

Converged SIP Modules

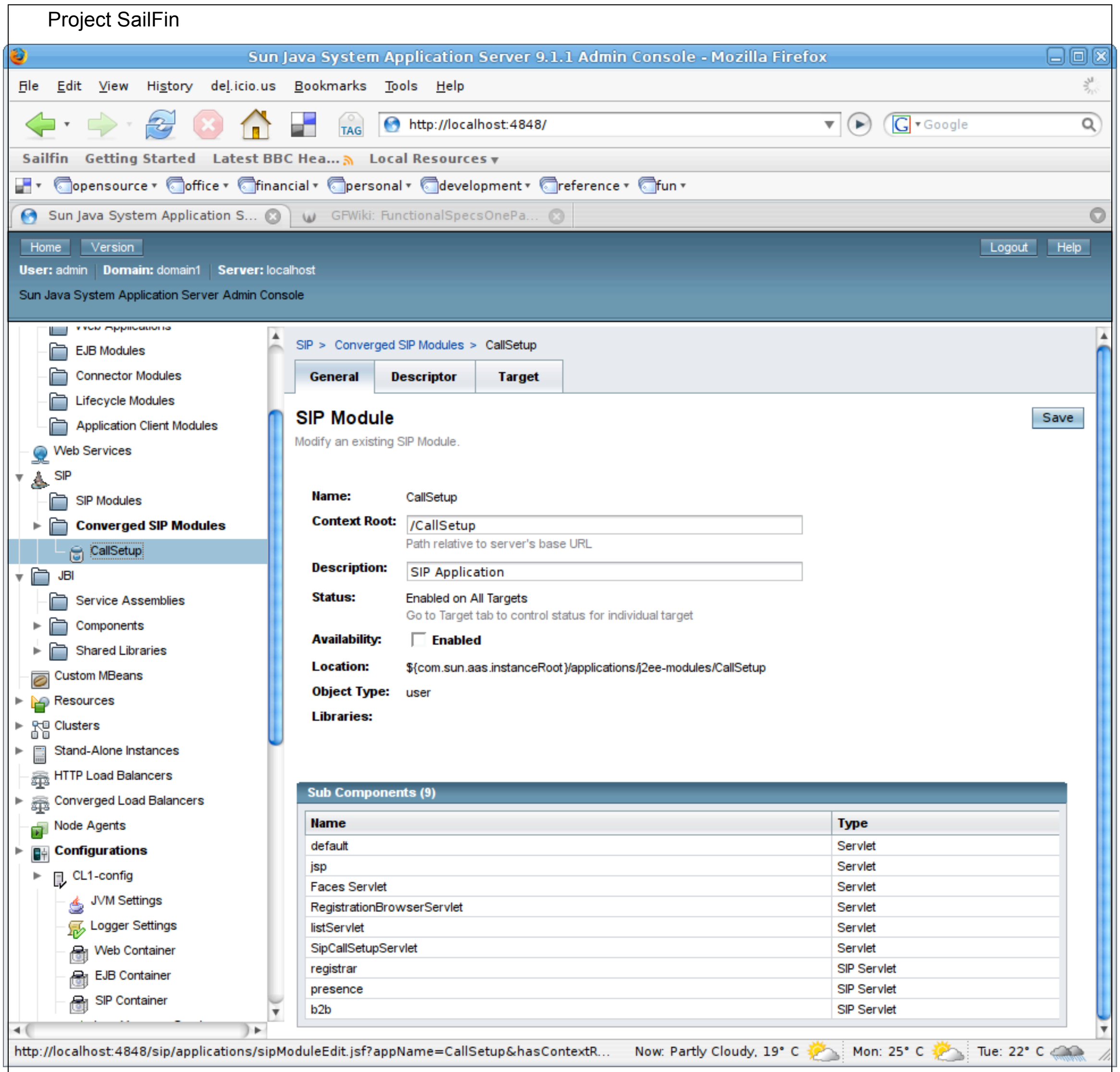
sip.convergedSipModules.help

Converged SIP Modules (1)

View: Detailed

	Name	Status	Context Root	Target	Action
□	CallSetup	Enabled on All Targets	/CallSetup	✓ server	Redeploy

Sip Modules and Converged SIP Modules


Sun Java System Application Server 9.1.1 Admin Console - Mozilla Firefox
File Edit View History del.icio.us Bookmarks Tools Help
http://localhost:4848/
Google
Sailfin Getting Started Latest BBC Hea... Local Resources
opensource office financial personal development reference fun
Sun Java System Application S...
GFWiki: FunctionalSpecsOnePa...
Home Version
Logout Help
User: admin Domain: domain1 Server: localhost
Sun Java System Application Server Admin Console
EJB Modules
Connector Modules
Lifecycle Modules
Application Client Modules
Web Services
SIP
SIP Modules
Converged SIP Modules
CallSetup
JBI
Service Assemblies
Components
Shared Libraries
Custom MBeans
Resources
Clusters
Stand-Alone Instances
HTTP Load Balancers
Converged Load Balancers
Node Agents
Configurations
CL1-config
JVM Settings
Logger Settings
Web Container
EJB Container
SIP Container
http://localhost:4848/sip/applications/sipModuleEdit.jsf?appName=CallSetup&hasContextR...
Now: Partly Cloudy, 19° C
Mon: 25° C
Tue: 22° C

SIP > Converged SIP Modules > CallSetup

General | Descriptor | Target

SIP Module

Save

Modify an existing SIP Module.

Name: CallSetup

Context Root: /CallSetup
Path relative to server's base URL

Description: SIP Application

Status: Enabled on All Targets
Go to Target tab to control status for individual target

Availability: □ **Enabled**

Location: ${com.sun.aas.instanceRoot}/applications/j2ee-modules/CallSetup

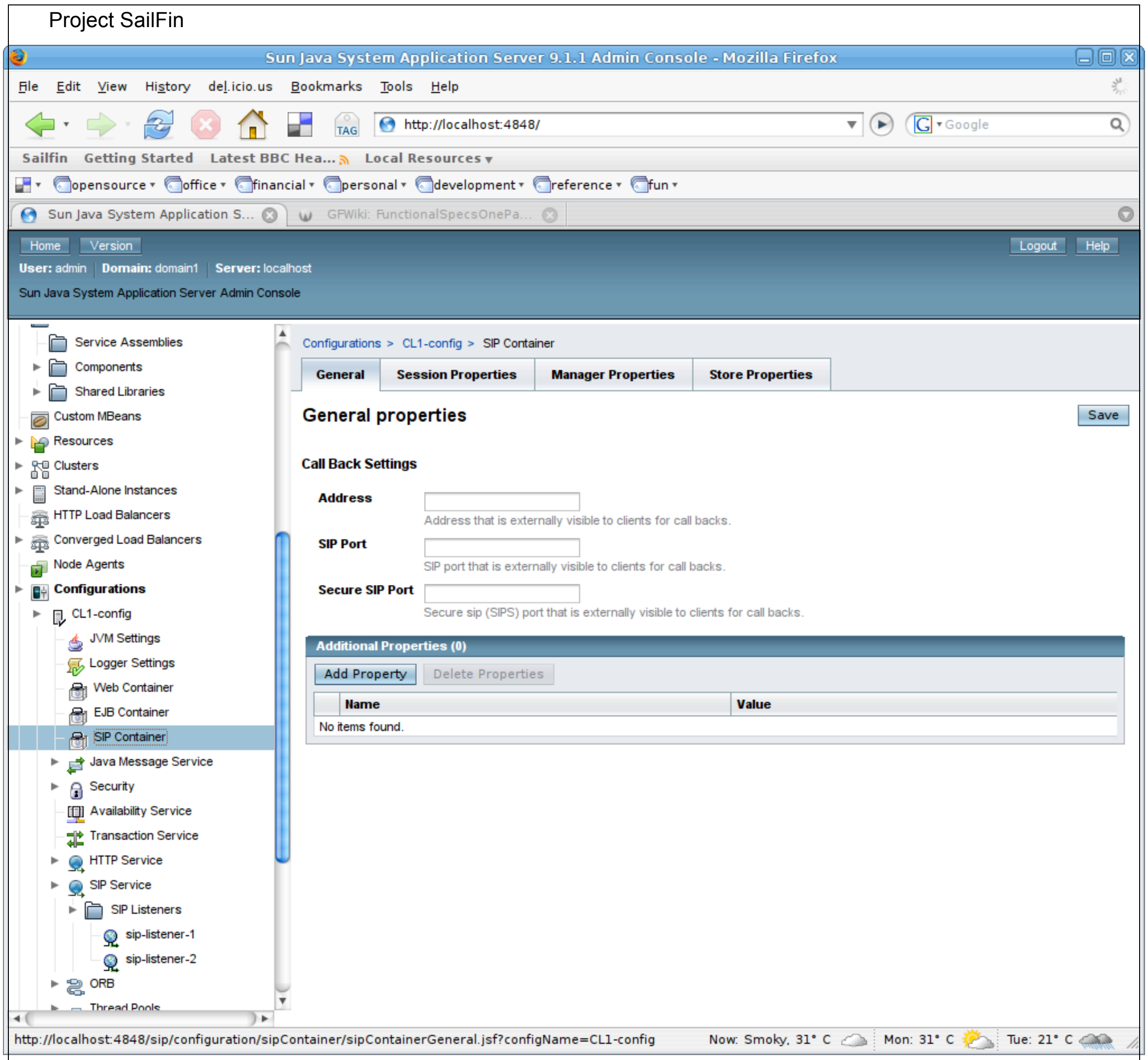
Object Type: user

Libraries:

