

# Sip Message Inspection Adapter One Pager

- 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. Bug/RFE Number(s):
    - 4.2.1 Bug/RFE Numbers from Issue Tracker
    - 4.2.2 Requirement Ids that are being addressed as a part of this proposal.
  - 4.3. In Scope:
  - 4.4. Out of Scope:
  - 4.5. Interfaces:
    - 4.5.1 Exported Interfaces
    - 4.5.2 Imported interfaces
    - 4.5.3 Other interfaces (Optional)
  - 4.6. Doc Impact:
  - 4.7. Admin/Config Impact:
    - 4.7.1 Configuration changes needed
    - 4.7.2 CLI / GUI impact if any
  - 4.8. HA Impact:
  - 4.9. I18N/L10N Impact:
  - 4.10. Packaging & Delivery:
    - 4.10.1 Binaries in which the code is delivered
  - 4.11. Security Impact:
  - 4.12. Compatibility Impact
  - 4.13. Dependencies:
    - 4.13.1 Changes required in GlassFish
    - 4.13.2 Third Party APIs
  - 4.14 Miscellaneous
- 5. Open Issues

- [6. Reference Documents:](#)
- [7. Schedule:](#)
- [7.1. Projected Availability:](#)

# 1. Introduction

## 1.1. Project/Component Working Name

SailFin 2.0 / Sip Message Inspection Adapters

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

Peter Strand: peter.strand@ericsson.com

## 1.3. Date of This Document:

Version	Date	Comments	Author
0.1	2009-02-27	First version	Peter Strand
0.2	2009-03-09	Added toc and changed name of config parameters	Peter Strand
1.0	2009-04-15	Updated w.r.t implemntation	Peter Strand

# 2. Project Summary

## 2.1. Project Description:

The feature SipMessageInspection is based on Java Logging and is intended as a tool to trace Sip messages. The trace output information is stored in the server.log file. This document describes some improvements to the existing feature SipMessageInspection. Note that there is also a related document "Enhancements to SMI and Access logging" Onepager describing some base improvements for the reporter framework for example

the base support to intercept outgoing requests and responses.

In addition to a improved default implementation a new feature is proposed: support of user configurable log adapter classes (runtime configurable). With this feature integrators/developers/support staff can balance the amount of log information with respect to the load of the system.

See below for a more detailed technical description.

## **2.2. Risks and Assumptions:**

N/A

# **3. Problem Summary**

## **3.1. Problem Area:**

## **3.2. Justification:**

- Modify Sip Message inspection with respect to the refined Reporter interface/framework.
- Provide a reasonable default implementation of Sip Message Inspection with more helpful information in less log records
- Add support for user configurable log adapters

# **4. Technical Description:**

## **4.1. Details:**

SipMessageInspection feature is intended as a mean of troubleshooting Sip traffic through SGCS. It is activated by setting the smi log domain to level FINE. When active it will log information like the text representation of the request to the SGCS log, one record per inspection point per layer that has the SipMessageInspection configured (with default stack config it is configured for GrizzlyNetworkManager and ApplicationDispatcher). With the current implementation there are some problems:

- The current Sip Message Reporter does not log outgoing requests/responses
- The reporter logs one record per un-winded call stack (This duplicates the log records produced without being helpful for troubleshooting)

The aim is to provide a better default implementation of Sip Message Inspection with

more helpful information in less log records. For a OPTIONS/200 sequence this means that 4 log records will be stored:

- GrizzlyNetworkManager: log the Incoming request
- ServletDispatcher: log the Incoming request
- ServletDispatcher: log the sent outgoing response
- GrizzlyNetworkManager: log the outgoing response

In addition to some general improvements a user configurable adapter feature is proposed (runtime configurable). This functionality lets integrators/developers/support staff to define their own configurable Adapters to provide more fine grained control over what is being logged. This way the amount of log information can be balanced with respect to the load of the system. The proposed interface is shown and further described in the Code example section.

