PROGRESS®
ORBIX®

Actional Integration with Orbix

Version 3.3, SP11 March 2012

PROGRESS software
BUSINESS MAKING PROGRESS™

Progress Orbix v3.3.11

Third Party Acknowledgements: One or more products in the Progress Orbix v3.3.11 release includes third party components covered by licenses that require that the following documentation notices be provided:

Progress Orbix v3.3.11 incorporates OpenSSL/SSLeay v0.9.8.i technology from OpenSSL.org. Such Technology is subject to the following terms and conditions: LICENSE ISSUES

The OpenSSL toolkit stays under a dual license, i.e. both the conditions of the OpenSSL License and the original SSLeay license apply to the toolkit. See below for the actual license texts. Actually both licenses are BSD-style Open Source licenses. In case of any license issues related to OpenSSL please contact openssl-core@openssl.org.

OpenSSL License

Copyright (c) 1998-2008 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment:

"This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT ``AS IS'' AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

==

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License

Updated: 09-Mar-2012

Contents

List of Figures

CHAPTER 1

Orbix–Actional Integration

Orbix provides support for integration with Actional SOA management products. This chapter explains the main components and concepts used in this integration.

In this chapter

This chapter includes the following section:

Introduction

Overview

Actional is an SOA management product that provides operational and business visibility, policy-based security, and control of services and business processes in a heterogeneous runtime environment. This section explains the main concepts and components used in the Orbix–Actional integration.

Orbix and Actional

Integration between Orbix and Actional enables Orbix applications to be monitored by Actional SOA management tools. For example, you can use Actional to perform discovery, monitoring, auditing, and reporting on Orbix applications. You can also correlate and track all messages through your SOA network to perform dependency mapping and root cause analysis.

The Orbix–Actional integration is deployed on Orbix systems to enable reporting of management data back to the Actional server. The data reported back to Actional includes system administration metrics such as response time, fault location, auditing, and alerts based on policies and rules. The Orbix–Actional integration can be used with Orbix applications written in both Java and C++.

Actional SOA management

The main components in the Actional SOA management system are the Actional server, Actional agents, and Actional intermediaries.

The Actional server is the central engine that correlates data received from Actional agents and distributes policies. The Actional agent collects data about service traffic from an application server and applies policies. The Actional intermediary acts as a proxy that brokers interaction between Web service applications and systems built on them.

All Actional components are Java applications. The Actional server uses the Jetty application server by default, while its web console uses JSP and Adobe Flash.

Figure 1 shows a high-level overview of the main Actional components.

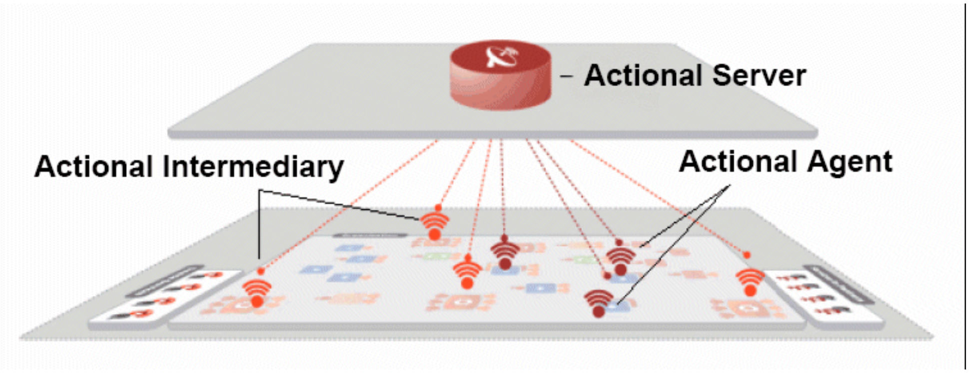


Figure 1: *High-Level Actional Overview*

Managed nodes

A node is defined as a system on the current network. A node with an Actional agent installed is referred to as an *instrumented node* or a *managed node*.

The managed node uses Actional's interceptor API to send monitoring data to the Actional agent. On any managed node, one Actional agent and one or more interceptors must be running.

Actional server

The Actional server is a central management server that manages nodes containing an Actional agent. The Actional server correlates the data it receives from each of its agents, and distributes policies to those agents. It enables an administrator to analyze service network data and create system-wide policies.

The Actional server hosts a database and pings Actional agents to obtain management data at configured time intervals. It analyzes the management data and displays it in a console—for example, the **Actional Management Server Administration Console**. This is a Web application deployed on Apache Tomcat, which provides runtime management and agent configuration. In addition, any alerts triggered at the Actional agent are sent immediately to the Actional server.

The default Actional server database is Apache Derby. Other supported databases include:

- PostgreSQL
- OpenEdge
- MSDE
- SQL Server
- Oracle
- DB2

By default, the Actional server uses port `4040` (for example, `http://*HostName*:4040/lgserver/`).

Actional agent

An Actional agent runs on each Orbix host that you wish to manage, and is used to provide instrumentation data back to the Actional server. The Actional agent includes two main components: an analyzer, and one or more interceptors. The analyser gathers and evaluates data such as records, statistics, and alerts. The interceptors collect data about service traffic from an application server, and apply policies to that traffic.

Actional agents are provisioned from the Actional server to establish initial contact and send configuration to the Actional agent. There is one Actional agent per managed node. By default, the Actional agent uses port `4041` (for example, `http://*HostName*:4041/lgagent/`).

Actional intermediary

An Actional intermediary is an in-network service broker that includes an integrated Actional agent. It serves as a proxy for Web service applications, providing features such as security, bridging, and activity tracking. The Actional intermediary supports application servers such as WebLogic, WebSphere, JBoss, and Oracle.

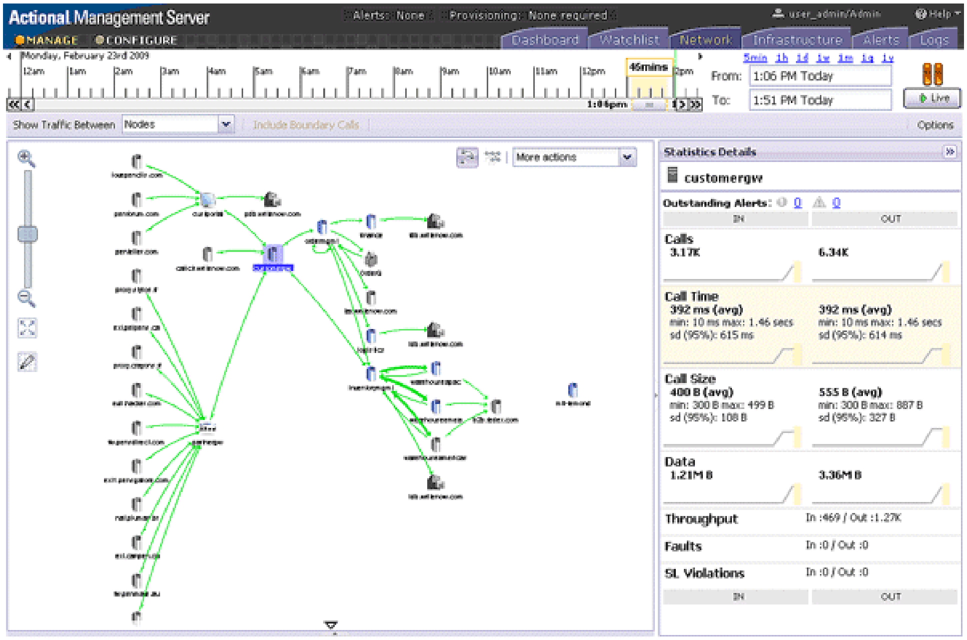
Actional agent interceptor SDK

The Actional Agent Interceptor Software Development Kit (SDK) is an Actional-specific API used to create custom interceptors. These can be used to send management instrumentation data from an application to the Actional agent.

Actional SOA management tools

In this guide, Actional is the general term used to describe the Actional SOA management system in which all data is stored and viewed. This simplifies the architecture of Actional for the sake of this discussion.

Figure 2 shows an example of the **Actional Management Server Administration Console**. Managed nodes are displayed as blue boxes, and unmanaged nodes are displayed as grey boxes. The green arrows indicate the message flow through various nodes. Clicking on each of the nodes shows more in-depth information regarding the response time, alerts and warnings, and so on.



Monday, February 23rd 2009

From: 1:06 PM Today

To: 1:51 PM Today

Show Traffic Between: Nodes

customergw

Outstanding Alerts: 0 0

	IN	OUT
Calls	3.17K	6.34K
Call Time	392 ms (avg) min: 10 ms max: 1.46 secs sd (95%): 615 ms	392 ms (avg) min: 10 ms max: 1.46 secs sd (95%): 614 ms
Call Size	400 B (avg) min: 300 B max: 499 B sd (95%): 108 B	555 B (avg) min: 300 B max: 887 B sd (95%): 327 B
Data	1.21M B	3.36M B

Throughput In :469 / Out :1.27K

Faults In :0 / Out :0

SL Violations In :0 / Out :0

Figure 2: *Actional Management Server Administration Console*

NGSO mapping

When you click and drill down in the Actional **Path Explorer** view, the organization of the information displayed is *Node–Group–Service–Operation* (NGSO). In Orbix, this translates to *Host–Module–Interface–Operation*. Table 1 shows the mapping from Actional to Orbix.

Table 1: *NGSO Mapping*

Actional	Orbix
Node	Host
Group	Module
Service	Interface
Operation	Operation

Further information

For detailed information on all Actional features, see the Actional product documentation.

Orbix–Actional Integration Architecture

Overview

This section shows a basic Actional architecture, simplified for the purposes of this discussion. It explains how Actional interceptors provide data to the Actional agent, and how the Actional server manifest is used to correlate the origin and business flow of a request.

It then shows the Orbix–Actional integration architecture, and explains how Orbix plug-ins and Orbix interceptors are used to configure integration with Actional.

Basic Actional architecture

Figure 3 shows a high-level overview of a basic Actional architecture from the perspective of a consumer and service provider.

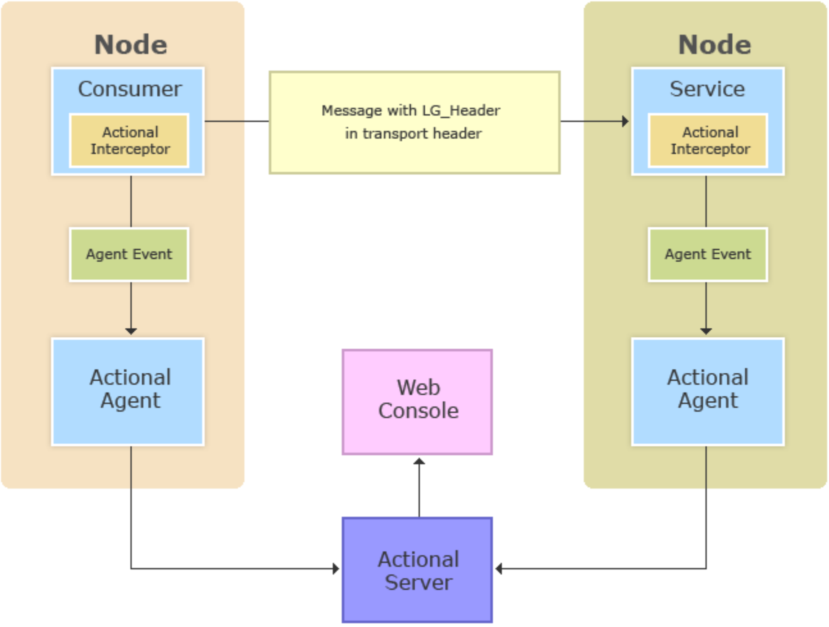


Figure 3: *Basic Actional Architecture*

In the interaction shown in Figure 3, the Actional interceptors sit in the flow between the application logic and the consumers and providers of other services. They intercept all inbound and outbound calls, and feed information about those calls to the Actional agent as asynchronous events.

The Actional agent is responsible for processing the event stream from the interceptors, computing and storing aggregate statistics, executing policies, and communicating with the Actional server.

The Actional server manifest (`LG_Header`) is a token that is sent in the transport header of the message to each participant in a call. This token identifies the origin and business flow of a request. For more details, see "Actional server manifest" on page 21.

Actional interceptors

Actional interceptors sit in the flow at the edge of an application, intercepting all incoming and outgoing messages. An Actional interceptor is designed as a lightweight component that imposes minimal overhead on the application (typically less than 100 microseconds per call).