Sub Components (9)

Name	Type
default	Servlet
jsp	Servlet
Faces Servlet	Servlet
RegistrationBrowserServlet	Servlet
listServlet	Servlet
SipCallSetupServlet	Servlet
registrar	SIP Servlet
presence	SIP Servlet
b2b	SIP Servlet

Details for a Converged SIP Module


Sun Java System Application Server 9.1.1 Admin Console - Mozilla Firefox
File Edit View History del.icio.us Bookmarks Tools Help
http://localhost:4848/
Sailfin Getting Started Latest BBC Hea... Local Resources
opensource office financial personal development reference fun
Sun Java System Application S... GFWiki: FunctionalSpecsOnePa...
Home Version Logout Help
User: admin Domain: domain1 Server: localhost
Sun Java System Application Server Admin Console
Service Assemblies
Components
Shared Libraries
Custom MBeans
Resources
Clusters
Stand-Alone Instances
HTTP Load Balancers
Converged Load Balancers
Node Agents
Configurations
CL1-config
JVM Settings
Logger Settings
Web Container
EJB Container
SIP Container
Java Message Service
Security
Availability Service
Transaction Service
HTTP Service
SIP Service
SIP Listeners
sip-listener-1
sip-listener-2
ORB
Thread Pools
Configurations > CL1-config > SIP Container
General Session Properties Manager Properties Store Properties
Save
Add Property Delete Properties
http://localhost:4848/sip/configuration/sipContainer/sipContainerGeneral.jsf?configName=CL1-config
Now: Smoky, 31° C Mon: 31° C Tue: 21° C

General properties

Call Back Settings

Address
Address that is externally visible to clients for call backs.

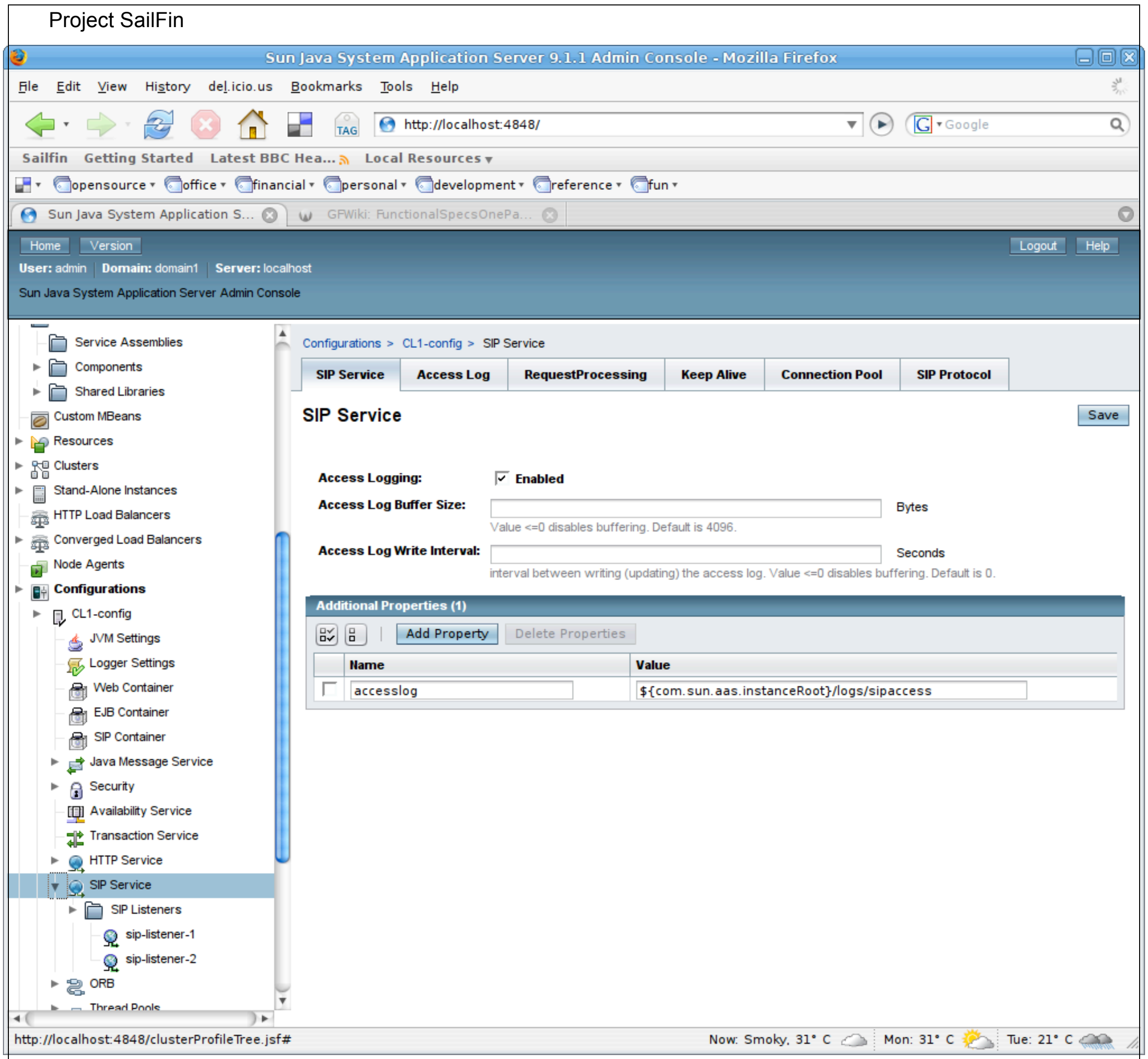
SIP Port
SIP port that is externally visible to clients for call backs.

Secure SIP Port
Secure sip (SIPS) port that is externally visible to clients for call backs.

Additional Properties (0)

Name	Value
No items found.	

SIP Container



SIP Service

Access Logging: ☑ Enabled

Access Log Buffer Size: Bytes

Value <=0 disables buffering. Default is 4096.

