

IBM® Tivoli® Netcool/OMNIbus Probe for
Alcatel-Lucent 5620 SAM v13 JMS
2.0

Reference Guide
July 28, 2016



Notice

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,”](#) on page 45.

Edition notice

This edition (SC27-6577-02) applies to version 2.0 IBM Tivoli Netcool/OMNIbus Probe for Alcatel-Lucent 5620 SAM v13 JMS and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC27-6577-01.

© **Copyright International Business Machines Corporation 2015, 2016.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this guide.....	V
Document Control Page.....	v
Conventions used in this guide.....	v
Chapter 1. Probe for Alcatel-Lucent 5620 SAM v13 JMS.....	1
Summary.....	1
Installing probes.....	3
Configuring the probe.....	4
Configuring the cross-launch application.....	4
Configuring Java.....	6
Jar files.....	6
Configuring lookup tables.....	7
Adapting the rules file.....	7
Establishing a secure connection using SSL.....	10
Configuring SSL.....	10
Displaying unicode and non-unicode characters.....	12
Running the probe.....	13
Probe Event Enrichment Rules.....	13
Data acquisition.....	13
Connectivity and firewall considerations.....	14
Resynchronization.....	14
Using durable subscriptions.....	15
Subscribing to JMS topics.....	15
Subscribing to the 5620-SAM-topic-xml-filtered topic.....	16
Filtering events.....	17
JMS acknowledgement modes.....	18
Alcatel-Lucent SAM server redundancy.....	18
Peer-to-peer failover functionality.....	19
Command line interface.....	20
HTTP/HTTPS command interface.....	20
Properties and command line options.....	23
Properties and command line options provided by the Java Probe Integration Library (probe-sdk- java) version 9.0.....	29
Elements.....	31
Common elements.....	32
Alarm event elements.....	32
Attribute value change event elements.....	33
State change event elements.....	34
Alarm status change event elements.....	34
Object deletion event elements.....	35
Relationship change event elements.....	35
Elements generated for object creation events.....	35
Client session event elements.....	39
File available event elements.....	39
Database activity event elements.....	39
Database connection state change event elements.....	39
Database error event elements.....	39
Database proxy state change event elements.....	40
Error messages.....	40
ProbeWatch messages.....	41

Troubleshooting.....	43
Known Issues.....	43
Appendix A. Notices and Trademarks.....	45
Notices.....	45
Trademarks.....	46

About this guide

The following sections contain important information about using this guide.

Document Control Page

Use this information to track changes between versions of this guide.

The IBM Tivoli Netcool/OMNIbus Probe for Alcatel-Lucent 5620 SAM v13 documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Netcool® Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSHTQ/omnibus/common/kc_welcome-444.html?lang=en

Document version	Publication date	Comments
SC27-6577-00	June 11, 2015	First IBM publication.
SC27-6577-01	July 31, 2015	Information about modifying the default triggers for the Netcool OMNIbus Probe for Alcatel-Lucent 5620 SAM v13 was added to the “Known Issues” on page 43 page.
SC27-6577-02	July 28, 2016	Guide updated for version 2.0 of the probe. “Summary” on page 1 updated. Support extended to 5620 SAM v14.0. Added support launch-in-context feature from the Active Event List in WebGUI. Descriptions for the EnableFailover , SecondaryHost , SecondaryHTTPPort , and SecondaryEJBPort properties added to “Properties and command line options” on page 23. Version 2.0 of the probe addresses the following Enhancement Requests: <ul style="list-style-type: none">• RFE 62030: Improve support for Alcatel-Lucent SAM server redundancy by the ALU 5620 SAM Probe. Added support for the ability to configure a single probe instance with the option to failover between the primary server and the backup server.• RFE 72415: Provide event enrichment rules to enable ITNM RCA function with OMNIbus.

Conventions used in this guide

All probe guides use standard conventions for operating system-dependent environment variables and directory paths.

Operating system-dependent variables and paths

All probe guides use standard conventions for specifying environment variables and describing directory paths, depending on what operating systems the probe is supported on.

For probes supported on UNIX and Linux operating systems, probe guides use the standard UNIX conventions such as `$variable` for environment variables and forward slashes (/) in directory paths. For example:

```
$OMNIHOME/probes
```

For probes supported only on Windows operating systems, probe guides use the standard Windows conventions such as `%variable%` for environment variables and backward slashes (\) in directory paths. For example:

```
%OMNIHOME%\probes
```

For probes supported on UNIX, Linux, and Windows operating systems, probe guides use the standard UNIX conventions for specifying environment variables and describing directory paths. When using the Windows command line with these probes, replace the UNIX conventions used in the guide with Windows conventions. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Note : The names of environment variables are not always the same in Windows and UNIX environments. For example, `%TEMP%` in Windows environments is equivalent to `$TMPDIR` in UNIX and Linux environments. Where such variables are described in the guide, both the UNIX and Windows conventions will be used.

Operating system-specific directory names

Where Tivoli Netcool/OMNIbus files are identified as located within an *arch* directory under NCHOME or OMNIHOME, *arch* is a variable that represents your operating system directory. For example:

```
$OMNIHOME/probes/arch
```

The following table lists the directory names used for each operating system.

Note : This probe may not support all of the operating systems specified in the table.

Operating system	Directory name represented by arch
AIX® systems	aix5
Red Hat Linux® and SUSE systems	linux2x86
Linux for System z	linux2s390
Solaris systems	solaris2
Windows systems	win32

OMNIHOME location

Probes and older versions of Tivoli Netcool/OMNIbus use the OMNIHOME environment variable in many configuration files. Set the value of OMNIHOME as follows:

- On UNIX and Linux, set `$OMNIHOME` to `$NCHOME/omnibus`.
- On Windows, set `%OMNIHOME%` to `%NCHOME%\omnibus`.

Chapter 1. Probe for Alcatel-Lucent 5620 SAM v13 JMS

The Alcatel-Lucent 5620 Service Aware Manager (SAM) is a network management system that is used to manage network nodes. Distributed Alcatel-Lucent 5620 SAM elements are monitored by the Alcatel-Lucent 5620 SAM database using GUI and network management components through a Java-based application. The Probe for Alcatel-Lucent 5620 SAM v13 acquires data from Alcatel-Lucent 5620 SAM using a Java™ Messaging System (JMS).