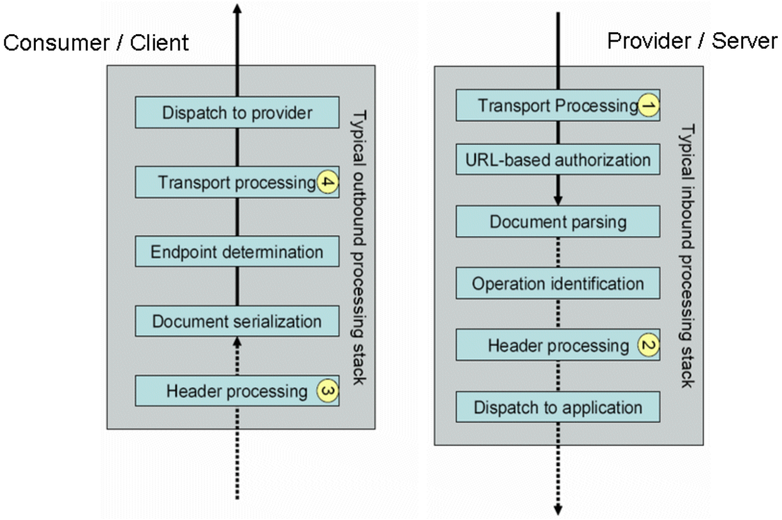


Figure 4: *Actional Interceptors*

The interceptor must perform the following tasks to gain the full functionality of the Actional server:

1. Extract an Actional server manifest (if any) from the incoming request document.
2. Insert an Actional server manifest into any outgoing request documents.
3. Transfer the interceptor context along the internal business flow, from the incoming interceptor, to any related outgoing interceptors.
4. Send the Actional agent an event for each incoming or outgoing document.

Actional server manifest

The Actional server sends an Actional server manifest (`LG_Header`) with a request document to provide information about the request's origin and the business flow that the request belongs to.

The Actional server manifest is used by the Actional server to correlate information it receives, from multiple agents, about interactions between different services. For this reason, the server manifest is sometimes referred to as a correlation ID.

The consumer and provider of the service must have an agreed mechanism (transport or protocol) for transferring the manifest. The following is an example `LG_Header`:

```
Interaction=CgJkcB+YlN0ZyBABdysAAA==;
Locus=ApM1eYBGBAR4LFJ1VvHOdg==;
Flow=CgJkcB+YlN0ZyBABdSsAAA==;
UpstreamOpID=FtfEJXM1nqJ0C995IBMkEQ==;
Path=7Qg2aVWCdwmP8gGebyLWYA==;
name=E_10-2-100-112-e0c7c3-110c80b4df0--7fdd-INITIATED;
CPTime=1171591682345;
FlowFields=MF1:1254;MF2:1589;
```

The main components in the server manifest are the `Interaction`, `Locus`, `Flow`, and `UpstreamOpID`. The other components are optional.

Orbix–Actional integration architecture

The Orbix–Actional integration is built using an extensible Orbix plug-in architecture. This means that Orbix–Actional integration can be enabled by adding a monitoring plug-in to your Orbix configuration. No code changes are necessary for Orbix client and server applications.

Figure 5 shows an overview of the Orbix–Actional integration architecture from an Orbix client-server perspective. This builds on the architecture shown in Figure 3, with the addition of Orbix monitoring plug-in. In Figure 5, the CORBA GIOP message also includes the `LG_Header` in a GIOP service context. A GIOP service context is a general mechanism for including out-of-band data in a GIOP request or reply message. Service contexts in GIOP are analogous to headers in other protocols such as HTTP.

Orbix interceptors

In the Orbix-Actional integration, an Orbix plug-in for Actional must also be added to your Orbix client and server processes. These plugins are loaded into the process via a config variable.

The Orbix monitoring plug-in is implemented using a propriety feature of Orbix 3.3.x called Filters and ServiceContext Handlers. This allows us to intercept the Request messages at various points in the lifecycle of sending/receiving of requests in the ORB. This enables high-level request processing to be performed.

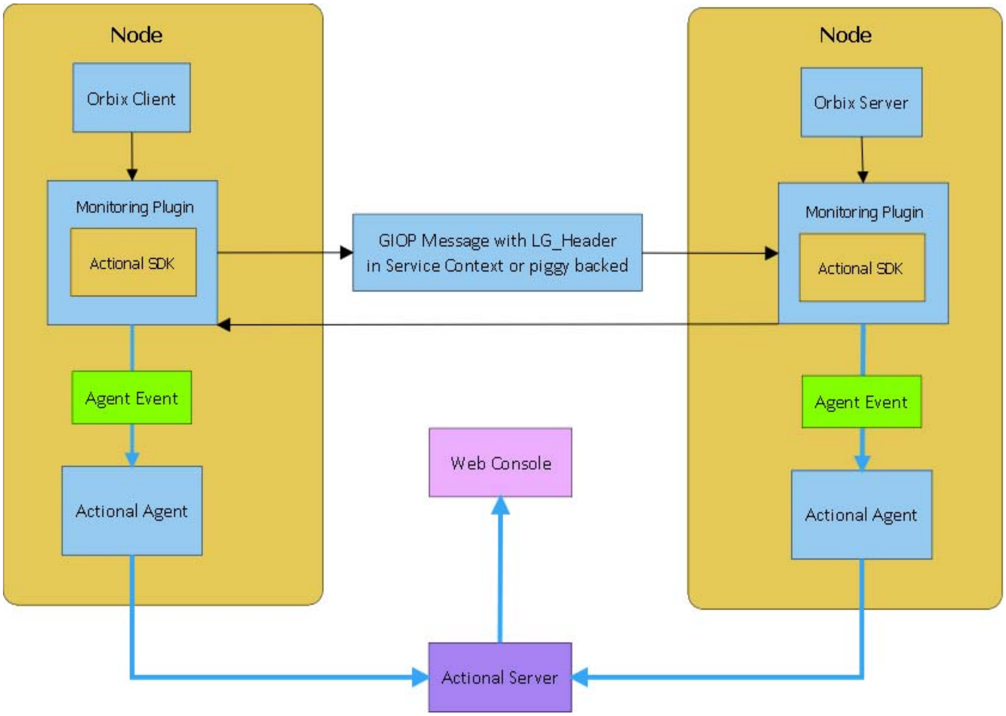


Figure 5: *Orbix-Actional Integration Architecture*

CHAPTER 2

Configuring Orbix for Actional Integration

This chapter explains the steps required to configure Orbix for integration with Actional SOA management products.

In this chapter

This chapter includes the following sections:

Configuring Orbix Java Applications

Overview

This section explains how to configure Orbix Java applications for integration with Actional. It shows some examples from the Orbix Actional integration demo:

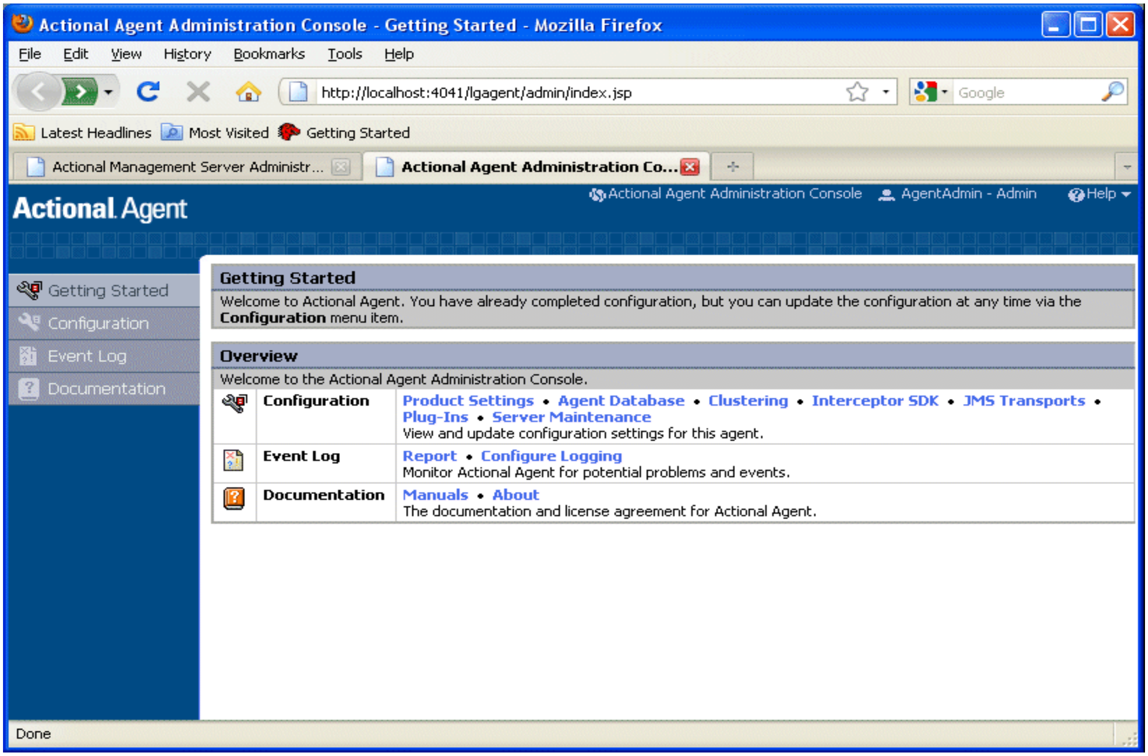
```
OrbixInstallDir/demos/common/monitoring
```

Update your Actional SDK

You must first update your Actional SDK JAR file as follows:

1. In the **Actional Agent Administration Console**, select **Getting Started|Interceptor SDK** (see Figure 6), and download the Windows `(.zip`) or UNIX (`.tar)` file. This includes the `actional-sdk.jar`, documentation, and samples.
2. Replace the existing `actional-sdk.jar` in the following location with the version that you downloaded:

   ```
   OrbixInstallDir/lib
   ```



Actional Agent Administration Console - Getting Started - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:4041/lgagent/admin/index.jsp
Google
Latest Headlines Most Visited Getting Started
Actional Management Server Administr...
Actional Agent Administration Co...
Actional Agent
Actional Agent Administration Console
AgentAdmin - Admin
Help
Getting Started
Configuration
Event Log
Documentation
Done

Getting Started

Welcome to Actional Agent. You have already completed configuration, but you can update the configuration at any time via the **Configuration** menu item.

Overview

Welcome to the Actional Agent Administration Console.

	Configuration	Product Settings • Agent Database • Clustering • Interceptor SDK • JMS Transports • Plug-Ins • Server Maintenance View and update configuration settings for this agent.
	Event Log	Report • Configure Logging Monitor Actional Agent for potential problems and events.
	Documentation	Manuals • About The documentation and license agreement for Actional Agent.

Figure 6: *Actional Agent Administration Console*

Configuring Actional Monitoring

There are two configuration variables that control the monitoring plugin:

- Specify the monitoring plug-in
- Specify the Uplink.cfg folder.

Specifying the monitoring plug-in name

To tell the Orbix Java runtime to load the java monitoring plugin, add the following config variable to the OrbixWeb scope in your orbixweb3.cfg configuration file:

```
IT_ORB_INITIALISORS =
   "IE.Iona.OrbixWeb.Monitoring.MonitoringPI";
```

Specify the Uplink.cfg folder

If you install the Actional Agent's uplink.cfg into a non-default location this config variable may come in handy as it will set the com.actional.lg.interceptor.config system property. Else you need to add the property as -Dcom.actional.lg.interceptor.config=<path-to-uplink-cfg> to the java command.

```
IT_ACTIONAL_LG_INTERCEPTOR = <path-to-uplink-cfg>
```

The OrbixWeb jar does not contain a manifest to pull in the monitoring.jar, so any java applications will need to modify their classpaths to ensure that both the monitoring.jar and actional-sdk.jar are present on the classpath.

Configuring Orbix C++ applications

Overview

This section explains how to configure Orbix C++ application for integration with Actional. It shows some examples from the Orbix Actional integration demo:

```
OrbixInstallDir/demos/monitoring
```

Setting your environment

No changes are necessary if the Actional `Uplink.cfg` configuration file is located in its default path:

UNIX	`/var/opt/actional/LG.Interceptor`
Windows	`%systemroot%\system32\LG.Interceptor`

On a 64-bit windows system, if the agent runs with a 32-bit java VM, then the default location is:

`%systemroot%\SysWow64\LG.Interceptor`, otherwise it is `%systemroot\system32\LG.Interceptor.`

The `Uplink.cfg` file is responsible for communication between the Actional interceptors and the analyzer in the Actional agent.