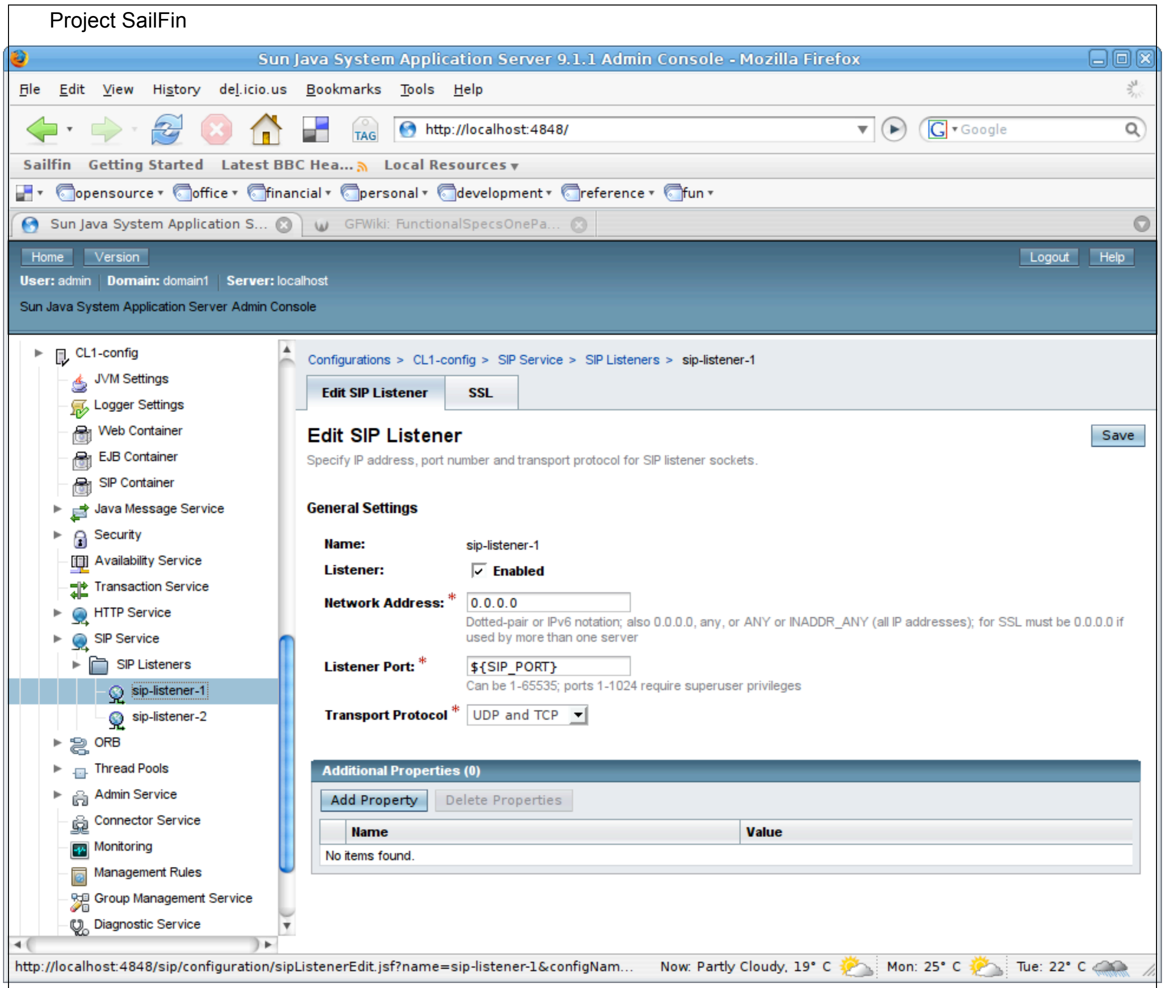
Access Log Write Interval: Seconds

interval between writing (updating) the access log. Value <=0 disables buffering. Default is 0.

Additional Properties (1)

	Name	Value
□	accesslog	${com.sun.aas.instanceRoot}/logs/sipaccess

SIP Service



User: admin **Domain:** domain1 **Server:** localhost

Sun Java System Application Server Admin Console

Edit SIP Listener

Specify IP address, port number and transport protocol for SIP listener sockets.

General Settings

Name: sip-listener-1

Listener: ☑ **Enabled**

Network Address: * 0.0.0.0

Dotted-pair or IPv6 notation; also 0.0.0.0, any, or ANY or INADDR_ANY (all IP addresses); for SSL must be 0.0.0.0 if used by more than one server

Listener Port: * ${SIP_PORT}

Can be 1-65535; ports 1-1024 require superuser privileges

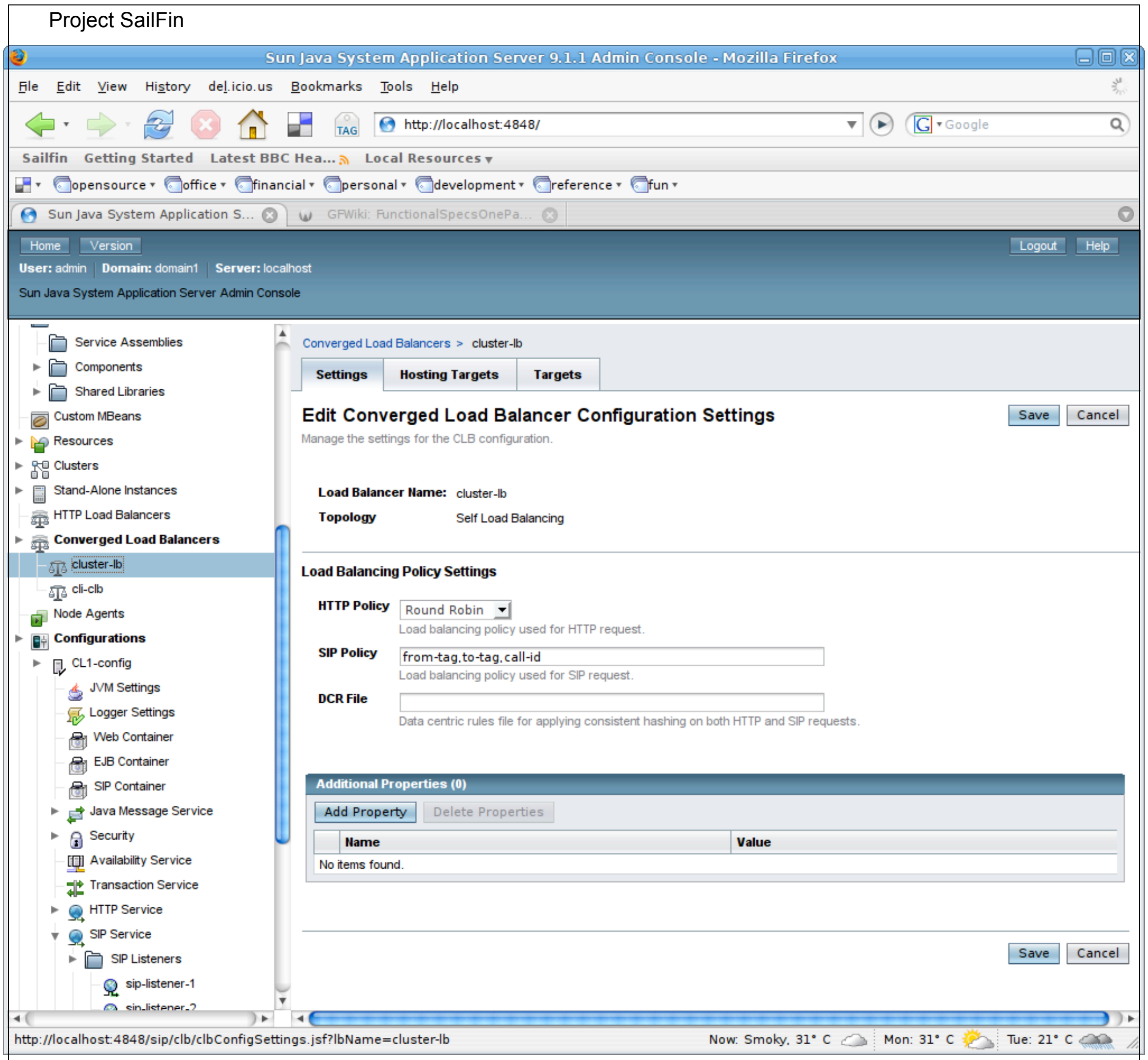
Transport Protocol * UDP and TCP

Additional Properties (0)

Name	Value
No items found.	

http://localhost:4848/sip/configuration/sipListenerEdit.jsf?name=sip-listener-1&configNam...

SIP Listeners



Edit Converged Load Balancer Configuration Settings

Manage the settings for the CLB configuration.

Load Balancer Name: cluster-lb

Topology: Self Load Balancing

Load Balancing Policy Settings

HTTP Policy: Round Robin
Load balancing policy used for HTTP request.

SIP Policy: from-tag,to-tag,call-id
Load balancing policy used for SIP request.

DCR File:
Data centric rules file for applying consistent hashing on both HTTP and SIP requests.

Additional Properties (0)

Name	Value
No items found.	

Converged Load Balancer

2.1.2.4 Internationalization

The administration GUI will be completely internationalized. The administration GUI uses the Woodstock visual components under the JSF Templating layer to display the data in various screens. The Woodstock components have an i18n strategy whereby the strings to be displayed for viewing can be picked up from an externalized properties file.

All the user viewable strings will be picked up from a defined properties file and no string for user display will be hard coded. Localizing the GUI will involve writing new properties file for that specific Locale. The Sailfin administration GUI will have its own properties file.

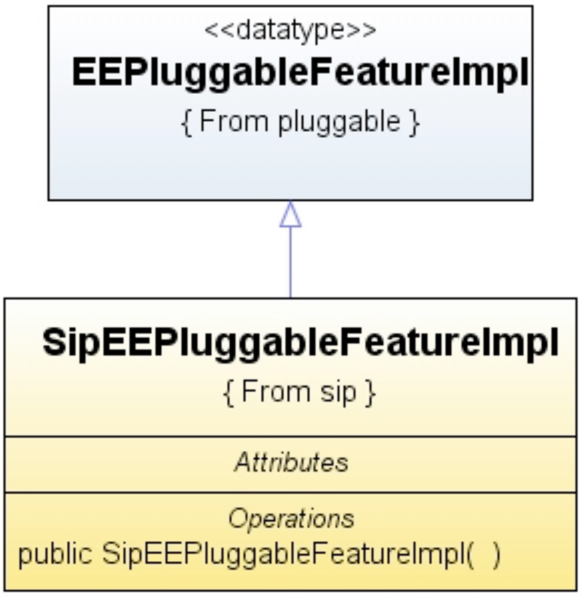
2.1.2.5 Supported Browsers

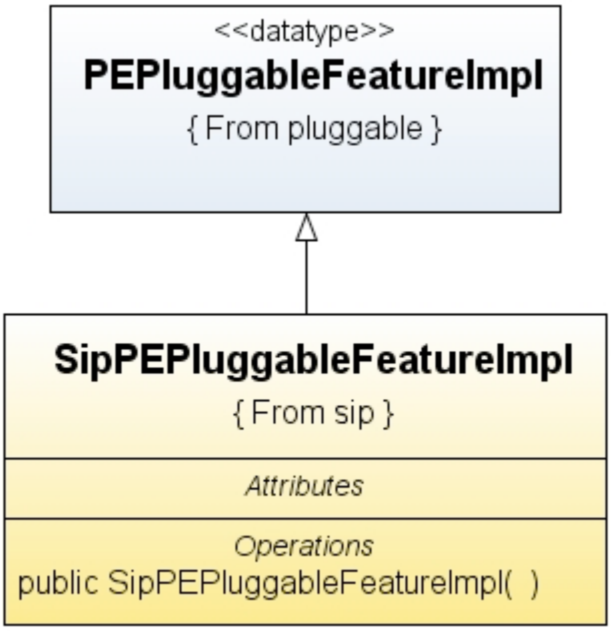
The browsers supported will be the same as supported by GlassFish administration GUI.