The Probe for Alcatel-Lucent 5620 SAM v13 supports all revisions of Alcatel-Lucent 5620 SAM V13.0 and V14.0.

Note : This probe is not backwardly compatible to previous versions of Alcatel-Lucent 5620 SAM. For Alcatel-Lucent 5620 SAM V9.0, V10.0, V11.0, and V12.0, use the Netcool/OMNIBus Probe for Alcatel-Lucent 5620 SAM v10 JMS. For download details of that probe, see the following Release Notice:

<http://www-01.ibm.com/support/docview.wss?uid=swg21641822>

This guide contains the following sections:

- [“Summary” on page 1](#)
- [“Installing probes” on page 3](#)
- [“Configuring the probe” on page 4](#)
- [“Running the probe” on page 13](#)
- [“Probe Event Enrichment Rules” on page 13](#)
- [“Data acquisition” on page 13](#)
- [“Properties and command line options” on page 23](#)
- [“Elements” on page 31](#)
- [“Error messages” on page 40](#)
- [“ProbeWatch messages” on page 41](#)
- [“Troubleshooting” on page 43](#)
- [“Known Issues” on page 43](#)

Summary

Each probe works in a different way to acquire event data from its source, and therefore has specific features, default values, and changeable properties. Use this summary information to learn about this probe.

The following table provides a summary of the Probe for Alcatel-Lucent 5620 SAM v13.

Probe target	Alcatel-Lucent/Nokia 5620 Service Aware Manager (SAM) v13.0 and V14.0
Probe executable names	nco_p_alcatel_5620_sam_v13
Probe installation packages	omnibus-arch-probe-nco-p-alcatel-5620-sam-v13-version
Package version	2.0

<i>Table 3. Summary (continued)</i>	
Probe supported on	For details of supported operating systems, see the following Release Notices on the IBM Software Support website: http://www-01.ibm.com/support/docview.wss?uid=swg21959382
Properties files	<code>\$OMNIHOME/probes/arch/alcatel_5620_sam_v13.props</code> <code>\$OMNIHOME/etc/nco_http.props</code> Details about storing commands in this file are described in “Storing commands in the nco_http properties file” on page 23.
Rules files	<code>\$OMNIHOME/probes/arch/alcatel_5620_sam_v13.rules</code>
Alarm pre-classification configuration files	<code>AdvCorr36.include.compat.rules</code> <code>alcatel_5620_sam-preclass.include.snmptrap.rules</code> <code>alcatel_5620_sam-preclass.snmptrap.lookup</code> These configuration files are used by the probe for alarm pre-classification. For details see “Configuring pre-classification” on page 7.
SQL files	<code>files advcorr.sql</code> Details about running this SQL file are described in “Configuring the ObjectServer to support intra-device correlations” on page 8.
Lookup file	<code>alcatel_5620_sam_v13.lookup</code> This lookup file is described in “Configuring lookup tables” on page 7. <code>modify_dedup_trigger.sql</code> This sql file is described in “Known Issues” on page 43 under the Modifying the default Netcool/OMNIbus deduplication triggers heading.
Advanced JMS filter XML file	<code>registerNotification.txt</code> The Alcatel-Lucent 5620 SAM server uses this file to determine which events to send to the probe when the JmsTopic property is set to <code>5620-SAM-topic-xml-filtered</code> . It is specified by the AdvancedFilterConfigFile property of the <code>alcatel_5620_sam_v13.props</code> file, for details see “Properties and command line options” on page 23.
environment file	<code>\$OMNIHOME/probes/java/nco_p_alcatel_5620_sam_v13.env</code> Configuring the environment file is described in “Jar files” on page 6.
Requirements	For details of any additional software that this probe requires, refer to the <code>description.txt</code> file that is supplied in its download package.

<i>Table 3. Summary (continued)</i>	
Connection method	Java Messaging System (JMS) and HTTP(s) Port (optional)
Remote connectivity	The Probe for Alcatel-Lucent 5620 SAM v13 can connect to a remote device. Details of the remote device are specified using the Host , EJBPort , and HTTPPort properties in the properties file; for descriptions of these properties, see “ Properties and command line options ” on page 23.
Multicultural Support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 or IPv6 Note : The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus 7.4.0.
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm . For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the <i>IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide</i> .

Installing probes

All probes are installed in a similar way. The process involves downloading the appropriate installation package for your operating system, installing the appropriate files for the version of Netcool/OMNIbus that you are running, and configuring the probe to suit your environment.

The installation process consists of the following steps:

1. Downloading the installation package for the probe from the Passport Advantage Online website.

Each probe has a single installation package for each operating system supported. For details about how to locate and download the installation package for your operating system, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html

2. Installing the probe using the installation package.

The installation package contains the appropriate files for all supported versions of Netcool/OMNIbus. For details about how to install the probe to run with your version of Netcool/OMNIbus, visit the following page on the IBM Tivoli Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_install_intro.html

3. Configuring the probe.

This guide contains details of the essential configuration required to run this probe. It combines topics that are common to all probes and topics that are peculiar to this probe. For details about additional configuration that is common to all probes, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Configuring the probe

After installing the probe, you need to make various configuration settings to suit your environment.

This section contains topics on the Alcatel-Lucent 5620 SAM configuration requirements:

- [“Configuring Java” on page 6](#)
- [“Configuring the cross-launch application” on page 4](#)
- [“Jar files” on page 6](#)
- [“Configuring lookup tables” on page 7](#)
- [“Configuring pre-classification” on page 7](#)
- [“Establishing a secure connection between the probe and Alcatel-Lucent SAM-O” on page 10](#)
- [“Configuring SSL connections” on page 10](#)
- [“Displaying unicode and non-unicode characters” on page 12](#)

Configuring the cross-launch application

The probe is supplied with a launch-in-context feature that enables you to launch the SAM Fault Management Web app from Netcool/OMNIBus Web GUI Event Viewer right-click tool using two separate methods. Users can configure the GUI manually. Users may also configure the Web GUI using WAAPI which requires createCrossLaunchTool.xml and modifyAlertsMenu.xml.

Users must run addObjectFullName.sql by running one of these commands according to the platform to add ObjectFullName field in alerts.status.