If the `Uplink.cfg` is not located in its default path, you must specify the path to this file as follows:

UNIX	`export LG_INTERCEPTORCONFIG=`*`PathToFile`*
Windows	`set LG_INTERCEPTORCONFIG=`*`PathToFile`*

Configuring the Orbix monitoring plug-in

You can configure the monitoring plug-in by editing the settings in your application configuration scope in your Orbix configuration file. This includes the following steps:

- Specify the monitoring plug-in
- Specify the monitoring log filter

Specifying the monitoring plug-in

You can tell Orbix to load the monitoring plugin by adding the following config variable to the Orbix scope:`orbix3.cfg`.

```
Orbix.IT_ORB_INITIALISORS = "it_monitoring";
```

You can also set the configuration variable in the environment, for example:

- "Windows: set IT_ORB_INITIALISORS=it_monitoring & client.exe
- "Unix: IT_ORB_INITIALISORS=it_monitoring ./client

Specifying the monitoring log filter

You can specify the following config variable and it will output various diagnostic information about the monitoring plugin, if you need more verbose information.:

```
Orbix.IT_LOGGING_FILTERS="monitoring";
```

This will create a logfile called monitoring.log in the current folder with all the logging from the plugin.

Troubleshooting Orbix

Overview

This section provides some tips to help troubleshoot your Orbix integration with Actional.

Ensure Actional Plugin is loaded

To verify that the Orbix monitoring plug-in is loaded and participating correctly.

C++

You can specify the IT_LOGGING_FILTERS config variable to ensure that the logging from the monitoring plug-in being outputted to the log file:

```
[17:42:16.080, pid: 20765 tid: -1424223984]: Client Interaction:
   url: Plugin/Tester
    Operation: find_name
    Group: Plugin
    Service: Tester
    HostName: jewel]
```

Also you can specify the environment variable IT_SHLIB_VERBOSE=1 in your environment, this will show a more verbose output of what the plug-in loader in Orbix is doing, or if it's loading the monitoring plugin.

```
[IT_Shlib_Manager, thread 1] loading it_monitoring.
[IT_Shlib_Manager, thread 1] Attempting to load library:
   libit_monitoring.3.3.gcc411.so.1
[IT_Shlib_Manager, thread 1] Located shlib:
   /vob/orbix/targets_orbix//lib/libit_monitoring.3.3.gcc411.so.
   1
[IT_Shlib_Manager, thread 1] About to load shlib:
   /vob/orbix/targets_orbix//lib/libit_monitoring.3.3.gcc411.so.
   1
[IT_Shlib_Manager, thread 1]
   /vob/orbix/targets_orbix//lib/libit_monitoring.3.3.gcc411.so.
   1 seems to be compatible with the shared libraries already
   loaded in this process.
[IT_Shlib_Manager, thread 1] Loading plugin:
   /vob/orbix/targets_orbix//lib/libit_monitoring.3.3.gcc411.so.
   1
```

Java

You can turn on diagnostics as normal with the setDiagnostics config variable, a level of 128 or higher will output the monitoring plug-in's information.

```
[MonitoringGeneric:group: Monitoring Service: Server2
   RepositoryID: Monitoring/Server2]
[MonitoringGeneric:ClientInteraction.setSelfAddr(10.2.2.141)]
[MonitoringGeneric:ClientInteraction.setPeerAddr(jewel)]
[MonitoringGeneric:ClientInteraction.setUrl(Monitoring/Server2)]
[MonitoringGeneric:ClientInteraction.setGroupName(Monitoring)]
[MonitoringGeneric:ClientInteraction.setService(Server2)]
[MonitoringGeneric:ClientInteraction.setOpName(get_command)]
[MonitoringGeneric:ClientInteraction.requestAnalyzed()]
```

CHAPTER 3

Configuring Actional for Orbix Integration

This chapter gives some basic guidelines on setting up Actional to run the Orbix Actional integration demo.

In this chapter

This chapter includes the following sections:

Prerequisites

Overview

This section describes prerequisites for integration between Actional SOA management products and Orbix.

Actional products

The following Actional products should be installed:

- Actional Management Server 8.0 (Actional server)
- Actional Flex Point 8.0 (Actional agent/intermediary)

Alternatively, the following Actional products can be installed separately:

- Actional Point of Operational Visibility 8.0 (Actional agent)
- Actional Client Security Enforcement 8.0 (Actional intermediary)

Actional agents

You must ensure that Actional agents are set up on each Orbix host node that you wish to manage. The provisioning of Actional agents is performed using the Actional server. For some basic details, see “Configuring Actional for Orbix Integration” on page 35.

For full details on how to set up Actional agents on managed nodes, see the Actional product documentation.

Further information

For information on installing Actional products, and the full range of platform and database versions supported by Actional, see the Actional product documentation.

This Orbix integration with Actional supports the full range of operating systems and compilers supported by Orbix. For more details, see the Orbix Installation Guide.

Configuring Actional

Overview

This section provides some basic configuration guidelines on Actional agent and server configuration.

Actional agent configuration

No specific Actional agent configuration settings are required for integration with Orbix. For example, for the purposes of the Orbix–Actional integration demos, the Actional agent can be started with the default configuration settings.

Actional server configuration

The following sample configuration steps describe how to set up the Actional server to run an simple Orbix–Actional demo:

1. Install the Actional server with typical installation options, and select the Apache Derby database.
2. Specify the following URL in your browser:

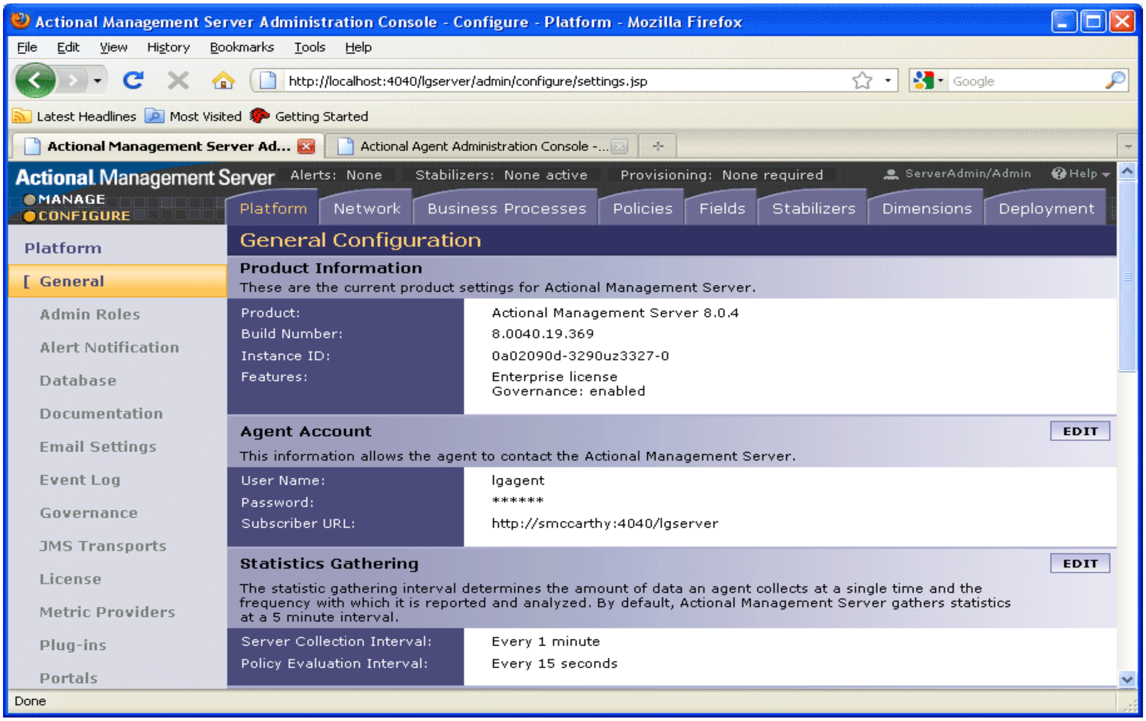
   ```
   http://localhost:4040/lgserver
   ```

3. If this is a new installation click **Start**, and follow the new Actional server setup steps.

   Otherwise, if the Actional server is already installed, perform the following steps:

   i. In the Actional console Web interface, select the **Configure** radio button in the top left of the screen.

   ii. Select the **Platform** tab. This displays the general configuration settings, as shown in Figure 7.



General Configuration

Product Information

These are the current product settings for Actional Management Server.

Product: Actional Management Server 8.0.4
Build Number: 8.0040.19.369
Instance ID: 0a02090d-3290uz3327-0
Features: Enterprise license
Governance: enabled

Agent Account

This information allows the agent to contact the Actional Management Server.

User Name: lgagent
Password: ******
Subscriber URL: http://smccarthy:4040/lgserver

Statistics Gathering

The statistic gathering interval determines the amount of data an agent collects at a single time and the frequency with which it is reported and analyzed. By default, Actional Management Server gathers statistics at a 5 minute interval.

Server Collection Interval: Every 1 minute
Policy Evaluation Interval: Every 15 seconds

Figure 7: *Actional Server Configuration Settings*

Creating a managed node

To create a managed node for a simple Orbix demo, perform the following steps:

1. In the Actional **Configure** view menu bar, open the **Network** tab. This displays the **Network Nodes**.
2. Select **Add**. This displays **Node Creation / Managing Agents**.
3. Click **Managed Node**.

Configuring a new node

To configure a managed node for the demo, perform the following steps in the wizard:

Step 1: New Node - Identification

1. Specify the **Name** as `agent1`.
2. Specify the **Display icon** as `Auto Discover`.
3. Click **Next**.

Step 2: New Node - Management

1. Specify the **Transport** as `HTTP/S`.
2. Supply your Actional agent user name and password.
3. Ensure that **Override Agent Database** is checked.
4. Click **Next**.

Step 3: New Node - Agents

1. Specify the following URL:

   http://*HostName*:4041/lgagent

   You can specify a host name or an IP address in this URL.
2. Click **Add**. The agent URL is added.
3. Click **Next**.

Step 4: New Node - Endpoints

1. For **Endpoints**, add the hostname, fully qualified hostname, or IP address.
2. Click **Next**.

Step 5: New Node - Filters

1. Do not specify any filters for the demo.
2. Click **Next**.

Step 6: New Node - Trust Zone

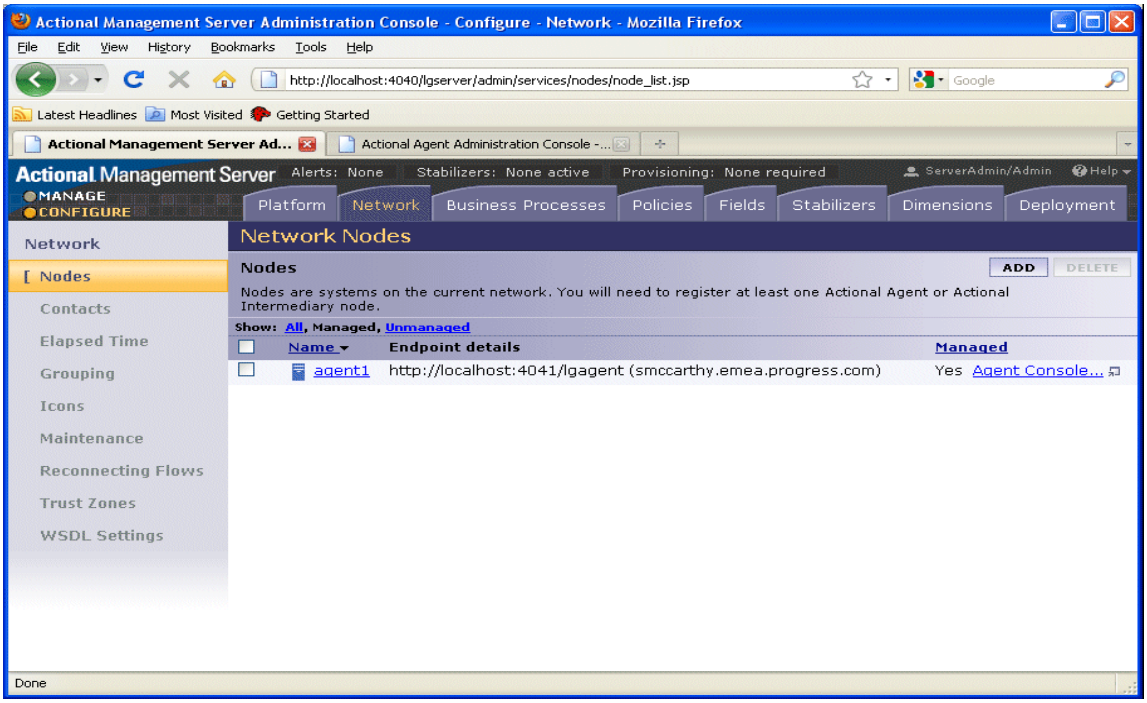
1. Do not specify a trust zone for the demo.
2. Click **Finish**.

The newly created managed node now needs to be provisioned.