2.1.3 Administration Backend

2.1.2.6 Pluggable Feature Extensions

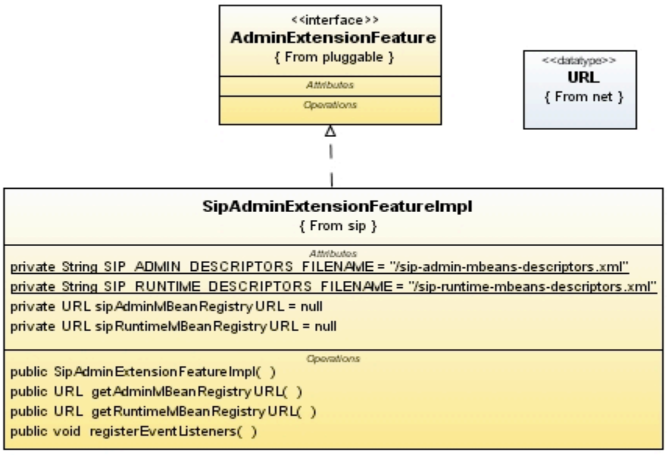
GlassFish administration infrastructure needs to be enhanced so that it would be possible to plug-in new SailFin modules like the administration MBeans, monitoring, AMX. This feature class will be defined using the existing system property *com.sun.appserv.pluggable.features* The SIP pluggable feature implementation class will extend either *PEPluggableFeatureImpl* or *EEPluggableFeatureImpl* and will contain the properties that define the implementation classes for pluggable features used within SailFin. For example *SipPEPluggableFeatureImpl* that extends *PEPluggableExtensionFeatureImpl* will load all the features for developer profile. *SipEEPluggableFeatureImpl* that extends *EEPluggableExtensionFeatureImpl* will load all the features for the cluster profile. Note that all these features will be loaded only for SailFin and not for GlassFish. The implementation classes in GlassFish will all contain no-op methods.





2.1.2.6.1 Administration Extension Feature

This feature will be used to load the JMX MBeans descriptor file for the SailFin MBeans. SailFin Mbeans will be described in *sip-admin-mbeans-descriptors.xml* (See next section for description of all the Mbeans)

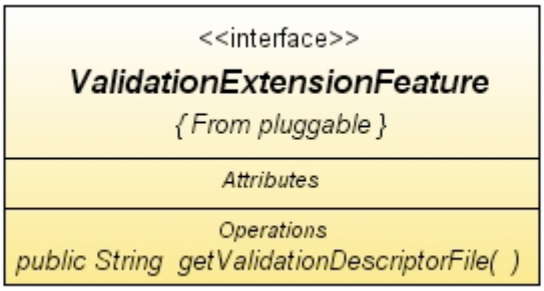

<<interface>>
AdminExtensionFeature
{ From pluggable }
Attributes
Operations
<<datatype>>
URL
{ From net }
SipAdminExtensionFeatureImpl
{ From sip }
Attributes
private String SIP_ADMIN_DESCRIPTORS_FILENAME = "/sip-admin-mbeans-descriptors.xml"
private String SIP_RUNTIME_DESCRIPTORS_FILENAME = "/sip-runtime-mbeans-descriptors.xml"
private URL sipAdminMBeanRegistryURL = null
private URL sipRuntimeMBeanRegistryURL = null
Operations
public SipAdminExtensionFeatureImpl( )
public URL getAdminMBeanRegistryURL( )
public URL getRuntimeMBeanRegistryURL( )
public void registerEventListeners( )

2.1.2.6.2 Monitoring Manager Extension Feature

This feature will be used to load all the SailFin monitoring Mbeans. See section 2.1.2.11 for details on extensions done in GlassFish monitoring SPI

2.1.2.6.3 Server Validator Extension Feature

This feature will be used to load the SailFin validation descriptors file. This file will basically describe the validation rules for all the new elements.


<<interface>>
ValidationExtensionFeature
{ From pluggable }
Attributes
Operations
public String getValidationDescriptorFile( )

2.1.2.6.4 AMX Support Extension Feature

AMX support should be extended to load the SailFin configuration and monitoring AMX Mbeans. Please see the AMX Extension design document for more details.

2.1.2.7 SailFin Administration MBeans

The *AdminExtensionFeature* will be used by *MBeanRegistryFactory* to load the new JMX Mbeans defined in the descriptor file *sip-admin-mbean-descriptors.xml*

The purpose of the Mbeans listed in the table below is to access and modify the persistent data (domain.xml) on DAS and reconfigure instances with these changes dynamically.

Sl No	Mbean Type	ObjectName/XPath/Dotted Name
1	sip-configs	ObjectName:{0}:type=sip-configs,category=config Xpath:/domain/configs Dotted Name:domain.configs
2	sip-service	ObjectName:{0}:type=sip-service,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service Dotted Name:{1}.sip-service
3	sip-listener	ObjectName:{0}:type=sip-listener,id={2},config={1},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/sip-listener[@id='{2}'] Dotted Name:{1}.sip-service.sip-listener.{2}
4	sip-protocol	ObjectName:{0}:type=sip-protocol,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/sip-protocol Dotted Name:{1}.sip-service.sip-protocol
5	sip-link	ObjectName:{0}:type=sip-link,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/sip-protocol/sip-link Dotted Name:{1}.sip-service.sip-protocol.sip-link
6	sip-timers	ObjectName:{0}:type=sip-timers,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/sip-protocol/sip-timers Dotted Name:{1}.sip-service.sip-protocol.sip-timers
7	converged-lb-configs	ObjectName:{0}:type=converged-lb-configs,category=config Xpath:/domain/converged-lb-configs Dotted Name:domain.converged-lb-configs
8	converged-lb-config	ObjectName:{0}:type=converged-lb-config,name={1},category=config Xpath:/domain/converged-lb-configs/converged-lb-config[@name='{1}'] Dotted Name:domain.converged-lb-configs.{1}
9	converged-lb-policy	ObjectName:{0}:type=converged-lb-policy,converged-lb-config={1},category=config Xpath:/domain/converged-lb-configs/converged-lb-config[@name='{1}']/converged-lb-policy Dotted Name:domain.converged-lb-configs.{1}.converged-lb-policy
10	converged-lb-cluster-ref	ObjectName:{0}:type=converged-lb-cluster-ref,ref={2},converged-lb-config={1},category=config Xpath:/domain/converged-lb-configs/converged-lb-config[@name='{1}']/converged-lb-cluster-ref[@ref='{2}'] Dotted Name:domain.converged-lb-configs.{1}.converged-lb-cluster-ref.{2}
11	converged-load-balancer	ObjectName:{0}:type=converged-load-balancer,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/availability-service/converged-load-balancer Dotted Name:{1}.availability-service.converged-load-balancer