The files are located here:

On Unix:

```
$OMNIHOME/bin/nco_sql -server <objectserver_name> -user <username> -password <password> < <path_to_file>/addObjectFullName.sql
```

On Windows:

```
%NCHOME%\bin\redist\isql.exe -S <objectserver name> -U <username> -P <password> -i <path_to_file>\addObjectFullName.sql
```

The createCrossLaunchTool.xml, modifyAlertsMenu.xml and addObjectFullName.sql are located here:

```
$OMNIHOME/probes/<arch>/5620_SAM_CrossLaunch/
```

```
$OMNIHOME/probes/<arch>/5620_SAM_CrossLaunch/addObjectFullName.sql
```

addObjectFullName.sql is required for the two methods. After running this, you need to uncomment the following lines in the rules file:

```
#if (exists($objectFullName))
#{
    #@ObjectFullName = $objectFullName
#}
```

Note : createCrossLaunchTool.xml and modifyAlertsMenu.xml are not required if you are configuring Web GUI manually.

Configure the Web GUI manually

To configure the Web GUI manually, complete the following steps.

1. Edit the object server alerts.status table to add ObjectFullName field using sql file addObjectFullName.sql

For example:

```
$OMNIHOME/bin/ncs_sql -server NCOMS -username root -password  
' ' </opt/IBM/tivoli/netcool/omnibus/probes/  
linux2x86/5620_SAM_CrossLaunch/addObjectFullName.sql
```

2. Login into the IBM Tivoli Netcool/OMNIBus WebGUI Dashboard Application Service Hub.

https://<DASH IP or hostname>:16311/ibm/console

Refer to the *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide* to create a new tool of script type named 5620SAM-CrossLaunch with the following inputs.

For example:

- a. Open the Event Management Tool (from the Dashboard's side bars)
- b. Select 'Tool Configuration' and create a new tool named "5620SAM-CrossLaunch"
- c. Choose type as script and input script as below:

```
var objName='{@ObjectFullName}'; var samhost = '127.0.0.1';  
var address = 'http://' + samhost + '/FaultManagement?  
view=alarmListImpacts&objectFullName='+ objName;  
window.open (address, "5620SAM Cross Launch");
```

- d. * replace the IP <127.0.0.1> with 5620 SAM IP / hostname accordingly
3. Refer to *IBM Tivoli Netcool/OMNIBus Web GUI Administration and User's Guide* to perform the menu configuration and modify alert menu to include 562SAM-CrossLaunch that should be created in step 2 from available items.

For example:

- a. **Open Event Management Tool > Menu Configurations**
- b. Click on "alerts" from the available menu and click on the "Modify" button.
- c. Add the **5620SAM-CrossLaunch** tool that was previously created in step 2 from the available items to the current items.
- d. Click **Save**.
- e. Open **Event Viewer**.
- f. Click **Refresh**.
- g. Right click on one of the alarms received from SAM. You should see **5620SAM-CrossLaunch** as an option.

Configure the Web GUI using WAAPI

To configure the Web GUI using the WAAPI, complete the following steps.

1. Refer to the *IBM Tivoli Netcool/OMNIBus Web GUI Administration API (WAAPI) User's Guide* to create a new tool by using createCrossLaunchTool.xml. 5620SAM-CrossLaunch must not exist before this step.

For example:

- a. Go to WebGUI WAAPI bin dir.
- b. Modify the createCrossLaunchTool.xml to change the samhost in below line to your SAM EMS host ip or name and save the xml file.
- c. After reviewing and modifying createCrossLaunch.xml run this command:

```
$WAAPI_BIN_DIR/bin/runwaapi -file  
$OMNIHOME/probes/<platforms>/5620_SAM_CrossLaunch/createCrossLaunchTool.xml  
-username <WAS_USER_ID> -password  
<WAS_USER_PASSWÖRD>
```

2. Refer to the *IBM Tivoli Netcool/OMNIBus Web GUI Administration API (WAAPI) User's Guide* to modify the alert menu by using modifyAlertsMenu.xml.

Note : Running the modifyAlertsMenu.xml will overwrite your existing alerts menu items. Make sure in modifyAlertsMenu.xml the content under modify.menu does not overwrite any of your existing items in the Alerts menu.

Make any changes if required, or you can run the manual step in above section to add in the newly created 5620SAM-CrossLaunch tool to the Event Viewer alerts right-click tool menu.

3. After reviewing and modifying the modifyAlertsMenu.xml run this command:

```
$WAAPI_BIN_DIR/bin/runwaapi -file  
$OMNIHOME/probes/<platforms>/5620_SAM_CrossLaunch/modifyAlertsMenu.xml  
-username <WAS_USER_ID> -password <WAS_USER_PASSWORD>
```

4. Open **Event Viewer** and click **Refresh**.

5. Right click on one of the alarms received from SAM. You should see **5620SAM-CrossLaunch** as an option.

Configuring Java

You should configure your PATH and JAVA_HOME environment variable path to at least JRE 7 path before running the probe.

To do so on UNIX and Linux operating systems, run the following command:

```
export PATH=/usr/bin/jre7/jre/bin:$PATH  
export JAVA_HOME=/usr/bin/jre7/jre
```

To do so on Windows operating systems, run the following command:

```
set PATH=C:\\Java\\Jre7\\jre\\bin;%PATH%  
set JAVA_HOME=C:\\Java\\Jre7\\jre
```

Jar files

As a part of the configuration process, you must copy the sam0ss.jar file to the java folder in your Netcool/OMNIbus installation.

On UNIX and Linux operating systems, the java folder is in the following location:

```
$OMNIHOME/probes/java
```

On Windows operating systems, the java folder is in the following location:

```
%OMNIHOME%\probes\win32
```

Configuring the SAMJARHOME environment variable

If you want to use a path other than the default path mentioned in the previous section for the sam0ss.jar file, you must configure the SAMJARHOME environment variable accordingly.

On UNIX and Linux operating systems, configure the \$SAMJARHOME environment variable in the probe environment file (\$OMNIHOME/probes/java/nco_p_alcatel_5620_sam_v13.env) to include the path to sam0ss.jar. To do this, add the following line:

```
SAMJARHOME=directory_path
```

Where *directory_path* is the full path to sam0ss.jar file

For example, add the line:

```
SAMJARHOME=/opt/5620sam/sam0ss.jar
```

Note : As an alternative to setting the SAMJARHOME environment variable, copy the sam0ss.jar file to \${OMNIHOME}/probes/java and the probe will load the Jar file from there.