Provisioning a new node

To provision the new node to bring it under management, perform the following steps:

1. Select the **Configure** radio button at the top left of the screen.
2. Select the **Deployment** tab from the **Configure** menu bar.
3. The **Provisioning** page is displayed, and `agent1` is listed as not provisioned.
4. Select the `agent1` check box.
5. Click **Provision**. This displays a message when complete: `Successfully provisioned`.
6. Click the **Manage** radio button at the top left of the screen. You should see `agent1` added to the **Network** view as shown in Figure 8.



Nodes are systems on the current network. You will need to register at least one Actional Agent or Actional Intermediary node.

Show: All, Managed, Unmanaged

□	Name	Endpoint details	Managed
□	agent1	http://localhost:4041/lgagent (smccarthy.emea.progress.com)	Yes Agent Console...

Figure 8: *Actional Server Provisioned Node*

Further information

For more details on setting up and running Actional SOA management tools, see the Actional product documentation.

Troubleshooting Actional

Overview

This section provides some tips to help troubleshoot your Actional integration with Orbix.

Setting default polling

For demonstration purposes, to update the display in your Actional server console more frequently, you can set the default polling to a shorter time span as follows:

1. Select the **Configure** radio button at the top left of the screen.
2. Select the **Platform** tab from the **Configure** menu bar.
3. In **Statistics Gathering** on the right, select **EDIT**.
4. Set the **Server Collection Interval** to 1 minute by using the drop down list.
5. Set the **Policy Evaluation Interval** to 15 seconds.

Ensuring events are reported to the Actional Agent

To ensure that Orbix monitoring events are being reported to your Actional agent, perform the following steps:

1. Ensure your Actional agent is running, and added as a managed node in your Actional server.
2. Verify that the agent generated the `Uplink.cfg` file in the directory specified during installation. If this file was not specified during the installation, it should be in the following default path (which should have write permission):

UNIX	`/var/opt/actional/LG.Interceptor`
Windows	`%systemroot%\system32\LG.Interceptor`

3. Open your Actional agent console and login:

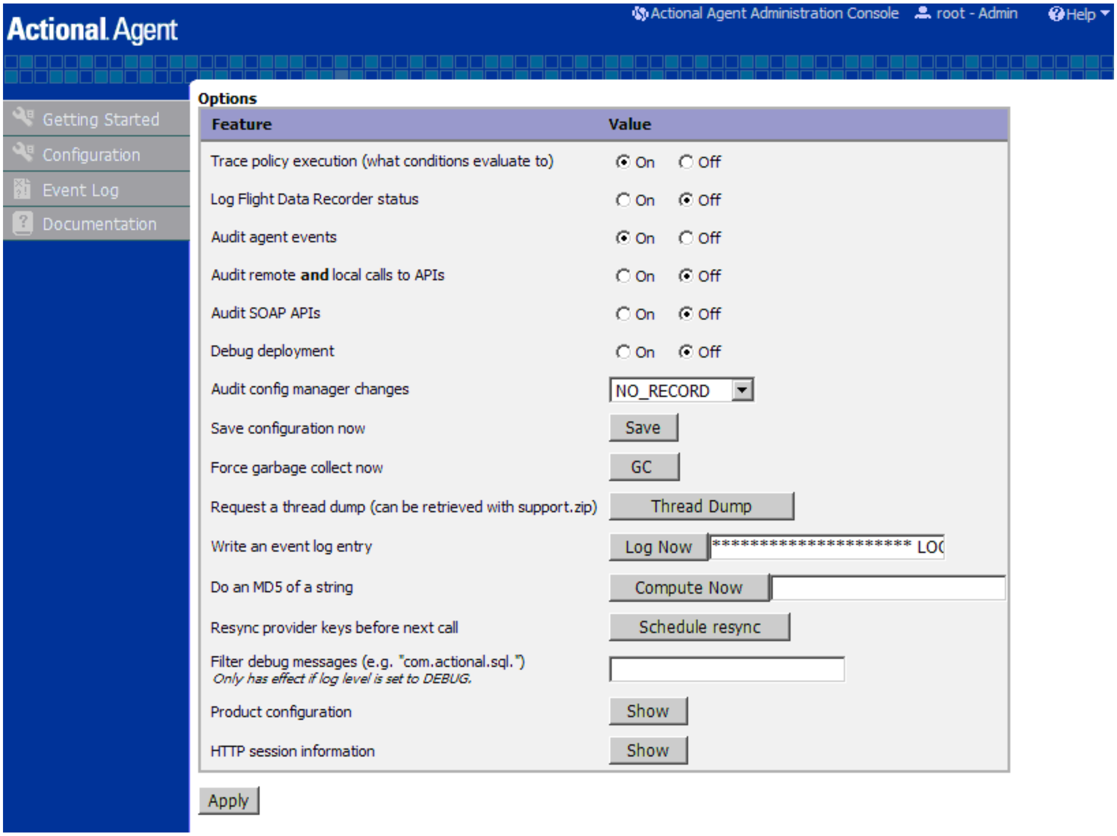
```
http://AgentHostName:Port/lgagent/
```

4. Specify the following URL to display the **Options** page shown in Figure 9:

   `http://`*`AgentHostName`*`:`*`Port`*`/lgagent/admin/options.js`

5. For **Audit agent events**, Click **On**.

6. Click **Apply**.



Options

Feature	Value
Trace policy execution (what conditions evaluate to)	● On ○ Off
Log Flight Data Recorder status	○ On ● Off
Audit agent events	● On ○ Off
Audit remote **and** local calls to APIs	○ On ● Off
Audit SOAP APIs	○ On ● Off
Debug deployment	○ On ● Off
Audit config manager changes	NO_RECORD
Save configuration now	Save
Force garbage collect now	GC
Request a thread dump (can be retrieved with support.zip)	Thread Dump
Write an event log entry	Log Now ********************** LO(
Do an MD5 of a string	Compute Now
Resync provider keys before next call	Schedule resync
Filter debug messages (e.g. "com.actional.sql.") *Only has effect if log level is set to DEBUG.*	
Product configuration	Show
HTTP session information	Show

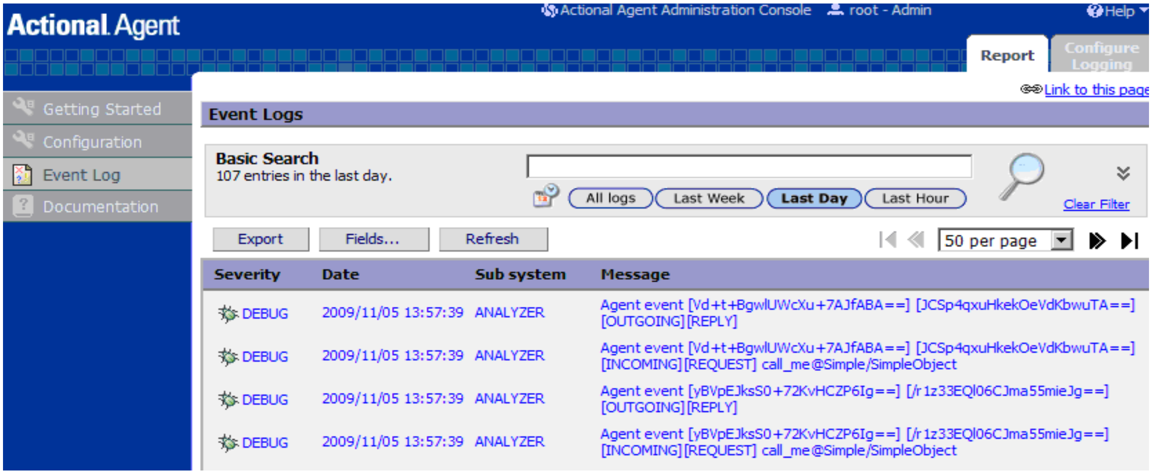
Apply

Figure 9: *Actional Agent Options*

Note: These settings are not persistent, and are reset when the Actional agent is restarted.

Viewing agent events

When **Audit agent events** is turned on, all external events coming from the Orbix monitoring plug-in can be reviewed in the Actional agent **Event Logs**, shown in Figure 10.



Event Logs

Basic Search
107 entries in the last day.

Severity	Date	Sub system	Message
DEBUG	2009/11/05 13:57:39	ANALYZER	Agent event [Vd+t+BgwlUWcXu+7AJfABA==] [JCSp4qxuHkekOeVdKbwuTA==] [OUTGOING][REPLY]
DEBUG	2009/11/05 13:57:39	ANALYZER	Agent event [Vd+t+BgwlUWcXu+7AJfABA==] [JCSp4qxuHkekOeVdKbwuTA==] [INCOMING][REQUEST] call_me@Simple/SimpleObject
DEBUG	2009/11/05 13:57:39	ANALYZER	Agent event [yBVpEJksS0+72KvHCZP6Ig==] [/r1z33EQl06CJma55mieJg==] [OUTGOING][REPLY]
DEBUG	2009/11/05 13:57:39	ANALYZER	Agent event [yBVpEJksS0+72KvHCZP6Ig==] [/r1z33EQl06CJma55mieJg==] [INCOMING][REQUEST] call_me@Simple/SimpleObject

Figure 10: *Actional Agent Event Logs*

Figure 10 shows `INCOMING`, `OUTGOING`, `REQUEST`, and `REPLY` events reported from the monitoring plug-in. If these events are not reported, the path for the `uplink.cfg` may be incorrect, and the monitoring plug-in can not find the agent.

C++ applications

For C++ applications, verify that the `LG_INTERCEPTORCONFIG` environment variable is set correctly, and points to the directory where the agent has written the `uplink.cfg` file.

Java applications

For Java applications, verify that the `com.actional.lg.interceptor.config` property is passed on to the application correctly, and points to the directory where the agent has written the `uplink.cfg` file.

When incoming monitoring events are arriving at the agent, and the agent is configured correctly, you should see the calls displayed in the Actional server console **Network** view, as shown in Chapter 2.

Further information

For any problems with Actional agent configuration, please refer to the Actional product documentation.

CHAPTER 4

Managing Orbix Applications in Actional

This chapter shows examples of managing a simple Orbix application and Orbix domain services in Actional SOA management tools.

In this chapter

This chapter includes the following sections:

Monitoring Orbix Applications

Overview

When your Orbix applications are configured for integration with Actional, they can be monitored using the Actional SOA management tools. No code changes are required for monitoring of Orbix applications.

For example, when you run the simple Orbix `actional_demo`, the **Actional Management Server Administration Console** displays the managed node that the demo is running on. Invocations are displayed as arrows flowing to and from managed components.

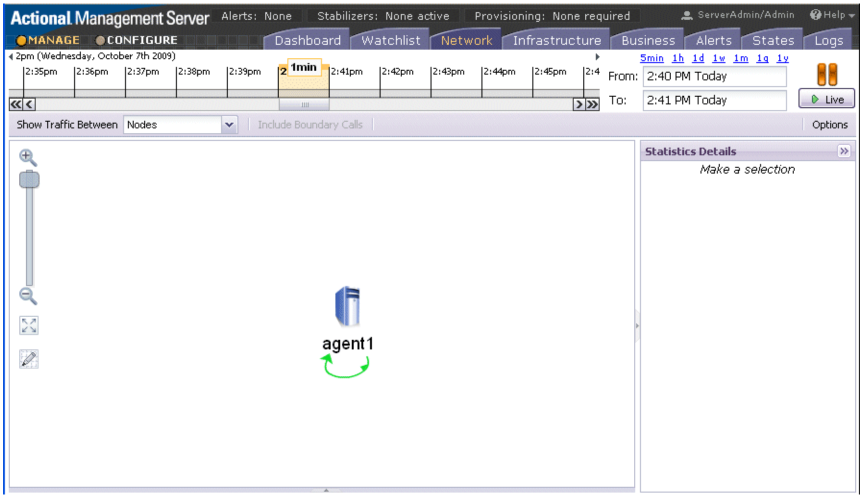
The Orbix `monitoring demo` illustrates the simple use of the ORB monitoring plug-in to report calls made between Orbix clients and servers to Actional. This demo is similar to `demos/corba/orb/simple`, and shows how to configure visibility of your application in Actional. For details on how to run this demo, see the `README` text files in the following directory:

```
OrbixInstallDir/demos/monitoring
```

Network view

The Actional network view displays the traffic between various components in your network environment. These include nodes, packages, services and operations.

Figure 11 shows the running Orbix `actional_demo` displayed in the **Network** tab of the **Actional Management Server Administration Console**. In this simple demo, the **Network** tab displays the Actional agent on the Orbix managed node that the demo is running on. This agent reports the monitoring data back to the Actional server. The single invocation is displayed as a green arrow flowing from the node and back to itself. In more complex examples with multiple nodes, the arrows flow between nodes.



From: 2:40 PM Today

To: 2:41 PM Today

Show Traffic Between: Nodes

Figure 11: *Actional Server Network View*

By default, the **Network** view shows traffic between nodes. There is only one node in this case. You can also select to show traffic between packages in the top left of the screen. Figure 12 shows the traffic between the Orbix client and server packages.