12	proxy	ObjectName:{0}:type=proxy,config={1},category=config Xpath:/domain/configs/config[@name='{1}']/availability-service/converged-load-balancer/proxy Dotted Name:{1}.availability-service.converged-load-balancer.proxy
13	ssl	ObjectName:{0}:type=ssl,config={1},sip-listener={2},category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/sip-listener[@id='{2}']/ssl Dotted Name:{1}.sip-service.sip-listener.{2}.ssl
14	access-log	ObjectName:{0}:type=access-log,config={1},parent=sip-service,category=config Xpath:{0}:type=access-log,config={1},parent=sip-service,category=config Dotted Name:{1}.sip-service.access-log
15	request-processing	ObjectName:{0}:type=request-processing,config={1},parent=sip-service,category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/request-processing Dotted Name:{1}.sip-service.request-processing
16	keep-alive	ObjectName:{0}:type=keep-alive,config={1},parent=sip-service,category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/keep-alive Dotted Name:/domain/configs/config[@name='{1}']/sip-service/keep-alive
17	connection-pool	ObjectName:{0}:type=connection-pool,config={1},parent=sip-service,category=config Xpath:/domain/configs/config[@name='{1}']/sip-service/connection-pool Dotted Name:{1}.sip-service.connection-pool
18	sip-container	ObjectName:{0}:type=sip-container,config={1},category=config Xpath:{0}:type=sip-container,config={1},category=config Dotted Name:{1}.sip-container
19	session-config	ObjectName:{0}:type=session-config,config={1},container=sip-container,category=config Xpath:/domain/configs/config[@name='{1}']/sip-container/session-config Dotted Name:/domain/configs/config[@name='{1}']/sip-container/session-config
20	session-manager	ObjectName:{0}:type=session-manager,config={1},container=sip-container,category=config Xpath:/domain/configs/config[@name='{1}']/sip-container/session-config/session-manager Dotted Name:{1}.sip-container.session-config.session-manager
21	manager-properties	Object Name:{0}:type=manager-properties,config={1},container=sip-container,category=config Xpath:/domain/configs/config[@name='{1}']/sip-container/session-config/session-manager/manager-properties Dotted Name:{1}.sip-container.session-config.session-manager.manager-properties

22	store-properties	ObjectName:{0}:type=store-properties,config={1},container=sip-container,category=config Xpath:/domain/configs/config[@name='{1}']/sip-container/session-config/session-manager/store-properties Dotted Name:{1}.sip-container.session-config.session-manager.store-properties
23	session-properties	ObjectName:{0}:type=session-properties,config={1},container=sip-container,category=config Xpath:/domain/configs/config[@name='{1}']/sip-container/session-config/session-properties Dotted Name:{1}.sip-container.session-config.session-properties
24	identity-assertion-trust	ObjectName:{0}:type=identity-assertion-trust,id={2},config={1},category=config Xpath:/domain/configs/config[@name='{1}']/security-service/identity-assertion-trust[@id='{2}'] Dotted Name:{1}.security-service.identity-assertion-trust.{2}
25	trust-handler	ObjectName:{0}:type=trust-handler,id={2},config={1},category=config Xpath:/domain/configs/config[@name='{1}']/security-service/identity-assertion-trust[@id='{2}']/trust-handler Dotted Name:{1}.security-service.identity-assertion-trust.{2}.trust-handler
26	trusted-entity	ObjectName:{0}:type=trusted-entity,id={3},identity-assertion-trust={2},config={1},category=config Xpath:{0}:type=trusted-entity,id={3},identity-assertion-trust={2},config={1},category=config Dotted Name:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}
27	ip-address	ObjectName:{0}:type=ip-address,trusted-entity={3},identity-assertion-trust={2},config={1},category=config Xpath:/domain/configs/config[@name='{1}']/security-service/identity-assertion-trust[@id='{2}']/trusted-entity[@id='{3}']/ip-address Dotted Name:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.ip-address
28	host-name	ObjectName:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.ip-address Xpath:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.ip-address Dotted Name:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.host-name
29	principal	ObjectName:{0}:type=principal,trusted-entity={3},identity-assertion-trust={2},config={1},category=config Xpath:/domain/configs/config[@name='{1}']/security-service/identity-assertion-trust[@id='{2}']/trusted-entity[@id='{3}']/principal Dotted Name:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.principal

30	converged-lb-server-ref	ObjectName:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.principal Xpath:{1}.security-service.identity-assertion-trust.{2}.trusted-entity.{3}.principal Dotted Name:domain.converged-lb-configs.{1}.server-ref.{2}

2.1.2.8 Server config beans

Server beans (config beans) need to be generated for all the new elements. The new elements have to be described in RNG format. There will be new elements for SIP listener configuration, converged load balance configuration, security configuration.

2.1.2.9 Event framework support

To support dynamic reconfiguration, element change event listener interfaces need to be exposed. The actual notification piece or the event listener implementation that will dynamically reload the attributes will be done by respective modules. SIP container related listeners will be implemented by the SIP container administration backend (i.e the integration module). Converged Load Balancer related listeners will be implemented by the CLB module. The new SIP pluggable feature for admin will expose a method to register the listener implementations and its corresponding events in the *AdminEventListenerRegistry* which can be invoked from the SIP container/CLB startup.

2.1.2.10 Converged Load Balancer Administration

The converged load balancer is an intrinsic component of the Sailfin instance. An administrator can configure the converged load balancer(CLB) using CLI / GUI. A new set of commands are provided to create and delete the converged load balancers and their configuration.

Configuration:

The runtime configuration for the Converged Load Balancer is specified in *converged-loadbalancer.xml*. This file resides in the central repository(CR) maintained by DAS(Domain Administration Server). During the creation of SailFin instances, the file gets synchronized along with other configuration files in CR.

CLB uses another policy configuration file called Data Centric Rules file (or dcr.xml) The selection of the server instance to forward the request, is based on a hash key. The key is extracted from incoming SIP and HTTP requests according to suitable rules. All applications share the same rules defined in a common rule file. The DCR file can be manually edited and uploaded to DAS. Once the file is uploaded the instances are notified to sync the file.

Dynamic re-configuration of the Converged Load balancer :

The CLB administration infrastructure in the DAS listens for the *ConfigContextEvent* and updates the *converged-loadbalancer.xm*l. If there is a change in the *converged-loadbalancer.xml*, an admin event is triggered to notify the SailFin LB instances. The LB instances upon receiving the event notification, pull the *converged-loadbalancer.xml* from the central repository using synchronization API. Once the local copy gets updated, the loadbalancer is dynamically reconfigured.

Similarly, DAS will generate an event (whenever a rules file is uploaded to DAS) that will be sent to the instances which will then pull this file into its config directory.

2.1.2.11 Monitoring

The current EAS (SailFin) statistics need to be exposed to the GlassFish monitoring framework. The various Stats instances need to be registered with *MonitoringRegistry*. *MonitoringRegistry* converts these Stats Objects into MBeans using reflection and presents them as attributes to the management clients (CLI and GUI).

To plug-in SIP monitoring, a new feature needs to be introduced in the pluggable feature factory. This feature will basically get a handle to the new SIP monitoring manager that will handle registration/unregistration. To support pluggability, the following changes need to be done:

1. *MonitoredObjectType.java* : Expose a method to create new monitoring objects.
2. *MonitoringObjectNames.java*: Expose method to create a new ObjectName for some external entity.
3. *DottedNameFactory.java*: Expose methods to generate a specific dotted name for a monitored object
4. *MonitoringRegistrationHelper.java*: During the monitoring initialization, this class will load external monitoring modules.
5. *StatsDescriptionHelper.java*: Expose methods to load external resource bundles.

For dynamic reconfiguration support, a new property “sip-service” will be introduced under module-monitoring-level. *ModuleMonitoringLevel* event listener needs to be enhanced to listen for property changes. By default, “sip-service” property will be set to “OFF”. When the property changes to “LOW”, “HIGH”, the SailFin monitoring objects will be registered.

Please refer to container integration specification for details on the actual statistics that will be exposed for the various layers in EAS.

2.2 Profile Management

The following configuration profiles will be supported. These profiles will basically overwrite the GlassFish profiles with new transformation rules specific to SailFin. GlassFish profiles operate on a default domain.xml template file. We will need to add extra style sheets to introduce the default SIP elements for the various modules.

- **Developer:** The *profile.propertie*s will include a new file *sipdomain.xml.xs*l which will contain the XSL rules for introduction of all the new SIP elements. Additional tokens will be introduced for configuring the default SIP ports.
- **Cluster:** Cluster profile setup will require 4 new style sheets to define the transformation rules. The cluster profile will configure a cluster (without any instances) with self load balancing rules which will be the default cluster topology.

Please refer to GlassFish profiles document for more details on profile management.

2.3 SIP Container Integration

The administration interfaces will be consumed by the SIP Servlet Container. The following features being implemented in SIP container will be discussed in a separate specification:

- Configuration
- Monitoring
- Internationalized Logging
- Application Verifier
- Call Flow and SIP Message Inspection

3 Performance

<How do you want performance team to measure this sub-system? Any micro benchmarks necessary?Any goals? Anticipated scalability limits or goals?>

Performance requirements will be covered in the umbrella specification.

4 Management

<Describe how performance, management status, and diagnostic information is exposed. How does this feature handle dynamic configuration changes?>

4.1 Interfaces

<How is this feature(s) configured by administrator? Does it introduce new commands or modify existing ones? Show syntax of expected administrative commands and response codes. What is the schema for new configuration? Show the DTD snippets. What are their default values? What are the validation rules? List stability level for each of the above [committed|evolving|unstable|standard]. Does it consume interfaces from other projects or sub-systems (imported) or produce interfaces for consumption (exported).>

4.1.1 Exported Interfaces

Interface	Proposed Stability Classification	Specified in	Comments
Configuration	EVOLVING	sun-domain_1_4.dtd	Contains all the SIP configuration parameters
Administration CLI	EVOLVING	This document	New SailFin admin commands
Administration GUI	EVOLVING	This document	The document contains the high level architecture and GlassFish administration GUI pluggability proposal.