On Windows operating systems, configure the %SAMJARHOME% environment variable to include the path to sam0ss.jar. To do this, run the following command:

```
set SAMJARHOME=directory_path
```

Where *directory_path* is the full path to sam0ss.jar file

For example, run the following command:

```
set SAMJARHOME=C:\5620sam\jar
```

Note : As an alternative to setting the SAMJARHOME environment variable, copy the sam0ss.jar file to %OMNIHOME%\probes\win32 and the probe will load the Jar file from there.

Location of the jar files

If you use the default installation, you can find the jar file in the following location of the Alcatel-Lucent 5620 SAM server:

```
/opt/5620sam/server/nms/integration/SAM_0/sam0ss.jar
```

Configuring lookup tables

The probe is supplied with a lookup table that contains details of the various types of alarms that Alcatel-Lucent 5620 SAM generates. You may need to update the rules file to include the path to the lookup table in your installation.

The following lookup file is delivered with the Probe for Alcatel-Lucent 5620 SAM v13:

```
alcatel_5620_sam_v13.lookup
```

This file is installed in the following location:

```
$OMNIHOME/probes/includes
```

The path to the lookup file is included in the rules file. If you are not running the probe from \$OMNIHOME/probes, you must change this path accordingly.

Note : You cannot use \$OMNIHOME in the paths to the lookup files.

Configuring pre-classification

In addition to automated deduplication and generic-clear event correlation, this probe also supports alarm pre-classification. Pre-classification allows you to identify and flag within the probe rules file the causal relevance of each event type. This feature enables you to overcome many of the shortcomings of Root Cause Analysis (RCA) systems by more efficiently and accurately determining the root cause of a failure from the events processed.

Advanced correlation is implemented by building a lookup table referenced by the probe rules file to determine the causal relevance of a received event before it is forwarded to the Netcool/OMNIBus ObjectServer.

Details about the causal relevance of each event type are stored in the alcatel_5620_sam-preclass.snmptrap.lookup file. You can modify this file to add to or change the causal relevance details.

The entry for each event type uses the following format:

```
{ "Event_Id", "Causal_relevance" }
```

For example: {"alcatel-5620-sam-LinkDown", "0"},

The causal relevance of each event type is defined using one of the of following predefined integer values:

0 – Unknown

- 1 – Root Cause
- 2 – Symptom
- 3 – Singularity
- 4 – Information

To determine the causal relevance assigned to an entry, use the following guidelines:

Unknown

If the condition represented by an event cannot be classified as a root cause, symptom, singularity, or informational event, it is Unknown.

Root Cause

A condition that is known not to be caused by any other detectable condition.

Symptom

A condition that is known to always be caused by another detectable condition. Generally these are events that indicate degraded conditions or failures of higher level entities or processes.

Singularity

If an event represents a degraded condition or failure that is known to not be a root cause, and cannot be caused by another condition, it is a singularity event.

Information

If an event provides information about a system that does not represent a degraded condition or failure, or it indicates the clearing, or resolution, of a previously occurring fault related condition, it is an informational event.

To use the alarm pre-classification feature provided with this probe, you must perform the tasks described in the following topics:

- [“Configuring the ObjectServer to support intra-device correlations” on page 8](#)
- [“Manually adding conversions to the ObjectServer” on page 9](#)

Configuring the ObjectServer to support intra-device correlations

You must perform this task as an IBM Tivoli Netcool/OMNIBus V7.x user with ISQLWrite permissions.

From a command prompt, run the extracted `advcorr.sql` script using one of the following platform-dependent, case-sensitive commands.

UNIX and Linux operating systems

On UNIX and Linux operating systems, run the following command:

```
$OMNIHOME/bin/ncosql -server objectserver_name -user username -password password < path_to_file/advcorr.sql
```

where:

`$OMNIHOME` is your installation location of Tivoli Netcool/OMNIBus V7.x

`objectserver_name` is the name assigned to your ObjectServer.

`username` and `password` are your ObjectServer login details.

`path_to_file` is the path to the `advcorr.sql` file.

Windows operating systems

On Windows operating systems, run the following command:

```
%NCHOME%\bin\redist\isql.exe -S objectserver_name -U username -P password -i path_to_file\advcorr.sql
```

where:

2. Right click **alert.status** and click **Add Conversion** to access the **Conversion Details** window.
3. Select **AdvCorrCauseType** from the **Column** dropdown list, and then create the following entries within the **Value** and **Conversion** fields, clicking **OK** to save each set of entries:

Table 4. Value and conversion mapping

Value	Conversion
0	Unknown
1	Root cause
2	Symptom
3	Singularity
4	Information

4. Repeat steps 2 and 3 to set up the same conversions for the CauseType column, substituting CauseType as the **Column** field entry in step 3.

On completion, the **Conversions** window appears.

5. Enable the pre-class rules by uncommenting the following lines found in the `alcatel_5620_sam_v13.rules` file:

```
include "../includes/alcatel_5620_sam-preclass.include.snmptrap.rules"
include "../includes/AdvCorr36.include.compat.rules"
```

Note : Without the configuration described in this topic, the probe will only use the default rules file configuration and will ignore the alarm pre-classification processing.

Establishing a secure connection between the probe and Alcatel-Lucent SAM-O

To establish a secure communication between the probe and the Alcatel-Lucent SAM-O interface, you should use an SSL connection with HTTPS. Such a connection significantly reduces the vulnerabilities of hijacked sessions and passwords.

Configuring SSL connections

If the Alcatel-Lucent 5620 SAM server is using a Secure Socket Layer (SSL) connection to encrypt data exchanged over JMS and HTTP, you will need to configure the truststore for the HTTPS connection on the Netcool/OMNIbus probe server.

Importing a certificate file into the truststore

To import a certificate file into the truststore, use the following steps:

1. *Either* obtain the Alcatel EMS security certificate from the certificate authority (CA) *or*, if you have the appropriate user access, export the Alcatel EMS keystore into a certificate on the Alcatel EMS server. Refer to the note below for details of where to find instructions about obtaining certificate files and how to generate a keystore file.