## 4.1.1 Code examples

### 4.1.1.2 Interface

This is the proposed interface which allows integrators/developers/support staff to define their own configurable Adapters. Please note the similarities with Reporter proposed interface. The difference is that the methods return a String representation of the data which should be logged. If the implementing class does not want anything to be logged then it just returns the method with a null value. Also note that the Sip Message Inspection base functionality should do the actual logging (using a predefined log record including information about the context of the interception) . With the help of SSA interfaces for storing attributes in SipServletRequests/SipSessions it should for example be possible to collect information without logging and only log if an error response is received.

```
package org.jvnet.glassfish.comms.admin.reporter.smi;

public interface SmiLogMessageAdapter {
    String getLogMessageIncomingRequest(SipServletRequest req);
    String getLogMessageIncomingResponse(SipServletResponse resp);
    String getLogMessageOutgoingRequest(SipServletRequest req);
    String getLogMessageOutgoingResponse(SipServletResponse resp);

    String getLogMessagePostIncomingRequest(SipServletRequest req, I
    String getLogMessagePostIncomingResponse(SipServletResponse resp)
}
```

## 4.2. Bug/RFE Number(s):

## **4.2.1 Bug/RFE Numbers from Issue Tracker**

## **4.2.2 Requirement Ids that are being addressed as a part of this proposal.**

N/A

## **4.3. In Scope:**

N/A

## **4.4. Out of Scope:**

## **4.5. Interfaces:**

### **4.5.1 Exported Interfaces**

SmiLogMessageAdapter, see section 4.1.1.2 Interface.

### **4.5.2 Imported interfaces**

N/A

### **4.5.3 Other interfaces**

## **4.6. Doc Impact:**

- Developers Guide
- Admin Guide
- Trouble shooting Guide

## **4.7. Admin/Config Impact:**

### **4.7.1 Configuration changes needed**

#### **4.7.1.1 Adapter Configuration**

This configuration is implemented as "Additional properties" ( Admin GUI/ Configuration -> Sip Service ). This means that no changes are needed in dtd for

domain.xml.

One adapter class per SipMessageReporter should be configurable. By default there will be one Sip Message Reporter for LAYER interception on GrizzlyNetworkmanager and one for Servlet invocation on ApplicationDispatcher. The path and/or name of the Adapter class for layers is intended to be supplied by integrators/customers/developers.

- smiServletAdapter
- smiLayerAdapter
- smiNetworkManagerAdapter

By default these parameters are not set (Sip Message Reporters will then use its default implementation/logging). The values should be configurable in runtime. The format of these parameters:

- <package.classname> (specified class should be available with container class loader)
- <package.classname ; external class path> (specified class should be available at supplied external class root)

#### **4.7.1.2 Log level Configuration**

By setting the smi log level it is possible control the amount of records which is logged: ( e.g. server-config.log-service.module-log-levels.property.smi=FINE ).

FINE:

- SERVLET interception

FINER:

- SERVLET interception
- NetworkManager from/to cluster (i.e excluding Loadbalancer network hops)

FINEST:

- SERVLET interception
- NetworkManager including Loadbalancer network hops

#### **4.7.2 CLI / GUI impact if any**

No GUI changes are needed.

## **4.8. HA Impact:**

## **4.9. I18N/L10N Impact:**

## **4.10. Packaging & Delivery:**

### **4.10.1 Binaries in which the code are delivered**

The classes implementing the base functionality of Sip Message Inspection feature is placed under sailfin/integration (ends up in comms-appserv-rt.jar).

The adapter classes may be packaged by the user in a jar file and be placed under glassfish/lib or they may be provided from an external path.

## **4.11. Security Impact:**

N/A

## **4.12. Compatibility Impact**

The proposed changed feature reuses the smi log level ( e.g. server-config.log-service.module-log-levels.property.smi=FINE ).

## **4.13. Dependencies:**

N/A

### **4.13.1 Changes required in GlassFish**

No.

### **4.13.2 Third Party APIs**

N/A

## **4.14 Miscellaneous**

---

Will this component work with Ipv6 addresses	Yes
Will this component work with JDK 64bit	Yes
Will this component require configuration using a sun-specific deployment descriptor.If yes, please specify below that configuration elements needed	No

## 5. Open Issues

Issue No	Description	Comments	Resolution
----------	-------------	----------	------------

## 6. Reference Documents:

## 7. Schedule:

### 7.1. Projected Availability:

SGCS 2.0