2.1.3

4.1.2 Configurables

The get/set commands of *asadmin* can be used to manipulate the following SIP parameters.

- **sip-service**
  Attributes list for elements under sip-service

Element	Attribute	Definition	Default	Range
server.sip-service.sip-protocol-config	error-response-enabled	Boolean flag to denote if error response should be sent. If "true", respond 400 on bad request or drop. Default is "false"	false	true|false
	default-tcp-transport	Boolean flag to denote if transport=tcp should be inserted in URI of contact and record-route header. Default is "false"	false	true|false
server.sip-service.sip-link	connection-alive-timeout-in-seconds	Defines the time, in seconds, a SIP link is allowed to be inactive before the connection is closed. Default is 120 seconds.	120	

	max-queue-length	Defines the maximum number of simultaneous write requests or connect requests, or both, that can be waiting to write on a link. Default is 50 requests.	50	
	write-timeout-in-millis	Defines timeout value in milliseconds for a single write operation of a SIP link. Default is 10ms.	10	1-50
	write-timeout-retries	Defines the number of retries to perform a single write operation of a SIP link. Default is 25.	25	1-25
server.sip-service.sip-timers	t1-in-millis	SIP timer T1 (round trip time estimate) in milliseconds. For unreliable transports (such as UDP), the client transaction retransmits requests at an interval that starts at T1 seconds and doubles after every transmission. T1 is an estimate of the round-trip time (RTT), and it defaults to 500 ms. Nearly all of the transaction timers described here scale with T1, and changing T1 adjusts their values.	500	
	t2-in-millis	SIP timer T2 (maximum retransmit interval for non-INVITE requests and INVITE responses) in milliseconds. For unreliable transports, requests are retransmitted at an interval which starts at T1 and doubles until it hits T2. If a provisional response is received, retransmissions continue for unreliable transports, but at an interval of T2. The default value of T2 is 4000ms, and it represents the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately.	4000	
	t4-in-millis	SIP timer T4 represents the amount of time the network will take to clear messages between client and server transactions. The default value of T4 is 5000ms.	5000	

- **DNS java**
  DNS agent configuration properties can be set using the create-jvm-options command. See http://www.dnsjava.org/dnsjava-current/README

Element	VM Option	Definition	Default	Range
server.java-config.jvm-options	dns.search	Defines a comma separated list of DNS servers. The servers can either be IP addresses or hostnames (which are resolved using Java's built in DNS support)		
	dns.server	Defines a comma separated list of DNS servers that is used when querying external domains. The servers can either be IP addresses or hostnames (which are resolved using Java's built in DNS support)	""	
	enum.top.domain	Defines the top domain to be used when translating a global telephone number to a ENUM domain name.	e164.arpa	

4.1.3 DTD changes for SIP container

The following new elements have been introduced:

λ **sip-service**: 0 or 1 occurrence of this element to represent a SIP service configuration containing various settings for access-log, SIP listeners, request processing, keep alive, connection pool, SIP protocol and additional properties.

```
<!ELEMENT sip-service
   (access-log?, sip-listener+, request-processing?, keep-alive?,
   connection-pool?, sip-protocol?, property*)>
```

λ **sip-listener**: 1 or more occurrences of this element to represent SIP listener(s). Default SIP listener listens on port 5060. For secure mode (transport=tls), port is 5061.

```
<!ELEMENT sip-listener (ssl?, property*)>


<!ATTLIST sip-listener
   id CDATA #REQUIRED
   address CDATA #REQUIRED
   port CDATA #REQUIRED
   transport (udp_tcp | tls) "udp_tcp"
   enabled %boolean; "true">
```

Attributes for sip-listener:

address

IP address of the listen socket. Can be in dotted-pair or IPv6 notation. Can also be any for INADDR-ANY. Configuring a listen socket to listen on any is required if more than one sip-listener is configured to it.

id

Unique identifier for sip listener.

port

Port number to create the listen socket on. Legal values are 1 - 65535. On Unix, creating sockets that listen on ports 1 - 1024 requires superuser privileges. Default SIP listener port is 5060. When transport=tls, it is 5061.

transport

Specifies the type of transport layer protocol. Default is “tcp_udp”

λ **sip-container-availability**: 0 or 1 occurrence representing SIP session persistence settings and additional properties.

```
<!ENTITY % sip-session-save-frequency "(sip-transaction)">
```

```
<!ELEMENT sip-container-availability (property*)>

<!ATTLIST sip-container-availability
    availability-enabled %boolean; #IMPLIED
    persistence-type CDATA "memory"
    persistence-frequency %sip-session-save-frequency; #IMPLIED
    persistence-scope %session-save-scope; #IMPLIED
    repair-during-failure %boolean; "true">
```

Attributes for sip-container-availability:

availability-enabled

This boolean flag controls whether availability is enabled for SIP session persistence.

persistence-type

Specifies the sip session persistence mechanism for sip applications that have availability enabled. Default is "memory".

repair-during-failure

Specifies whether a forward and reverse repair should be performed on an instance that has (re)joined the cluster.

persistence-frequency

The persistence frequency used by the session persistence framework.

persistence-scope

The persistence scope used by the session persistence framework.

λ **sip-protocol**: 0 or 1 occurrence containing settings for SIP link, SIP timers and additional properties.

```
<!ELEMENT sip-protocol (sip-link?, sip-timers?, property*)>


<!ATTLIST sip-protocol
    error-response-enabled %boolean; "false"
    default-tcp-transport %boolean; "false">
```

Attributes for sip-protocol:

default-tcp-transport

Boolean flag to denote if transport=tcp should be inserted in URI of contact and record-route header. Default is "false".

error-response-enabled

Boolean flag to denote if error response should be sent. If "true", respond 400 on bad request or drop. Default is "false".

λ **sip-link**: 0 or 1 occurrence representing SIP connection settings.

```
<!ELEMENT sip-link EMPTY>


<!ATTLIST sip-link
   connection-alive-timeout-in-seconds CDATA "120"
   max-queue-length CDATA "50"
   write-timeout-in-millis CDATA "10"
   write-timeout-retries CDATA "25">
```

Attributes for sip-link:

connection-alive-timeout-in-seconds

Defines the time, in seconds, a SIP link is allowed to be inactive before the connection is closed. Default is 120 seconds.

max-queue-length

Defines the maximum number of simultaneous write requests or connect requests, or both, that can be waiting to write on a link. Default is 50 requests.

write-timeout-in-millis

Defines timeout value in milliseconds for a single write operation of a SIP link. Range 1-50ms. Default is 10ms.

write-timeout-retries

Defines the number of retries to perform a single write operation of a SIP link. Range 1-25. Default is 25.

λ **sip-timers**: 0 or 1 occurrence representing SIP timers settings. Detailed descriptions for the timers can be found in RFC 3261 Chapter 17

```
<!ELEMENT sip-timers EMPTY>

<!ATTLIST sip-timers
   t1-in-millis CDATA "500"

   t2-in-millis CDATA "4000"
   t4-in-millis CDATA "5000">
```

Attributes for sip-timers:

t1-in-millis

SIP timer T1 (round trip time estimate) in milliseconds. For unreliable transports (such as UDP), the client transaction retransmits requests at an interval that starts at T1 seconds and doubles after every retransmission. T1 is an estimate of the round-trip time (RTT), and it defaults to 500 ms. Nearly all of the transaction timers described here scale with T1, and changing T1 adjusts their values.

t2-in-millis

SIP timer T2 (maximum retransmit interval for non-INVITE requests and INVITE responses) in milliseconds. For unreliable transports, requests are retransmitted at an interval which starts at T1 and doubles until it hits T2. If a provisional response is received, retransmissions continue for unreliable transports, but at an interval of T2. The default value of T2 is 4000ms, and it represents the amount of time a non-INVITE server transaction will take to respond to a request, if it does not respond immediately.

t4-in-millis

SIP timer T4 represents the amount of time the network will take to clear messages between client and server transactions. The default value of T4 is 5000ms.