To export the Alcatel EMS keystore file from the Alcatel EMS server into a certificate, use the following command:

```
./keytool -export -alias alias_name -keystore keystore_file -storepass
password -file certificate_file
```

Where:

alias_name is the keystore alias specified during Alcatel EMS keystore generation, for example: SAM13.

keystore_file is the path to and name of the Alcatel EMS keystore file, for example: /opt/samserver.keystore.

password is the Alcatel EMS keystore password, for example: the password of samserver.keystore.

certificate_file is the path to and name of the certificate file to be created, for example: /opt/samcert.

2. On the Netcool/OMNIbus probe server, you need to import the Alcatel EMS certificate *either* into an existing truststore *or* into a new truststore.

To import the Alcatel EMS certificate into *an existing* truststore, you need to point an alias to a key entry as the certificate reply. The alias must be the same as that specified during Alcatel EMS server keystore file generation, for example:

```
./keytool -import -trustcacerts -alias alias_name -file certificate_file
-keystore truststore_file -storepass password
```

To import the Alcatel EMS certificate into *a new* truststore, make sure that the alias that you specify does not already exist in your truststore, for example:

```
./keytool -import -trustcacerts -alias new_alias_name -file
certificate_file -keystore truststore_file -storepass password
```

In the preceding examples:

alias_name is the keystore alias specified during Alcatel EMS keystore generation, for example: SAM13.

new_alias_name is the keystore alias of a new keystore, for example: samprobe.

certificate_file is the path to and name of the certificate file created earlier, for example: /opt/samcert.

truststore_file is the path to and name of the truststore file that will contain the imported certificate, for example: /opt/samserver.truststore

password is the Alcatel EMS keystore password, for example: the password of samserver.truststore.

3. Verify that the certificate has been imported into the keystore using the following command:

```
./keytool -list -v -keystore truststore_file
```

Where:

truststore_file is the path to and name of the truststore file generated, for example: /opt/alusam.trustStore.

Note : For more details about configuring SSL security for the Alcatel-Lucent 5620 SAM server (including instructions about obtaining certificate files), refer to the *Alcatel 5620 SAM Installation and Upgrade Guide*.

Configuring the probe

To configure the probe to connect to the Alcatel-Lucent 5620 SAM server using an SSL connection, use the following steps:

1. Set the probe's **UseSSL** property to `true` and configure the probe's **HTTPPort** property to use the default Alcatel EMS HTTPS port, 8443.
2. Specify values for the following probe properties:
 - **TrustStore**: Specify the path of the probe's Java keystore that you created in the steps for importing the certificate into the truststore, for example: `/opt/alusam.trustStore`.
 - **TrustStorePassword**: Specify the password that you set for the Java keystore.
 - **CertificateStore**: Specify the path of the certificate keystore. This will be the same value as that set for the **TrustStore** property unless you manage them in different keystore files.
 - **CertificateStorePassword**: Specify the password set for the certificate keystore.

You can set both the **TrustStore** property and the **CertificateStore** property to the same keystore file where the license file is imported, or you can specify different keystore files.

Example SSL configuration property settings

The following example shows SSL configuration settings from the properties file of an example Probe for Alcatel-Lucent 5620 SAM v13:

```
Host                : "198.162.20.21"
HTTPPort           : 8443
UseSSL             : "true"
TrustStore         : "/opt/alusam.trustStore"
TrustStorePassword : "newpassword"
CertificateStore    : "/opt/alusam.trustStore"
CertificateStorePassword : "newpassword"
```

Displaying unicode and non-unicode characters

The probe can support multibyte characters and so can display both unicode and non-unicode characters.

Before using the probe to process data that contains multibyte characters, perform the following steps:

1. Ensure that the Alcatel-Lucent 5620 SAM server is configured to send data to the probe in UTF-8 format.

The probe can support only multibyte characters if the incoming data has been configured in correct UTF-8 format. Consult your Alcatel-Lucent 5620 SAM server documentation for details about how to configure the server to send data in UTF-8 format.

2. Check that the probe server has UTF-8 support enabled and that the correct locale is set; for example, set the locale to Chinese.

On Windows operating systems, use the following steps:

- a. Access the **Region and Language** section of the **Control Panel**.
- b. Select the **Formats** tab.
- c. Select **Format > Chinese (Simplified, PRC)**.
- d. Select the **Administrative** tab.
- e. Select **Change system locale**.
- f. Select **Current system locale > Chinese (Simplified, PRC)**.
- g. Click **OK**.
- h. Click **OK**.

On UNIX and Linux operating systems, set the system locale using the LANG and LC_ALL environment variables:

```
export LANG=zh_CN.utf8
export LC_ALL=zh_CN.utf8
```

3. Restart the ObjectServer.
4. Configure the ObjectServer to enable the insertion of UTF-8 encoded data. See the *Netcool/OMNIBus Installation and Deployment Guide*.
5. If you are running the probe on a Windows operating system, you must use the `-utf8enabled` command-line option each time you start the probe.

Note : Netcool/OMNIBus can only receive multibyte characters from the Network Element (NE) custom properties using the `additionalText` field sent in the alarm. The custom text field also supports multibyte characters. However, this is an attribute of the alarm policy, which is not sent in the alarm. UTF-8 encoding is used for all fields in the Alcatel-Lucent 5620 SAM server, but multibyte character support only applies to these few selected attributes. This is because the NE does not support multibyte characters.

Running the probe

To start the probe on UNIX or Linux operating systems, use the following command: `$OMNIHOME/probes/nco_p_alcatel_5620_sam_v13`

To start the probe on a Windows operating systems, use the following command: `%OMNIHOME%\probes\win32\nco_p_alcatel_5620_sam_v13.bat`

Probe Event Enrichment Rules

The default enrichment rules are compatible with the Probe for ALU SAM V13 and V14 developed for IBM Tivoli Network Manager 4.1.1 or newer. Default enrichment rules which sets the `LocalPriObj`, `NMosEventMap`, `LocalNodeAlias`, `RemoteNodeAlias` for alarms to be enriched depending on the type of event map to be used.