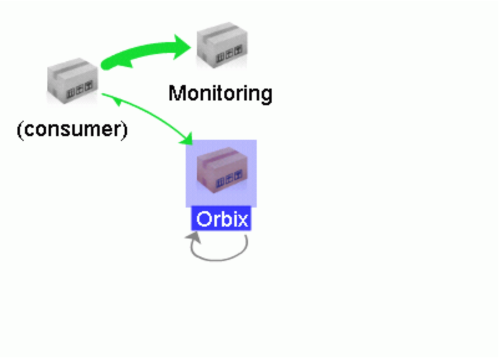


Figure 12: *Traffic Between Packages*

Path Explorer

Figure 13 shows the Orbix `actional_demo` displayed in the **Path Explorer** view of the **Actional Management Server Administration Console**.

To view this screen, double click on the managed node shown in Figure 11. Alternatively, click the **Display Path Explorer** button at the top right of the **Network** view.

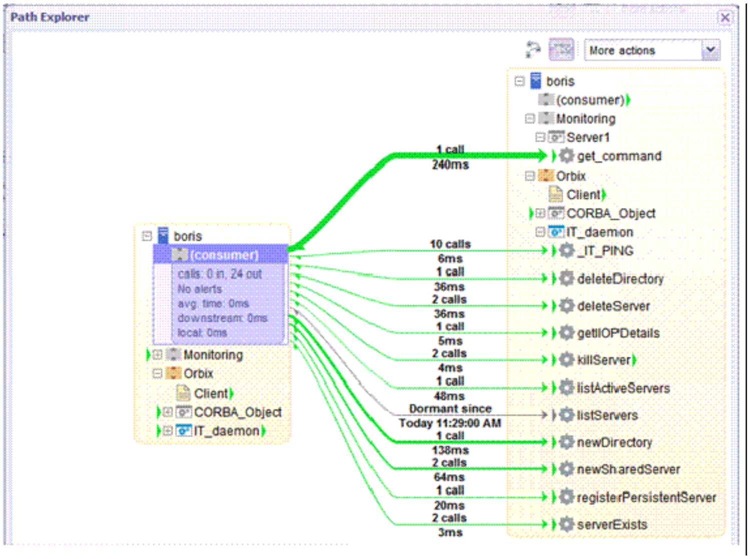


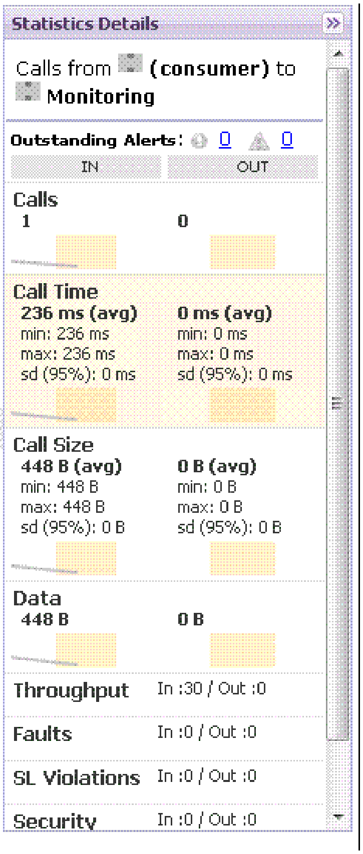
Figure 13: *Actional Server Path Explorer*

The **Path Explorer** view displays the relationships between different components in more detail. For example, you can view the call chain between services and consumers. Summary statistics are also displayed for the selected component.

Statistics details

The **Statistics Details** pane on the right displays statistics gathered by the selected component. These include the number of incoming and outgoing calls, call time, call size, and so on. Alerts, faults and violations are also displayed.

For example, Figure 14 shows the **Statistics Details** displayed for a client request to a server, when the operation is selected in the **Path Explorer**.



Statistics Details

Calls from **(consumer)** to **Monitoring**

Outstanding Alerts: 0 0

	IN	OUT
Calls	**1**	**0**
Call Time	**236 ms (avg)** min: 236 ms max: 236 ms sd (95%): 0 ms	**0 ms (avg)** min: 0 ms max: 0 ms sd (95%): 0 ms
Call Size	**448 B (avg)** min: 448 B max: 448 B sd (95%): 0 B	**0 B (avg)** min: 0 B max: 0 B sd (95%): 0 B
Data	**448 B**	**0 B**

Throughput In :30 / Out :0

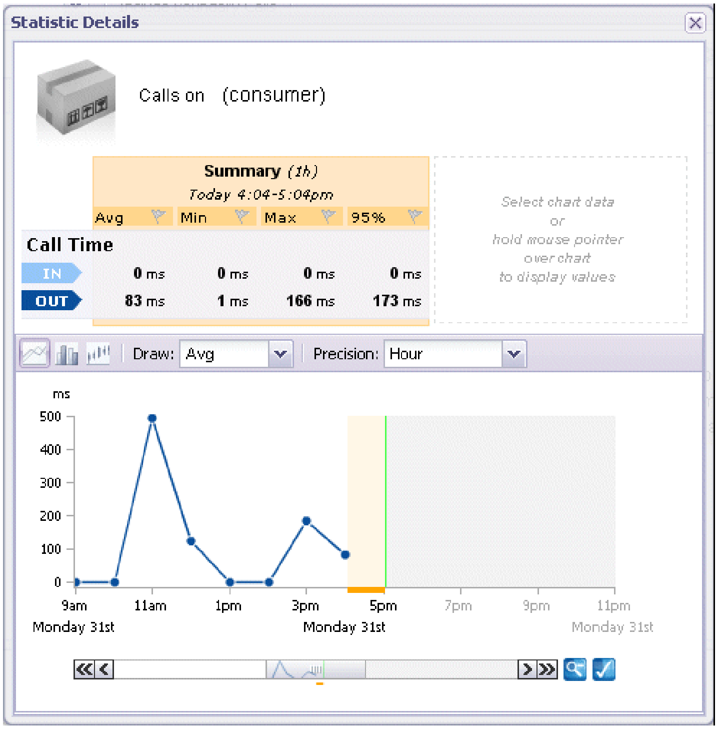
Faults In :0 / Out :0

SL Violations In :0 / Out :0

Security In :0 / Out :0

Figure 14: *Actional Server Statistics Details*

Double clicking on a particular statistic in this view (for example, **Call Size**) displays a summary chart. For example, Figure 15 shows a **Call Time** summary chart for the consumer.



Statistic Details

Calls on (consumer)

Summary *(1h)*
Today 4:04-5:04pm

Call Time	Avg	Min	Max	95%
IN	0 ms	0 ms	0 ms	0 ms
OUT	83 ms	1 ms	166 ms	173 ms

Select chart data or hold mouse pointer over chart to display values

Draw: Avg

Precision: Hour

Figure 15: *Actional Server Statistics Chart*

Server manifest

The Actional server manifest (`LG_Header`) is a unique ID used by the Actional server to correlate information it receives from agents about interactions between different applications. For example, when you run the client application in the Orbix `actional_demo`, the following `LG_Header` is output on the command line:

```
Interaction=CgIEAUD6LU2sLiQBBwAAAA==;
Locus=4/LcwgqvldfxotEoegsSGg==;
Flow=CgIEAUD6LU2sLiQBBgAAAA==;
UpstreamOpID=xPnAfuwlTEV7QGYoGRBgYA==;
CallerAddress=10.2.4.1;
```

Further information

For detailed information on using Actional SOA management tools, see the Actional product documentation.

Auditing Orbix Applications

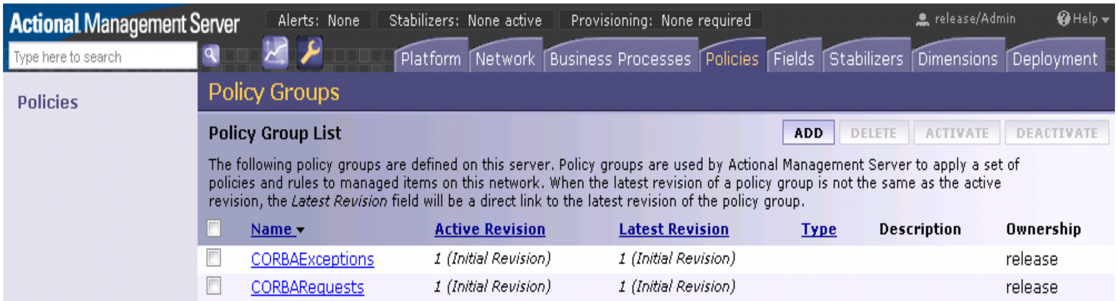
Overview

This section shows some simple examples of auditing the Orbix `actional_demo` and Orbix domain services.

Actional policy groups

Policy groups are used by Actional server to apply a set of policies and rules to managed items on your network. Policies and rules can be used to raise alerts on certain failure reasons. For example, when an Orbix operation takes too long to return, or when a specified IDL exception or fault is raised.

Figure 16 shows some example policy groups that have be defined in the **Policies** view. See configuring message fields section, for more detailed example on how to setup Policy Groups.



Policy Groups

Policy Group List

The following policy groups are defined on this server. Policy groups are used by Actional Management Server to apply a set of policies and rules to managed items on this network. When the latest revision of a policy group is not the same as the active revision, the *Latest Revision* field will be a direct link to the latest revision of the policy group.

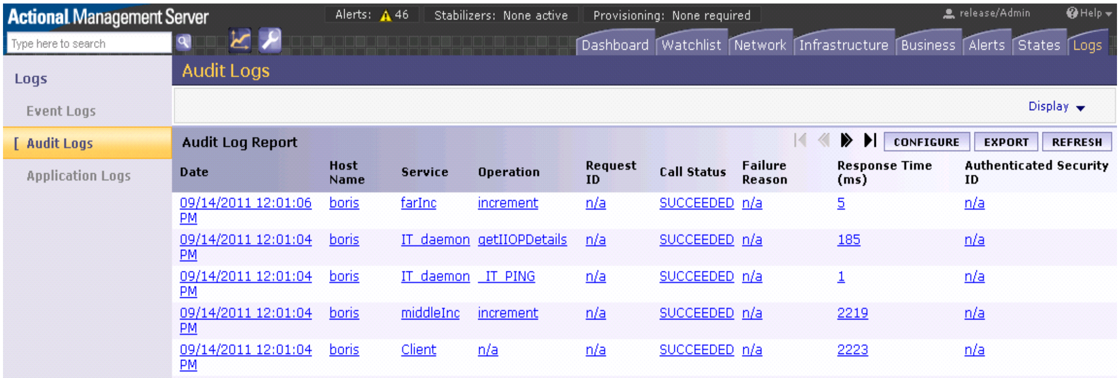
□	Name	Active Revision	Latest Revision	Type	Description	Ownership
□	CORBAExceptions	*1 (Initial Revision)*	*1 (Initial Revision)*			release
□	CORBARequests	*1 (Initial Revision)*	*1 (Initial Revision)*			release

Figure 16: *Actional Policy Groups*

Viewing audit logs

When you have defined policies for your network, you can use them to audit and monitor alerts on certain failure reasons (for example, when a specified IDL exception or fault is raised).

Figure 17 shows some example audit logs for the Orbix application in the **Logs** view.