4.1.4 DTD changes for converged load balancer administration

The following new elements related to converged load balancer configuration are described here:

λ **converged-lb-configs**: 0 or 1 occurrence of this element which describes the configured converged load balancer configurations in a domain.

```
<!ELEMENT converged-lb-configs (converged-lb-config*)>
```

λ **converged-lb-config**: 0 or more occurrences of this element which represents a converged load balancer view of the load balanced deployment. This is used by the converged load balancer to configure itself.

```
<!ELEMENT converged-lb-config

   (converged-lb-policy, (converged-lb-cluster-ref* | server-ref*),    property*)>


<!ATTLIST converged-lb-config
   name CDATA #REQUIRED>
```

Attributes for convered-lb-config:

name

The name of the converged load balancer configuration

- λ **converged-lb-cluster-ref**: 0 or more occurrences of this element relating to a converged cluster that needs to be load balanced.

```
<!ELEMENT converged-lb-cluster-ref EMPTY>


<!ATTLIST converged-lb-cluster-ref
    ref CDATA #REQUIRED
    self-loadbalance %boolean; "true">
```

Attributes for converged-lb-cluster-ref:

ref

A name to converged cluster defined.

self-loadbalance

Boolean attribute which specifies whether configured cluster self load balances incoming requests to itself. If it's configured to do so, load balancer is an intrinsic component of the participating server instances in the cluster. Default value is “true”.

- λ **converged-lb-policy**: Specifies the load balancing policy used by the converged load balancer.

```
<!ELEMENT converged-lb-policy (hash-rules, property*)>


<!ATTLIST converged-lb-policy
    http CDATA "round-robin"
    sip CDATA "from-tag,to-tag,call-id"
    dcr-file CDATA #IMPLIED>
```

Attributes for converged-lb-policy:

dcr-file

It denotes the data centric rules file name, which provides the rules for applying consistent hashing on both HTTP and SIP requests.

http

Specifies the load balancing policy used for the http requests. The default implied value is round-robin.

sip

Specifies the parameters on which consistent hashing policy is applied to obtain the hashkey. This can be specified as single value or comma seperated values of parameter names to hash on. In case more than one parameter is specified, then the concatenated values of the parameters would be used for applying the consistent hashing. The default value implied will be "from-tag,to-tag,call-id".

λ **converged-load-balancer**: 0 or 1 occurrence of this element describing a converged load balancer. JavaEE server upon startup can come up with a load balancer that facilitates high availability of converged application deployments - SIP and HTTP requests.

```
<!ELEMENT converged-load-balancer (proxy)>


<!ATTLIST converged-load-balancer
   name CDATA #REQUIRED
   config-file CDATA #REQUIRED
   auto-commit CDATA "false"

   converged-lb-config-name CDATA #REQUIRED>
```

Attributes for converged-load-balancer:

auto-commit

when true, config file changes should be propagated to clb instances immediately.

name

Converged load balancer's name.

config-file

Converged load balancer's configuration file. This can be an absolute or relative path. Relative path is resolved with the config directory of the server.

- **proxy**: Specifies the Proxy component of the JavaEE server. This would provide for pass through of HTTP/SIP requests to remote HTTP/SIP endpoint; that is configured via http-service/sip-service of the remote instance.

```
<!ELEMENT proxy (property*)>

<!ATTLIST proxy
   request-pool-size CDATA "50"
   send-retry-count CDATA "3"
   read-timeout-in-millis CDATA "1500">
```

Attributes:

read-timeout-in-millis
the duration for which we would wait for data from the client
in the socket channel

request-pool-size
denotes the number of request objects that will be created
and pooled by the proxy

send-retry-count
the number of retries the proxy would attempt with the remote
instance when sending of data fails.

Known Properties :
Eventually some of them may become attributes.

max-parallel-connections
maximum number of outbound connections to a backend
instance.

high-water-mark
maximum number of active outbound connections Controller
will handle.

connections-to-reclaim
number of LRU connections, which will be reclaimed in
case highWaterMark limit will be reached.

proxy-server-read-timeout
Timeout for proxy to read from the server channel once
response is available.

socket-receive-buffer-size
proxy-backend socket receive buffer size.

socket-send-buffer-size
proxy-backend socket send buffer size.

client-socket-read-timeout
client socket send read timeout.

4.1.5 DTD changes for Identity Assertion Trust Configuration

Identity assertion trust domain configuration information as per RFC 3325. P-Asserted - Identity header received from hosts and domains can be trusted. P-Asserted-Identity header has identity of a user who was authenticated at another node in the network.

Please refer to the SailFin security specification for more details.

```
<!ELEMENT identity-assertion-trust ((trusted-entity* | trust-handler))>
```

```
<!ATTLIST identity-assertion-trust
    id CDATA #REQUIRED

    is-default %boolean; "false">


<!-- trusted-entity
    Trusted intermediate trusted hosts/domains as per RFC 3325.


  attributes
    id
        Unique identifier for the trusted entity.

    trusted-as
                trusted-as with value 'intermediate' represents configuration information
                for incoming messages,if it has value 'destination' then configuration
                under trusted-entity is applied to outgoing messages.


  Used in:
    identity-assertion-trust
-->

<!ELEMENT trusted-entity (ip-address, host-name?, principal?)>


<!ATTLIST trusted-entity
    id CDATA #REQUIRED
    trusted-as (intermediate | destination) #IMPLIED>


<!-- trust-handler

  Used in:
    identity-assertion-trust

-->

<!ELEMENT trust-handler EMPTY>

<!ATTLIST trust-handler
    class-name CDATA #REQUIRED>

<!-- ip-address
    Identifies the trusted host on the network.eg : 129.169.223.2

  Used in:
    trusted-intermediate
```

```
-->

<!ELEMENT ip-address (#PCDATA)>

<!-- domain-name
   Identifies the trusted host on the network using domain names.
   eg: sun.com, cisco.com. All hosts from sun.com domain are trusted.

  Used in:
   trusted-intermediate

-->

<!ELEMENT host-name (#PCDATA)>
```

2.1.4 Stack Layer configuration:

Defines the configuration of a stack of layers, typically related to protocols, such as SIP. Used in sip-container

```
<!ELEMENT stack-config (stack-layer*, property*)>


<!ATTLIST stack-config

   layer-order CDATA #REQUIRED>

Attributes:
   layer-order
      A comma seperated list indicating the order of the stack
      layers. Use the the stack-layer.id for the list elements.

<!-- stack-layer
   Defines a layer of a stack.


  children
   property
      Any property is a javabean property injected in the layer
      class, if a corresponding javabean setter exist in the layer
      class.


  attributes
   class-name
      Fully qualified name of the layer class.
   id
      Unique identifier for stack-layer.


  Used in:
   stack-config
```

```
-->

<!ELEMENT stack-layer (property*)>

<!ATTLIST stack-layer
   id CDATA #REQUIRED
   class-name CDATA #REQUIRED>
```

4.1.6 New CLI Commands

4.1.6.1 SIP listener

The commands to create, delete and list SIP listener. The target option (create and delete commands) and operand (list command) is the name of the target being operated on.

Syntax

- create-sip-listener [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4949] [--secure | -s] [--user admin_user] [--passwordfile file_name] [--transport udp_tcp|tls] [--enabled=true] [--target target(Default server)] --siplisteneraddress address --siplistenerport listener_port sip_listener_id
- delete-sip-listener [--target target] sip_listener_id
- list-sip-listeners target

Argument	Definition	Default	Range
--siplisteneraddress	IP address or hostname (resolvable by DNS) to be used for listening to requests.		
--siplistenerport	Port number to create the listen socket on. On Unix, creating sockets that listen on ports 1 - 1024 requires super user privileges. Default SIP listener port is 5060. When transport=tls, it is 5061.	5060|5061	1-65535
--transport	Specifies the type of transport layer protocol.	udp_tcp	udp_tcp|tls
--enabled	Boolean attribute. If set to "false" this module will not be loaded at server start up.	true	
--target	Name of target being operated on. The valid targets for this command are config, instance, cluster, or server. By default the target is the 'server'.	server	

4.1.6.2 Converged Load Balancer

Commands to create delete and list converged load balancers and the converged load balancer configs.

λ **create-converged-lb-config**
The create-converged-lb-config command is used to create a converged load balancer configuration. The converged load balancer configuration name must be unique in the domain, and must not conflict with any Node Agent, configuration, cluster, or server instance names in the domain. This gives a flat

name space so that the dotted name notation can be used to access any of these entities without ambiguity. If config_name operand is not specified and --target option is specified, then a load balancer configuration is implicitly created with an unique name. If --target option is not specified and config_name operand is specified, then a load balancer configuration is created but with no references to any target. If both --target option and config_name operand are specified, then a lb configuration is created referencing the specified target. An error is displayed if neither --target option or config_name operand is not specified on the command line.

Syntax

create-converged-lb-config [--reloadinterval 60] [-httplbpolicy policy_name] [--siplbpolicy policy_name] [--dcrfilename file_name] [--target target] [--property (name=value)[:name=value]*] [clb_config_name]

Argument	Definition
--reloadinterval	reload pool interval in seconds at which load balancer has changed. If it has change, load balancer would reload it. Value of 0 would imply that polling is disabled.
--target	The target can either be a cluster or an unclustered instance.
--httplbpolicy	policy name to be used for routing the http requests
--siplbpolicy	policy name to be used for routing the http requests
--dcrfilename	filename of an xml file where complex rules are specified for both http and sip requests.
--property	name/value pairs
config_name	The name of the new converged load balancer configuration. This name must not conflict with any other load balancer groups, agents, configurations, clusters, or server instances in the domain.