A rules file specific to the Alcatel-Lucent 5620 SAM v13 JMS will be included in the Probe Extension Package guide. For details, see the Knowledge Center:

To use the rules files:

1. Download and install the Probe Extension package.
2. Edit the main probes rules files to uncomment the include statement to the event enrichment file. Update the file path of the enrichment file, for example include:

```
"$PROBE_EXT/eventenrichment/SAM5620/alcatel_5620_sam.enrichment.rules"
```

Users may customize this rules files to support more alarms for enrichment.

http://www.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/probe_ext_pack/wip/concept/probe_ext_pack_intro.html?lang=en

Data acquisition

Each probe uses a different method to acquire data. Which method the probe uses depends on the target system from which it receives data.

The Probe for Alcatel-Lucent 5620 SAM v13 acquires current alarms and downtime alarms (if there is a loss of connection) by connecting to the HTTP port on the Alcatel-Lucent 5620 SAM server. The probe can also obtain alarms as they are generated by subscribing to the Java Messaging Service (JMS) running on the Alcatel-Lucent 5620 SAM server.

The probe obtains data from Alcatel-Lucent 5620 SAM as follows:

1. The probe connects to the machine on which the Alcatel-Lucent 5620 SAM server is running specified by the **Host** property.
2. If the **InitialResync** property is set to true, the probe connects to the **HTTPPort**, logs onto the SAM server using the values specified for the **SamServerUserName** and **SamServerPassword** properties, and sends an HTTP request to perform a resynchronization.

The Alcatel-Lucent 5620 SAM server sends all currently active alarms.

3. The probe then connects to the **EJBPort** and subscribes to the JMS topic specified by the **JmsTopic** property to receive new events as they are generated.

Note : You can specify filter criteria that the probe sends with the request using the **JmsFilter** property.

The Alcatel-Lucent 5620 SAM server sends all events that meet the filter criteria to the probe.

4. The probe extracts event information using a SAX parser and sends it to the ObjectServer.
5. If the probe received a **TerminateClientSession** alarm from the Alcatel-Lucent 5620 SAM server, the probe ends the JMS session and shuts down.

Data acquisition is described in the following topics:

- [“Connectivity and firewall considerations” on page 14](#)
- [“Resynchronization” on page 14](#)
- [“Using durable subscriptions” on page 15](#)
- [“Subscribing to JMS topics” on page 15](#)
- [“Subscribing to the 5620-SAM-topic-xml-filtered topic” on page 16](#)
- [“Filtering events” on page 17](#)
- [“JMS acknowledgement modes” on page 18](#)
- [“Alcatel-Lucent SAM server redundancy” on page 18](#)
- [“Peer-to-peer failover functionality” on page 19](#)
- [“Command line interface” on page 20](#)

Connectivity and firewall considerations

To connect to the Alcatel-Lucent 5620 SAM server, the probe uses the **EJBPort** and **HTTPPort** properties.

The **EJBPort** property specifies the EJB port to which the probe connects. The default value for this property is 1099. The **HTTPPort** property specifies the HTTP port to which the probe connects. The default value for this property is 8080.

For connectivity within a firewall, Alcatel-Lucent 5620 SAM also uses the JMS port. The JMS service details are specified within the Alcatel-Lucent 5620 SAM server. The JMS service can only be altered from the Alcatel-Lucent 5620 SAM server configuration. The default value for the JMS port is 8093.

Resynchronization

On startup, you can instruct the probe to connect to an HTTP port of the Alcatel-Lucent 5620 SAM server, and send a SOAP/XML request to resynchronize all active alarms.

Using the **InitialResync** property, the probe can resynchronize alarms generated while the network elements managed by the SAM server are either **inService** (active) or **inMaintenance** (in the maintenance stage or inactive). The Alcatel-Lucent 5620 SAM server generates **inService** alarms when the associated network elements are active, and generates **inMaintenance** alarms when they are inactive.

To resynchronize only the active network elements and to receive only the **inService** alarms, set the **XMLretrieveUseInService** property to 1.

If you have specified a value for the **RecoveryFile** property, you must create the file manually. The server only sends alarms that have been generated since the timestamp indicated in the recovery file.

If you set the **ResynchInterval** property to a value greater than 1, the probe periodically resynchronizes with the Alcatel-Lucent 5620 SAM server. If you set the **ResynchInterval** property to a low value, you may experience performance issues if the probe has to resynchronize large amounts of active alarms each interval.

If you set the **Optimize** property to 1, the probe optimizes performance by not writing the resynchronized events received from the HTTP request and the JMS events to the log file.

Using durable subscriptions

The probe can make a durable or a non-durable subscription with a JMS topic.

In durable subscription to a topic, an active subscriber specified by the probe receives messages that were published on the topic while the subscriber was inactive. When the durable subscription is enabled, the JMS stores messages published on the specified topic while the subscriber is not active or disconnected from the JMS.

In a non-durable subscription to a topic, the subscriber only receives messages that are published on that topic while the subscriber is active. The JMS drops other messages that are sent while the subscriber is inactive.

Enabling durable subscriptions

You can specify the identifier of the subscriber in the **PersistentJmsId** property, and enable a durable subscription using the **Durable** property. When the probe reconnects with the same subscriber value, the JMS sends the stored events.

In durable subscription mode, if you stop the probe manually using **Ctrl - C** and the **StoreEvents** property is set to 0 the JMS session becomes inactive. However, when the probe is stopped unexpectedly the durable JMS session remains active (or if the **StoreEvents** property is set to 1). For example, if the probe is stopped due to a system crash, the JMS session remains active. This helps the probe receive alarms generated when it was inactive.

Note : Make sure that the same `JmsClientId` value specified using the **PersistentJmsID** property is part of the **JmsFilter** property value.

Disabling durable subscriptions

To disable the durable subscription mode, set the value for the **Durable** property to `false`. If the value specified for this property is `false`, the probe makes a non-durable subscription with the JMS topic and receives active alarms.

Note : The JMS removes the stored messages once the probe receives new messages, when the stored messages expire, or when the durable subscription is deleted.

Subscribing to JMS topics

The probe can subscribe to a JMS topic and receive events in XML format.

The following table describes the topics to which the probe can subscribe using the **JmsTopic** property.

JMS topic	Description
5620-SAM-topic-xml-fault	Subscribes to <code>fault</code> events in XML format.
5620-SAM-topic-xml-file	Subscribes to <code>findToFile</code> events in XML format.