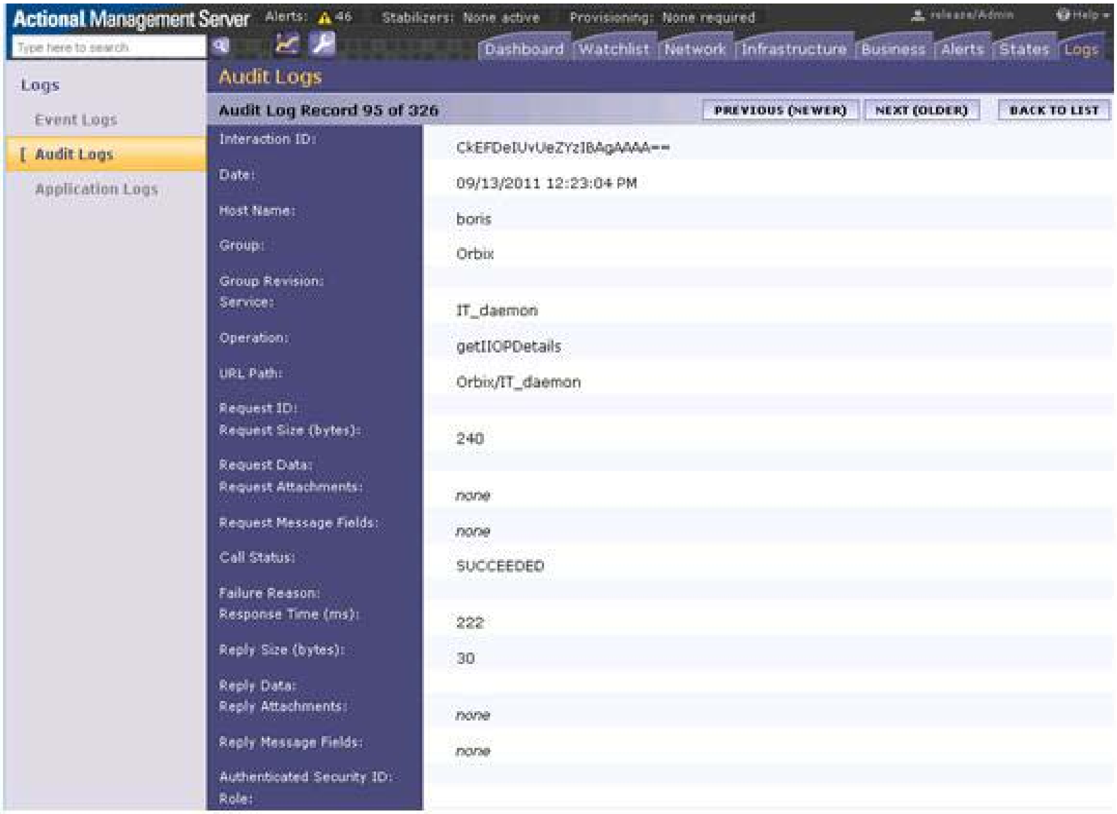
Audit Logs

Audit Log Report

Date	Host Name	Service	Operation	Request ID	Call Status	Failure Reason	Response Time (ms)	Authenticated Security ID
09/14/2011 12:01:06 PM	boris	farInc	increment	n/a	SUCCEEDED	n/a	5	n/a
09/14/2011 12:01:04 PM	boris	IT_daemon	getIIOPDetails	n/a	SUCCEEDED	n/a	185	n/a
09/14/2011 12:01:04 PM	boris	IT_daemon	_IT_PING	n/a	SUCCEEDED	n/a	1	n/a
09/14/2011 12:01:04 PM	boris	middleInc	increment	n/a	SUCCEEDED	n/a	2219	n/a
09/14/2011 12:01:04 PM	boris	Client	n/a	n/a	SUCCEEDED	n/a	2223	n/a

Figure 17: *Audit Logs from instrumented application*

Figure 18 shows an example audit log record displayed on clicking on an entry for the Orbix application in Figure 17.



Audit Logs

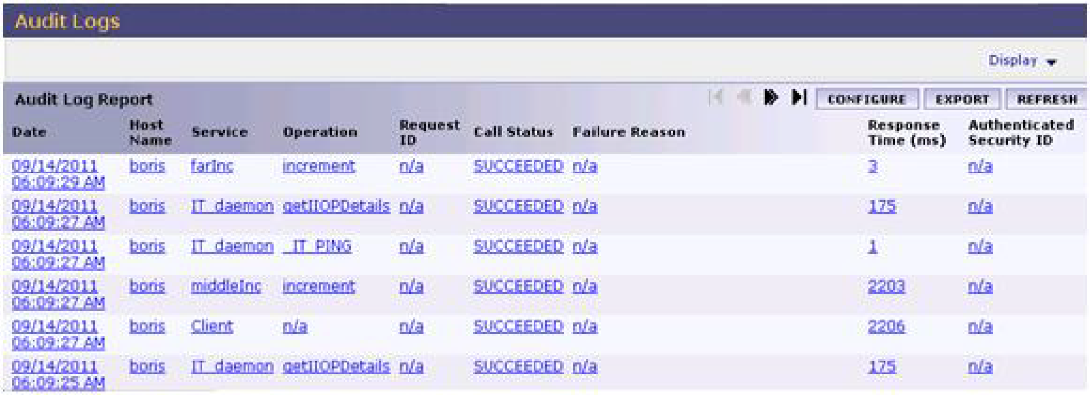
Audit Log Record 95 of 326

Interaction ID:	CkEFDeIUvUeZYzIBAgAAAA==
Date:	09/13/2011 12:23:04 PM
Host Name:	boris
Group:	Orbix
Group Revision:	
Service:	IT_daemon
Operation:	getIIOPDetails
URL Path:	Orbix/IT_daemon
Request ID:	
Request Size (bytes):	240
Request Data:	
Request Attachments:	*none*
Request Message Fields:	*none*
Call Status:	SUCCEEDED
Failure Reason:	
Response Time (ms):	222
Reply Size (bytes):	30
Reply Data:	
Reply Attachments:	*none*
Reply Message Fields:	*none*
Authenticated Security ID:	
Role:	

Figure 18: *Orbix Daemon call getIIOPDetails Audit Log Record*

The **Interaction ID** displayed at the top of the screen is used by the Actional server to correlate information it receives, from multiple agents, about interactions between different services.

Figure 19 shows some example audit logs for Orbix configuration domain services in the **Logs** view. The Orbix service displayed in this example is the Orbix node daemon.

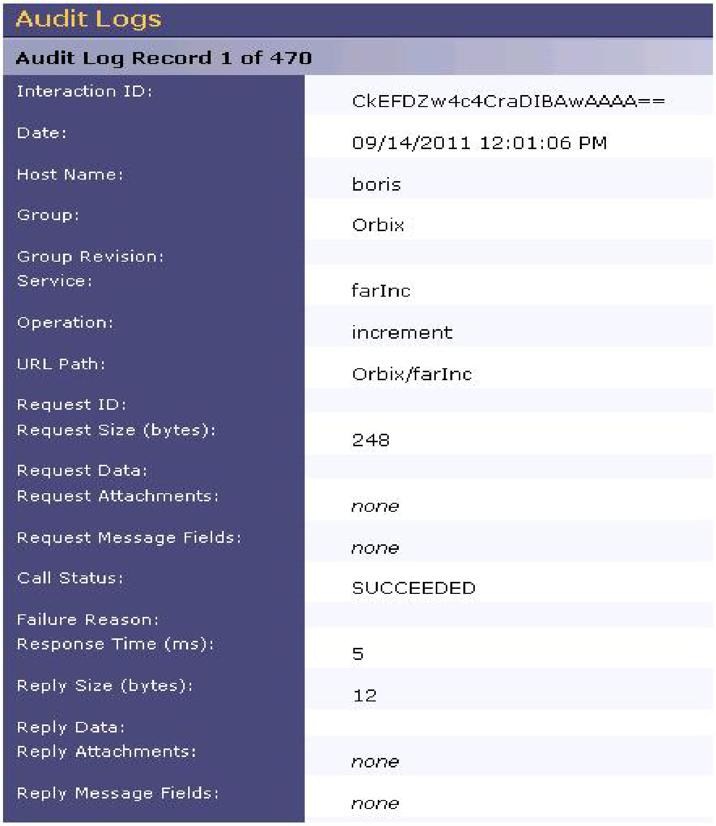


Audit Log Report

Date	Host Name	Service	Operation	Request ID	Call Status	Failure Reason	Response Time (ms)	Authenticated Security ID
09/14/2011 06:09:29 AM	boris	farInc	increment	n/a	SUCCEEDED	n/a	3	n/a
09/14/2011 06:09:27 AM	boris	IT_daemon	getIIOPDetails	n/a	SUCCEEDED	n/a	175	n/a
09/14/2011 06:09:27 AM	boris	IT_daemon	_IT_PING	n/a	SUCCEEDED	n/a	1	n/a
09/14/2011 06:09:27 AM	boris	middleInc	increment	n/a	SUCCEEDED	n/a	2203	n/a
09/14/2011 06:09:27 AM	boris	Client	n/a	n/a	SUCCEEDED	n/a	2206	n/a
09/14/2011 06:09:25 AM	boris	IT_daemon	getIIOPDetails	n/a	SUCCEEDED	n/a	175	n/a

Figure 19: *Audit Logs from application*

Figure 20 shows an example audit log record displayed on clicking an entry for the farInc server in Figure 19.



Audit Logs

Audit Log Record 1 of 470

Interaction ID:	CkEFDZw4c4CraDIBAwAAAA==
Date:	09/14/2011 12:01:06 PM
Host Name:	boris
Group:	Orbix
Group Revision:	
Service:	farInc
Operation:	increment
URL Path:	Orbix/farInc
Request ID:	
Request Size (bytes):	248
Request Data:	
Request Attachments:	*none*
Request Message Fields:	*none*
Call Status:	SUCCEEDED
Failure Reason:	
Response Time (ms):	5
Reply Size (bytes):	12
Reply Data:	
Reply Attachments:	*none*
Reply Message Fields:	*none*

Figure 20: *Orbix server farInc's Log Record*

Message Fields

Message fields are pieces of textual data that are reported, such as the TCP port.

For C++ applications, only the remote port is actually reported.

For Client C++ applications, this is the port to which the client is connected. This is also called the server port.

For Server C++ applications this is the local port of the application, or the port where the server is listening on.

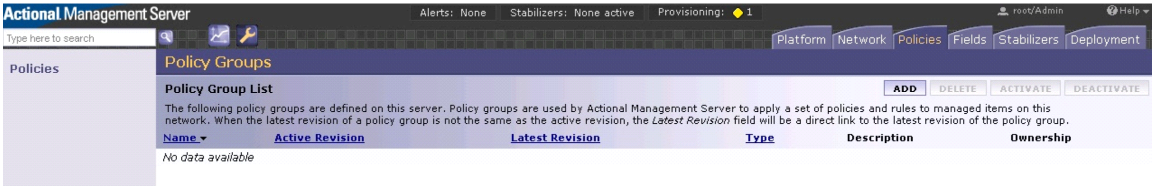
For Java applications, both the clientport and serverport are reported, which means both the local and remote ports of both sides of client and server.

Message fields are turned on by default and are immediately available once the Actional Management Server is configured to look for the message fields.

Configuring Actional to report Message Fields

The Actional Management Server console allows you to configure policies and alerts in configuration mode (click **States** tab, and on the **Stabilizers** page click the **configuration page** link).

1. To create a Policy Group, click the **Policies** tab.
2. Click **Add** to create a new policy group.
3. Enter the **Name, Type,** and **Description.**
4. Click **FINISH**.

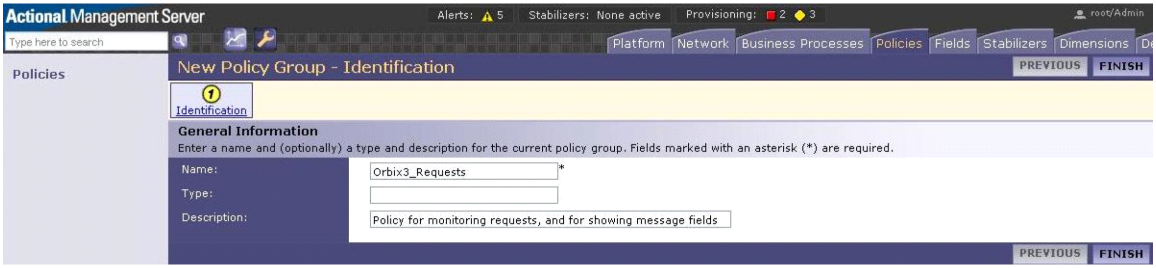


Policy Group List

The following policy groups are defined on this server. Policy groups are used by Actional Management Server to apply a set of policies and rules to managed items on this network. When the latest revision of a policy group is not the same as the active revision, the *Latest Revision* field will be a direct link to the latest revision of the policy group.

Name ▾	Active Revision	Latest Revision	Type	Description	Ownership
No data available					

Figure 21: *Creating a Policy*



New Policy Group - Identification

General Information

Enter a name and (optionally) a type and description for the current policy group. Fields marked with an asterisk (*) are required.

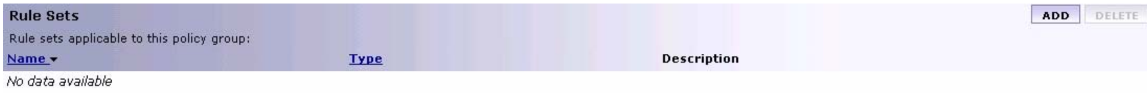
Name: Orbix3_Requests *

Type:

Description: Policy for monitoring requests, and for showing message fields

Figure 22: *Creating a Policy Group*

5. For creating a Rule Set, click the **Policies** tab, and click the name of the policy group to which you want to add a rule set.
6. In the **Rule Set** section, click **Add**.
7. Enter the **Name** and **Description** on the **Rule Set - Identification** page.
8. Click **OK**.