- **delete-converged-lb-config**

The delete-converged-lb-config command deletes a load balancer configuration. The load balancer must not reference any clusters or instances before it can be deleted.

Syntax

delete-converged-lb-config clb_config_name

Argument	Definition
clb_config_name	The name of the converged load balancer configuration to delete.

- **list-converged-lb-configs**

  The list-converged-lb-configs command lists the load balancer config and its clusters/instances, or all the load balancer config in the domain.

  Syntax

  list-converged-lb-configs [target]

Argument	Definition
target	target can either be a instance name or cluster. If the target is either a cluster or an instance, then the lb config that references to the the cluster or instance will be displayed. If target is not specified, then the list of all lb configs will be displayed.

- **create-converged-lb-ref**
The create-converged-lb-ref command is used to add an existing cluster to an existing load balancer configuration.

Syntax

create-converged-lb-ref –clbconfig config_name [--selfloadbalance] [--lbEnableAllInstances] [--lbEnableAllApplications] target

Argument	Definition
--clbconfig	The name of the converged load balancer configuration
--selfloadbalance	If it is true, cluster load balances the incoming requests to itself.
--lbenableallinstances	Enable all the associated instances for a target cluster
--lbenableallapplications	enable all the associated applications for a target
target	The target to be added to the load balancer configuration. Target can either be a cluster or an instance name.

- **delete-converged-lb-ref**
The delete-converged-lb-ref command is used to delete a cluster/instance reference from a load balancer config. It is important to note that if you wish not to interrupt users accessing applications in the server you should ensure that all of its instances have been disabled before removing the cluster.

Syntax

delete-converged-lb-ref --clbconfig config_name target

Argument	Definition
--clbconfig	The name of the converged load balancer configuration
target	The target to be deleted from the load balancer configuration. target can either be a cluster or an instance name

- **create-converged-lb**
The create-converged-lb command is used to create a converged loadbalancer.

Syntax

create-converged-lb --clbconfig config_name [--configfile file_name] [--autocommit] [--property (name=value)[:name=value]*] [--target target] load_balancer_name

Argument	Definition
--clbconfig	Name of the converged-lb-config used by this converged load balancer
--configfile	File name of the converged load balancer
--autocommit	Immediately propagate the changes to clb instances
--property	Name/value pairs
--target	target config name

load_balancer_name	Name of the converged load balancer

- **delete-converged-lb**
  Command to delete the loadbalancer element.

  Syntax

  delete-converged-lb [target (default server)]

Argument	Definition
load_balancer_name	lb name

- **list-converged-lbs**
  Command to list all converged LB configs.

  Syntax

  list-converged-lbs

- **set-dcr-file**
  Command to upload the DCR file to DAS. The uploaded file will then be synced up by the instances.

  Syntax

  set-dcr-file --dcrfile dcr_file_name [--echo=false] [--interactive=true] [--host localhost] [--port
  4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] clb_config_name

Argument	Definition
dcrfile	Path to the local dcr file which needs to be uploaded
clb_config_name	CLB config pertaining to the DCR file

2.1.4.3 Identity Assertion Trust Management Commands

Command to create/delete/list trust configurations.

- create-trust-config [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] [--isdefault=false] [--trusthandler=class_name] [--property (name=value)[:name=value]*] [--trustedas=intermediate|destination] [--ipaddress=ip_address] [--hostname=host_name] [--principal=principal_name] [--entityid=trusted_entity_id] [--target target(Default server)] [trust-id]

Note that the trust config and entity ids are automatically generated. They will be of form trustid-0, trustid-1.. entityid-0, entityid-1..

Invoking this command without any options will create a trust configuration with a default trust handler.

Argument	Definition
isdefault	Boolean flag to mark/unmark a trust config as

	default. Only one of the configured trust configs can be made as default
trusthandler	Specify the implementation class name that implements com.sun.enterprise.security.trust.TrustHandler
trustedas	trusted-as with value 'intermediate' represents configuration information for incoming messages,if it has value 'destination' then configuration under trusted-entity is applied to outgoing messages.
ipaddress	Identifies the trusted host on the network.eg : 129.169.223.2
hostname	Identifies the trusted host on the network using domain names. eg: sun.com, cisco.com. All hosts from sun.com domain are trusted.
principal	Principal of the client
entityid	Identifier for trusted entity

- delete-trust-config [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] [--target target(Default server)] trust-id

  This command will delete the specified trust config from the target.

- list-trust-configs [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] [--trustid=trust-id] [target(Default server)]

  This command can be used to list the trust configs as well as trusted entities.

- create-trusted-entity [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] [--trustedas=intermediate|destination] [--hostname=host_name] [--principal=principal_name] [--target target(Default server)] --trustid=trust_id --ipaddress=ip_address [entity_id]

  This command will create a trusted entity for given trust config. See create-trust-config for option descriptions.

- delete-trusted-entity [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure|-s=true] [--user admin_user] [--passwordfile file_name] [--target target(Default server)] --trustid=trust_id entity_id

  This command will delete the trusted entity from the trust config.

4.1.7 Changes to existing CLI commands

- create-ssl

Existing usage text:

Usage: create-ssl --type [http-listener|iiop-listener|iiop-service] --certname cert_name [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure | -

s] [--user admin_user] [--passwordfile file_name] [--ssl2enabled=false] [--ssl2ciphers ssl2ciphers] [--ssl3enabled=true] [--ssl3tlsciphers ssl3tlsciphers] [--tlsenabled=true] [--tlsrollbackenabled=true] [--clientauthenabled=false] [--target target(Default server)] [listener_id]

create-ssl should accept an additional type 'sip-listener' to configure the SSL element in the selected sip-listener.

λ delete-ssl

Existing usage text:

Usage: delete-ssl --type [http-listener|iiop-listener|iiop-service] [--terse=false] [--echo=false] [--interactive=true] [--host localhost] [--port 4848|4849] [--secure | -s] [--user admin_user] [--passwordfile file_name] [--target target(Default server)] [listener_id]

delete-ssl should accept an additional type 'sip-listener' to delete the SSL element in the selected sip-listener.

λ create-domain

For any profile (developer, cluster) create-domain should create a domain with SIP configuration. This can be handled by using GlassFish profile management feature. See section 2.2

5 Packaging, Files, and Location

<Does this feature add new jar files or extend existing ones? Where are they located?>

Name	Purpose
${com.sun.aas.installRoot}/lib/admin-cli.jar	(Private) CLI framework classes modifications to plugin extra command descriptors.
${com.sun.aas.installRoot}/lib/comms-appserv-admin-cli.jar	(Private) Serialized CLI commands and other command classes
${com.sun.aas.installRoot}/lib/comms-appserv-admin.jar	(Private) Implementation classes and interfaces
${com.sun.aas.installRoot}/lib/appserv-rt.jar	(Private) Generated config beans and new DTD sun-domain_1_4.dtd

6 Quality

<Guidelines you wish to provide on how this feature is tested. Scenarios that must be tested.>

λ Test cases for new CLI commands

7 Documentation Requirements

<List the required documentation to support this product feature.>

λ Man pages for new CLI commands.

λ Administration Reference

8 Open Issues

The following issues are still under investigation and not finalized as to how we will be achieving each of these:

- domain.xml extension: The idea is to see if we can have a separate sip-domain.xml to accommodate all the new SIP elements. (CLOSED)
- Event/Event Listeners pluggability: Enhance GlassFish admin to load extra (new) admin Mbeans from a separate Mbeans descriptor file(s). (CLOSED)
- AMX: Currently its not possible to expose SailFin AMX Mbeans directly. AMX layer needs to be extended to plug-in SailFin AMX Mbeans.
- CLI pluggability: Enhance CLI framework to load multiple command descriptors. Proposed changes for same is under review. (CLOSED)
- Default SIP domain.xml generation (CLOSED)
- SSL for SIP listener: Its still not clear which of the ssl element attributes apply for sip-listener. This will affect the create-ssl, create-sip-listener commands. (CLOSED)
- LB config parameters: The DTD elements for LB config are yet to be finalized. (CLOSED)

9 References

- SailFin Architectural Overview Document : http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/sailfin_umbrella.pdf
- SIP container integration specification: http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/sailfin_fsd_integration.doc
- SailFin Converged Load Balancer specification: http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/ConvergedLB-FSD.pdf
- SailFin Security specification: http://wiki.glassfish.java.net/attach/FunctionalSpecsOnePagers/sailfin_fsd_security.odt
- AMX Support Extension Design document: http://wiki.glassfish.java.net/Wiki.jsp?page=Amx_extension.html
- GlassFish administration infrastructure specification <add link>