<i>Table 5. JMS topics (continued)</i>	
JMS topic	Description
5620-SAM-topic-xml-general	Subscribes to general events (such as object creation and deletion) in XML format.
5620-SAM-topic-xml-stats	Subscribes to events related to statistics collection.
5620-SAM-topic-xml-filtered	Subscribes to all events using advanced filtering to limit the number of JMS messages that are sent.

Subscribing to the 5620-SAM-topic-xml-filtered topic

The Probe for Alcatel 5620 SAM v13 JMS can subscribe to the `5620-SAM-topic-xml-filtered` topic. This topic supports the advanced JMS message filter which allows you to base event message selection on the content of the event message body.

To subscribe to the `5620-SAM-topic-xml-filtered` topic, set the **JmsTopic** property to `5620-SAM-topic-xml-filtered`. The probe will receive events of type `EventVessel` and `XMLFilterChange`. You can limit the messages that the SAM sever sends to the probe by specifying an XML filter using the **AdvancedFilterConfigFile** property.

You can dynamically modify the XML message filter when the probe is running by issuing a `registerNotification` command from the CLI, or you can specify an alternative XML filter by issuing a `registerNotificationFile` command. For details of these commands, see [“Command line interface” on page 20](#).

The default **AdvancedFilterConfigFile** contains the following XML code:

```
<filter-Set>
  <filter>
    <and>
      <equal class="jmsEvent" name="MTOSI_NTType" value="NT_ALARM" />
      <not>
        <equal class="jmsEvent" name="MTOSI_NTType"
          value="NT_ATTRIBUTE_VALUE_CHANGE" />
      </not>
    </and>
  </filter>
</filter-Set>
<extraTags>
  <tag name="ALA_category" />
  <tag name="eventName" />
  <tag name="MTOSI_osTime" />
  <tag name="MTOSI_objectType" />
  <tag name="MTOSI_perceivedSeverity" eventName="ObjectCreation" />
  <tag name="ALA_alarmType" eventName="ObjectCreation" />
  <tag name="ALA_clientId" />
</extraTags>
```

When the probe subscribes to the `5620-SAM-topic-xml-filtered` topic, by default, the event vessel does not contain the JMS header properties of the individual events. You must use the **AdvancedFilterConfigFile** property to specify the JMS header properties (for example, `MTOSI_osTime`, `eventName`, and so on).

For information about how to configure complex filters for advanced filtering purposes, refer to the Alcatel-Lucent SAM documentation.

Note : The Probe for Alcatel 5620 SAM v13 only supports XML filters that have 90,000 characters or fewer. If the XML filter specified by the **AdvancedFilterConfigFile** property contains more than 90,000 characters, the probe writes the following warning message to the probe log file:

```
Probe only support AdvancedFilterConfigFile: ...
```

Filtering events

You can specify how the JMS filters the events sent to the probe by using the **JmsFilter** property. This property allows you to specify a filter that uses the JMS properties associated with the event.

You can filter events using any of the JMS properties. For example, the following filter is based on fault messages with a severity class of Warning:

```
ALA_category in (\'FAULT\') and MTOSI_perceivedSeverity in (\'Warning\')
```

By default, the JMS filters events sent to the probe using the following filter:

```
ALA_category not in (\'STATISTICS\', \'ACCOUNTING\')
```

Note : The JMS filter must contain `ALA_clientId in ('persistent_jms_id', '')` where `persistent_jms_id` is the value set for the **PersistentJmsId** property. If you change the value of the **PersistentJmsId** property, you must also change the value in the JMS filter accordingly.

The following table describes all JMS properties that are available for use in filters. For each property, the table lists the possible values that the property can take.

Table 6. JMS properties		
Property	Values	Description
MTOSI_NTType	ALA_OTHER NT_OBJECTCREATION NT_OBJECTDELETION NT_ATTRIBUTE_VALUE_CHANGE NT_STATE_CHANGE NT_ALARM NT_HEARTBEAT ALA_RELATIONSHIPCHANGE	Use this property to specify the type of notification that you want the filter to send to the probe.
MTOSI_osTime	Time in milliseconds	Use this property to specify the time when the event was created on the server. The filter sends this time of the event creation to the probe.
MTOSI_objectName	Object name	Use this property to specify the object name for the event that you want the filter to send to the probe.
MTOSI_objectType	Object type	Use this property to specify the object type for the event that you want the filter to send to the probe.

Table 6. JMS properties (continued)

Property	Values	Description
ALA_category	SERVICE EQUIPMENT ACCOUNTING GENERAL FAULT STATISTICS DATABASE SOFTWARE	Use this property to specify the category of the event that you want the filter to send to the probe.
ALA_allomorphic	Allomorphic class for the object type	Use this property to specify the allomorphic class for the object type that you want the filter to send to the probe.
ALA_topic	JMS topic	Use this property to specify the JMS topic for the message that you want the filter to send to the probe.
ALA_clientId	Specific client ID	Use this property to specify the client ID that you want the filter to send to the probe.

JMS acknowledgement modes

Within the Alcatel-Lucent 5620 SAM server, the probe acts as a JMS client and supports either of the two acknowledgement modes: DUPS_OK_ACKNOWLEDGE or AUTO_ACKNOWLEDGE. You can set the **JMSAcknowledgeMode** property to a mode in which the probe should run as a JMS client.

If the JMS is running in DUPS_OK_ACKNOWLEDGE mode, the probe allows the JMS provider to send a message more than once to the same destination. This mode can be chosen when the performance and throughput of the probe decrease to avoid duplicate messages.

If the JMS is running in AUTO_ACKNOWLEDGE mode, the probe automatically acknowledges a received message, and waits for a broker acknowledgement of the message.

Alcatel-Lucent SAM server redundancy

Two Alcatel-Lucent 5620 SAM servers can run in a redundancy pair (that is, one runs as the primary server and the other as a backup server). This affects the way that you configure the probe.

If the primary SAM server is down while the probe is connected, then the probe will attempt to connect to the secondary SAM server which will take over the primary server role. The probe cannot connect to the secondary SAM server if the primary server is still operational.

This feature works with the **RetryCount** property and the optionally with the **RetryInterval** property.

If **RetryCount** is 0 (i.e. retry is disabled), redundancy feature is disabled too.

The **RetryInterval** property should be set to the appropriate amount of time needed for the SAM server to complete the server activity switch.