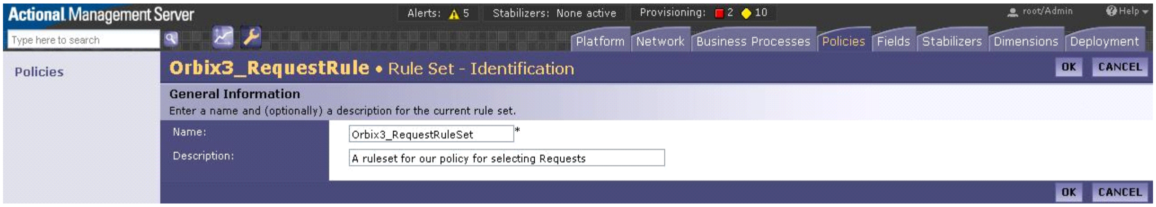


Rule Sets

Rule sets applicable to this policy group:

Name	Type	Description
No data available		

Figure 23: *Creating a new RuleSet*



Orbix3_RequestRule • Rule Set - Identification

General Information

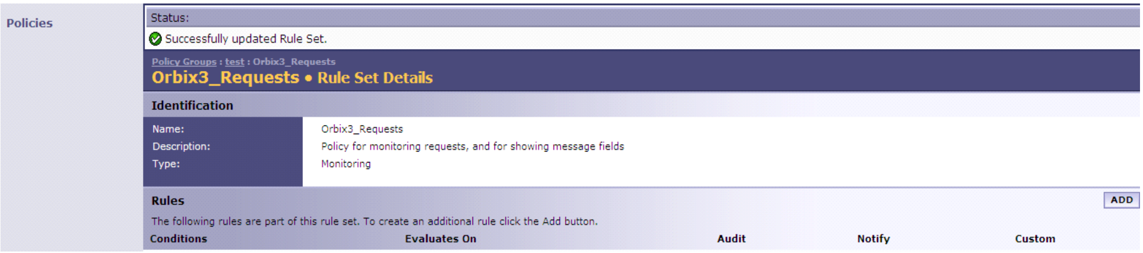
Enter a name and (optionally) a description for the current rule set.

Name: Orbix3_RequestRuleSet *

Description: A ruleset for our policy for selecting Requests

Figure 24: *Creating a RuleSet*

9. After creating the Rule Set, click **Add** on the Rule Set Summary.



Status:

✓ Successfully updated Rule Set.

Policy Groups : test : Orbix3_Requests

Orbix3_Requests • Rule Set Details

Identification

Name:	Orbix3_Requests
Description:	Policy for monitoring requests, and for showing message fields
Type:	Monitoring

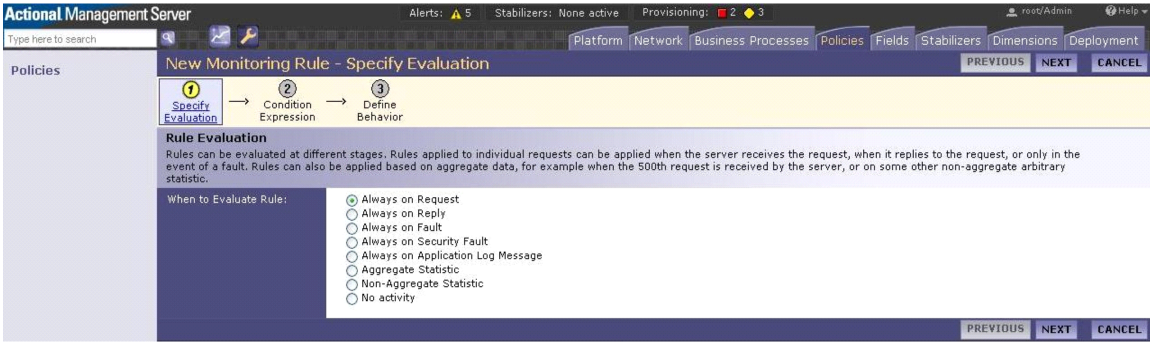
Rules

The following rules are part of this rule set. To create an additional rule click the Add button.

Conditions	Evaluates On	Audit	Notify	Custom

Figure 25: *Add Rule evaluation*

10. On the **New Monitoring Rule - Specify Evaluation** page, select **Always on Request**, and click **Next**.



New Monitoring Rule - Specify Evaluation

Rule Evaluation

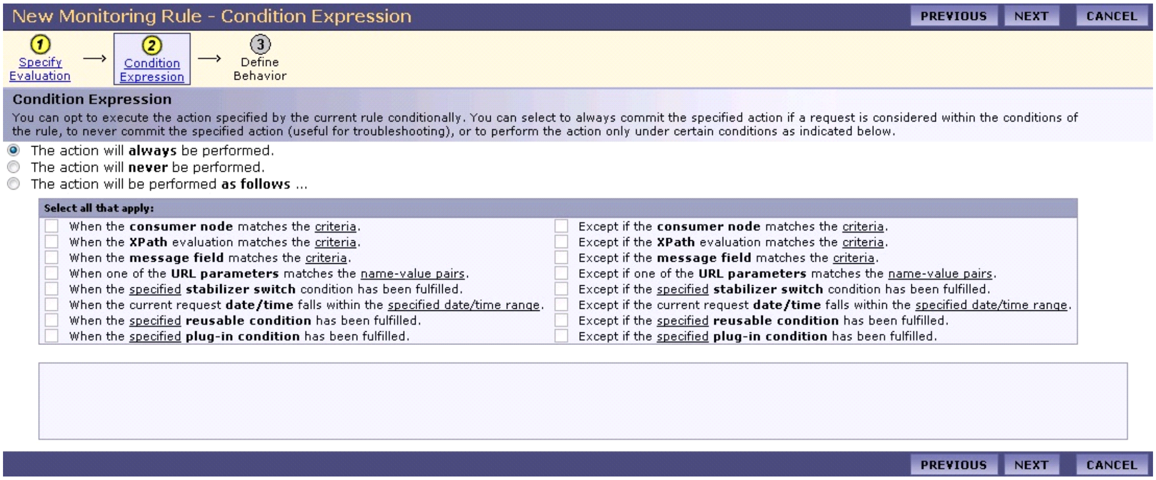
Rules can be evaluated at different stages. Rules applied to individual requests can be applied when the server receives the request, when it replies to the request, or only in the event of a fault. Rules can also be applied based on aggregate data, for example when the 500th request is received by the server, or on some other non-aggregate arbitrary statistic.

When to Evaluate Rule:

● Always on Request
○ Always on Reply
○ Always on Fault
○ Always on Security Fault
○ Always on Application Log Message
○ Aggregate Statistic
○ Non-Aggregate Statistic
○ No activity

Figure 26: *Creating a Rule, selecting a type of Rule*

11. On the **New Monitoring Rule - Condition Expression** page, select **The action will always be performed** option, and click **Next.**



Condition Expression

You can opt to execute the action specified by the current rule conditionally. You can select to always commit the specified action if a request is considered within the conditions of the rule, to never commit the specified action (useful for troubleshooting), or to perform the action only under certain conditions as indicated below.

● The action will **always** be performed.
○ The action will **never** be performed.
○ The action will be performed **as follows** ...

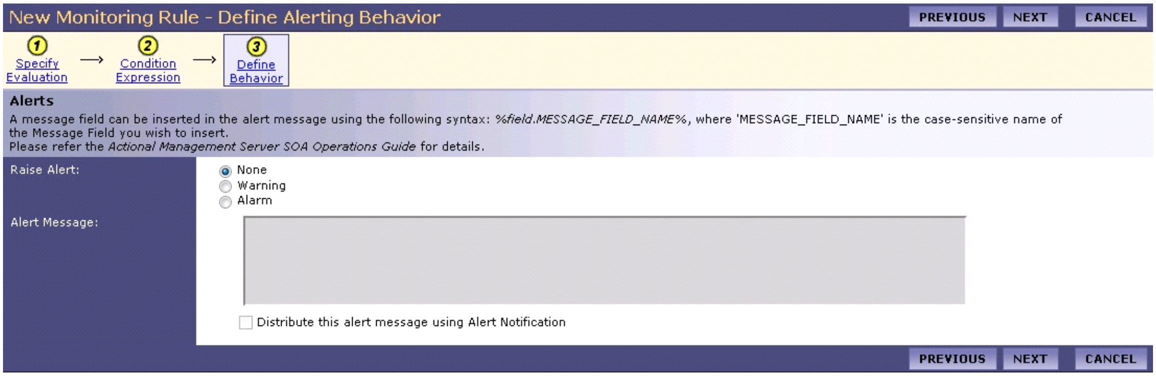
Select all that apply:

□ When the **consumer node** matches the criteria.
□ When the **XPath** evaluation matches the criteria.
□ When the **message field** matches the criteria.
□ When one of the **URL parameters** matches the name-value pairs.
□ When the specified **stabilizer switch** condition has been fulfilled.
□ When the current request **date/time** falls within the specified date/time range.
□ When the specified **reusable condition** has been fulfilled.
□ When the specified **plug-in condition** has been fulfilled.

□ Except if the **consumer node** matches the criteria.
□ Except if the **XPath** evaluation matches the criteria.
□ Except if the **message field** matches the criteria.
□ Except if one of the **URL parameters** matches the name-value pairs.
□ Except if the specified **stabilizer switch** condition has been fulfilled.
□ Except if the current request **date/time** falls within the specified date/time range.
□ Except if the specified **reusable condition** has been fulfilled.
□ Except if the specified **plug-in condition** has been fulfilled.

Figure 27: *Creating a Rule, selecting condition*

12. Any alerts or warnings need not to be setup, hence on the **New Monitoring Rule - Define Alerting Behavior** page, select **None** option, and click **Next**.



Alerts

A message field can be inserted in the alert message using the following syntax: *%field.MESSAGE_FIELD_NAME%*, where 'MESSAGE_FIELD_NAME' is the case-sensitive name of the Message Field you wish to insert.
Please refer the *Actional Management Server SOA Operations Guide* for details.

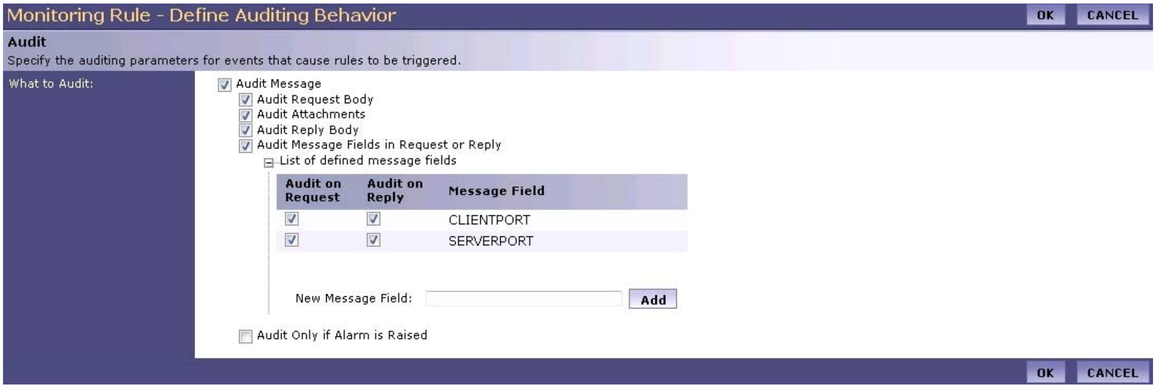
Raise Alert: ● None ○ Warning ○ Alarm

Alert Message:

□ Distribute this alert message using Alert Notification

Figure 28: *Creating a rule, specifying alerts*

13. Do not specify any alerts to be shown.
14. Click **Next** Button. Select all items in the request to be audited, including the message fields (that are already defined). Ensure the last check box **Audit Only if Alarm is Raised** is not selected. If this is selected, the policy will not audit the calls unless an alarm/fault is raised.
15. Click **Next** until you get the **Finish** button on the screen. Clicking **Finish** button takes you to the **Policy Group** view.



Audit

Specify the auditing parameters for events that cause rules to be triggered.

What to Audit:

☑ Audit Message
- ☑ Audit Request Body
- ☑ Audit Attachments
- ☑ Audit Reply Body
- ☑ Audit Message Fields in Request or Reply

⊟ List of defined message fields

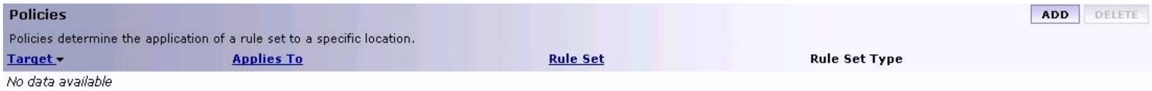
Audit on Request	Audit on Reply	Message Field
☑	☑	CLIENTPORT
☑	☑	SERVERPORT

New Message Field:

☐ Audit Only if Alarm is Raised

Figure 29: *Creating a rule, and specifying message fields*

16. Create a new policy, by clicking the Policy Group, and clicking **Add** on the **Policies** section.
17. Select the Policy that needs to be applied for all sites, and click **Next**.
18. On the **New Policy - Configuration** page, select **Every message** option, and click **Next**.

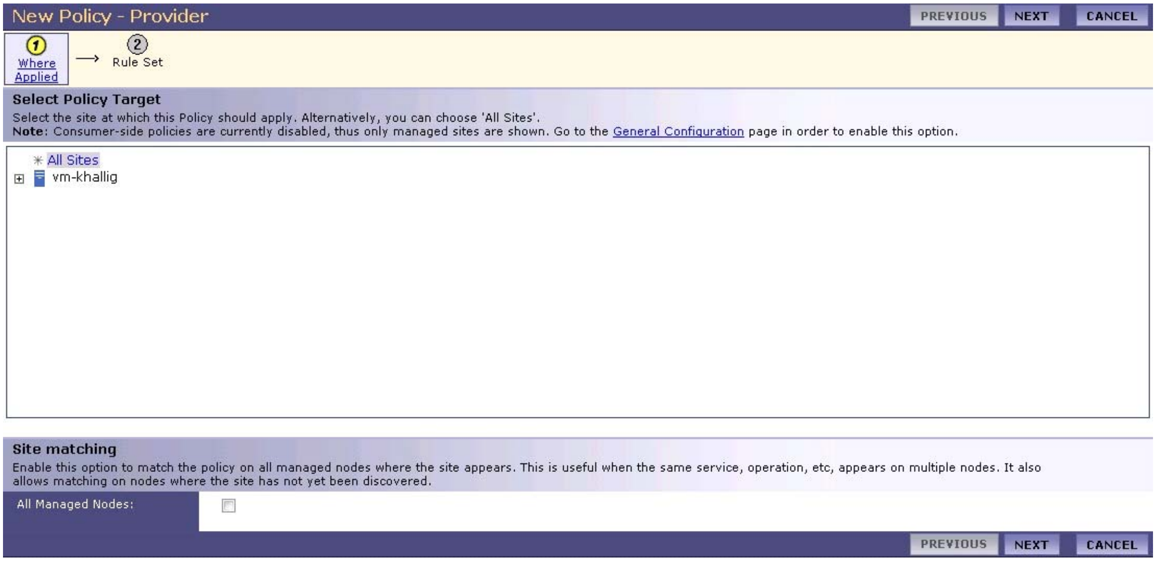


Policies

Policies determine the application of a rule set to a specific location.

Target ▾	Applies To	Rule Set	Rule Set Type
No data available			

Figure 30: *Creating a new Policy*



Select Policy Target

Select the site at which this Policy should apply. Alternatively, you can choose 'All Sites'.
Note: Consumer-side policies are currently disabled, thus only managed sites are shown. Go to the General Configuration page in order to enable this option.

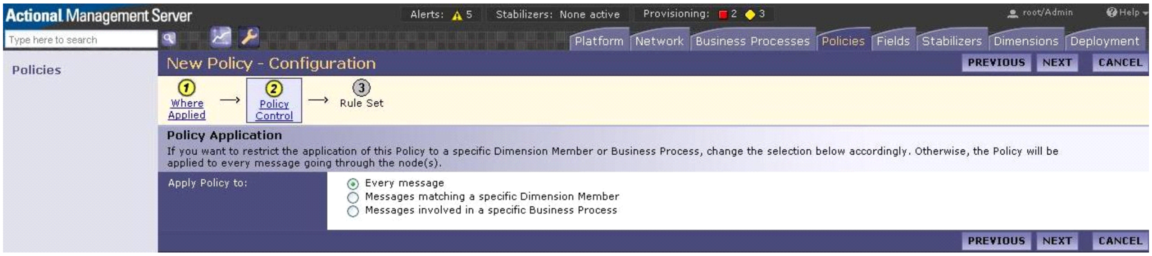
- All Sites
- ⊞ vm-khallig

Site matching

Enable this option to match the policy on all managed nodes where the site appears. This is useful when the same service, operation, etc, appears on multiple nodes. It also allows matching on nodes where the site has not yet been discovered.

All Managed Nodes: □

Figure 31: *Creating a new policy, select sites for policy*



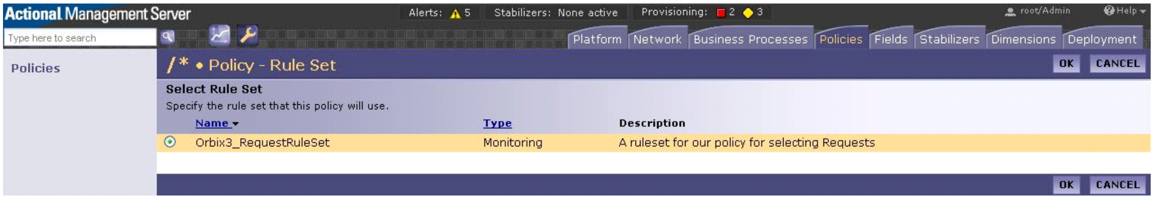
Policy Application

If you want to restrict the application of this Policy to a specific Dimension Member or Business Process, change the selection below accordingly. Otherwise, the Policy will be applied to every message going through the node(s).

Apply Policy to:
- ● Every message
- ○ Messages matching a specific Dimension Member
- ○ Messages involved in a specific Business Process

Figure 32: *Create a new Policy, selecting message type*

19. Select the **RuleSet** that needs to be associated with this policy, and click **OK.**



/* • Policy - Rule Set

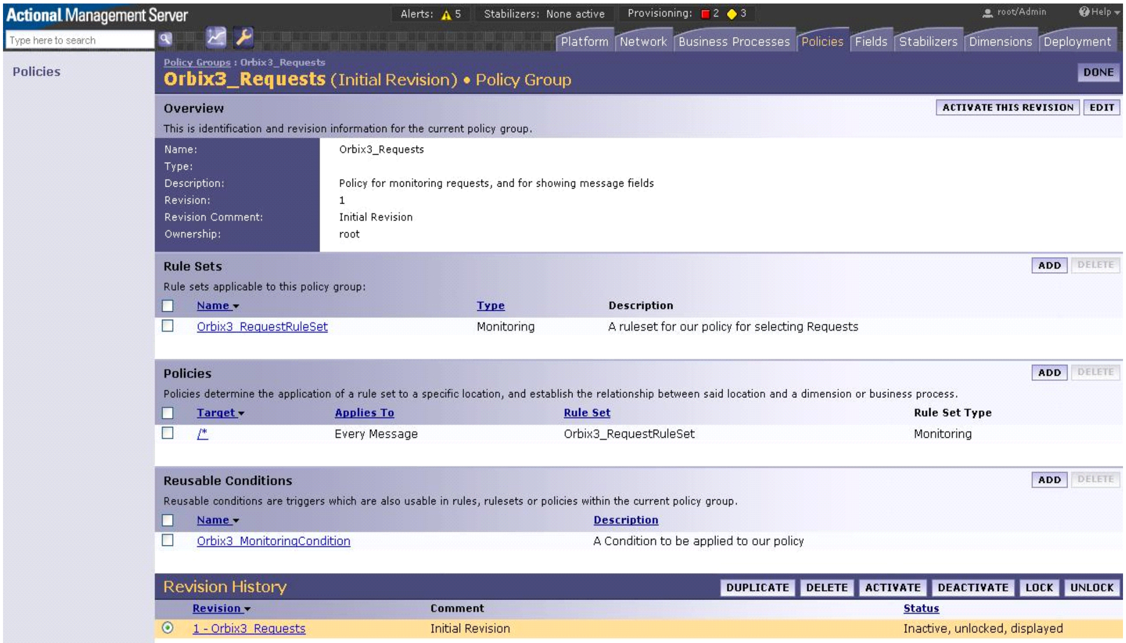
Select Rule Set

Specify the rule set that this policy will use.

	Name	Type	Description
●	Orbix3_RequestRuleSet	Monitoring	A ruleset for our policy for selecting Requests

Figure 33: *Create a new Policy, Selecting a ruleset for the policy*

20. Once you are completed creating your policy, click **ACTIVATE THIS REVISION** which locks the policy and applies it to all sites. All the nodes in the network need to be re-provisioned.



Policy Groups : Orbix3_Requests

Orbix3_Requests (Initial Revision) • Policy Group

Overview

This is identification and revision information for the current policy group.

Name:	Orbix3_Requests
Type:	
Description:	Policy for monitoring requests, and for showing message fields
Revision:	1
Revision Comment:	Initial Revision
Ownership:	root

Rule Sets

Rule sets applicable to this policy group:

□	Name	Type	Description
□	Orbix3_RequestRuleSet	Monitoring	A ruleset for our policy for selecting Requests

Policies

Policies determine the application of a rule set to a specific location, and establish the relationship between said location and a dimension or business process.

□	Target	Applies To	Rule Set	Rule Set Type
□	/*	Every Message	Orbix3_RequestRuleSet	Monitoring

Reusable Conditions

Reusable conditions are triggers which are also usable in rules, rulesets or policies within the current policy group.

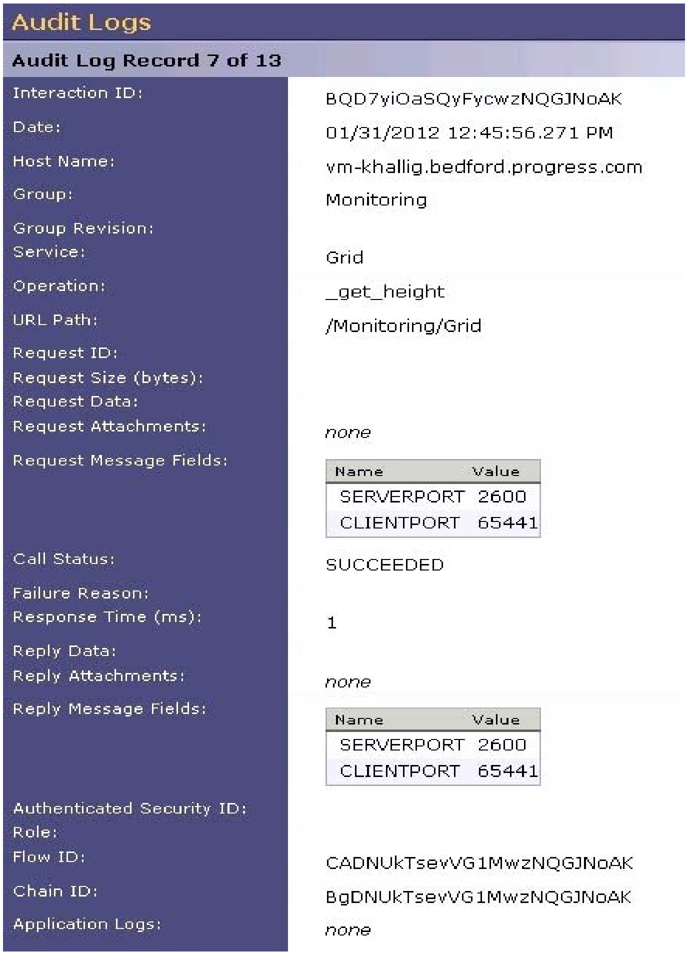
□	Name	Description
□	Orbix3_MonitoringCondition	A Condition to be applied to our policy

Revision History

	Revision	Comment	Status
●	1 - Orbix3_Requests	Initial Revision	Inactive, unlocked, displayed

Figure 34: *Overview of the Policy Group*

21. Once you complete re-provisioning, any new monitored calls are available in the Audit Logs. If you click on any of the new log entries, details of the message fields are displayed. To view the **Audit Logs,** click Stabilizers tab | Stabilizer Management page link | Audit Logs.



Audit Logs

Audit Log Record 7 of 13

Interaction ID:	BQD7yiOaSQyFycwzNQGJNoAK
Date:	01/31/2012 12:45:56.271 PM
Host Name:	vm-khallig.bedford.progress.com
Group:	Monitoring
Group Revision:	
Service:	Grid
Operation:	_get_height
URL Path:	/Monitoring/Grid
Request ID:	
Request Size (bytes):	
Request Data:	
Request Attachments:	*none*
Request Message Fields:	Name / Value SERVERPORT 2600 CLIENTPORT 65441
Call Status:	SUCCEEDED
Failure Reason:	
Response Time (ms):	1
Reply Data:	
Reply Attachments:	*none*
Reply Message Fields:	Name / Value SERVERPORT 2600 CLIENTPORT 65441
Authenticated Security ID:	
Role:	
Flow ID:	CADNUkTsevVG1MwzNQGJNoAK
Chain ID:	BgDNUkTsevVG1MwzNQGJNoAK
Application Logs:	*none*

Figure 35: *Audit log entry showing message fields*

Index