When the **EnableFailover** property is set to true the probe is trying to setup a connection with the new primary SAM server and you should see the following messages.

ProbeWatch message:

Connecting new primary SAM server.

Debug message:

Setting up connection to primary/secondary host.

When the probe is successfully connected to the primary SAM server, you will see the following probewatch message:

Connected to the primary SAM server.

These are the probe properties impacted by the SAM server redundancy:

- **Host** (The primary host in redundancy).
- **HTTPEndPoint** (This is the primary HTTP port in redundancy).
- **EJBPort** (EJBPort is the primary EJB port in redundancy).
- **EnableFailover**
- **SecondaryHost**
- **SecondaryHTTPEndPoint**
- **SecondaryEJBPort**

Note : If **EnableFailover** is set to true then the **RetryCount** must be set to a value greater than 0.

Peer-to-peer failover functionality

The Probe for Alcatel-Lucent 5620 SAM v13 supports failover configurations where two probes run simultaneously. One probe acts as the master probe, sending events to the ObjectServer; the other acts as the slave probe on standby. If the master probe fails, the slave probe activates.

While the slave probe receives heartbeats from the master probe, it will not forward events to the ObjectServer. If the master shuts down, the slave probe will stop receiving heartbeats from the master and any events it receives thereafter will be forwarded to the ObjectServer on behalf of the master probe. When the master is running again, the slave will continue to receive events, but will no longer send them to the ObjectServer.

Note : Peer-to-peer functionality does not work if you specify the same value for the **JmsFilter** property for both the master probe and the slave probe files. You must create two queues that both use the JMS filter that you want to use, and specify one queue in the master probe properties file and specify the other queue in the slave probe properties file. You must also specify different values for the **PersistentJmsID** property on the master and slave properties.

Example property file settings for peer-to-peer failover

The following settings show the peer-to-peer settings from the properties file of an example master probe:

```
PidFile      : "master_pid_file"
Server       : "NCOMS"
RulesFile    : "master_rules_file"
MessageLog   : "master_log_file"
PeerHost     : "slave_hostname"
PeerPort     : 5555 # [communication port between master and slave probes]
Mode        : "master"
JmsFilter: ALA_clientId in ('netcool@master@127.0.0.1', '') and ALA_category not in
('STATISTICS', 'ACCOUNTING')
PersistentJmsId : netcool@master@127.0.0.1
```

The following settings show the peer-to-peer settings from the properties file of the corresponding slave probe:

```
PidFile      : "slave_pid_file"
Server       : "NCOMS"
RulesFile    : "slave_rules_file"
```

```
MessageLog : "slave_log_file"
PeerHost   : "master_hostname"
PeerPort   : 5555 # [communication port between master and slave probes]
Mode       : "slave"
JmsFilter: ALA_clientId in ('netcool@slave@127.0.0.1', '') and ALA_category not in
('STATISTICS', 'ACCOUNTING')
PersistentJmsId : netcool@slave@127.0.0.1
```

Note : The properties files also contain all other properties required to configure the probe.

Command line interface

The probe is supplied with a command line interface (CLI) that allows you to manage the probe while it is running. For IBM Tivoli Netcool/OMNIBus V7.4 and later, use the HTTP/HTTPS command interface.

HTTP/HTTPS command interface

IBM Tivoli Netcool/OMNIBus Version 7.4.0 (and later) includes a facility for managing the probe over an HTTP/HTTPS connection. This facility uses the **nco_http** utility supplied with Tivoli Netcool/OMNIBus.

The HTTP/HTTPS command interface replaces the Telnet-based command line interface used in previous versions of IBM Tivoli Netcool/OMNIBus.

The following sections show:

- How to configure the command interface.
- The format of the **nco_http** command line.
- The format of the individual probe commands.
- The messages that appear in the log files.
- How to store frequently-used commands in a properties file.

For more information on the HTTP/HTTPS command interface and the utilities it uses, see the chapter on remotely administering probes in the *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide*.

Configuring the command interface

To configure the HTTP/HTTPS command interface, set the following properties in the probe's property file:

NHttpd.EnableHTTP: Set this property to `True`.

NHttpd.ListeningPort: Set this property to the number of the port that the probe uses to listen for HTTP commands.

Optionally, set a value for the following property as required:

NHttpd.ExpireTimeout: Set this property to the maximum time (in seconds) that the HTTP connection remains idle before it is disconnected.

The *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide* contains a full description of these and all properties for the HTTP/HTTPS command interface.

Format of the nco_http command line

The format of the **nco_http** command line to send a command to the probe is:

```
$OMNIHOME/bin/nco_http -uri probeuri:probeport/probes/generic_3gpp -datatype
application/json -method post -data '{"command":"command-name","params":
[command-parameters]}'
```

Where:

- *probeuri* is the URI of the probe.
- *probeport* is the port that the probe uses to listen for HTTP/HTTPS commands. Specify the same value as that set for the **NHttpd.ListeningPort**.

- *command-name* is the name of the command to send to the probe. The following command names are available:

```

help
name
registerNotification
registerNotificationFile
resync
shutdownprobe
version

```

- *command-parameters* is a list of zero or more command parameters. For commands that have no parameters, this component is empty. The command descriptions in the following section define the parameters that each takes.

Probe commands

The following sections define the structure of the JavaScript Object Notation (JSON)-formatted commands that you can send to the probe. There is an example of each command.

All the examples use a probe URI of `http://localhost` and a HTTP listening port of 8080.

help

Use the **help** command to receive help information about the HTTP/HTTPS command interface.

The format of the **-data** option for the **help** command is:

```
-data '{"command":"help","params":[]}'
```

The following command returns help information:

```
$OMNIHOME/bin/ncp_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -
datatype application/JSON -method POST -data '{"command":"help","params":[]}'
```

name

Use the **name** command to display the name of the probe.

The format of the **-data** option for the **name** command is:

```
-data '{"command":"name","params":[]}'
```

The following command displays the name of the probe:

```
$OMNIHOME/bin/ncp_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -
datatype application/JSON -method POST -data '{"command":"name","params":[]}'
```

registerNotification

Use the **registerNotification** command to register the advanced JMS filter XML file specified by the **AdvancedFilterConfigFile** property.

The format of the **-data** option for the **registerNotification** command is:

```
-data '{"command":"registerNotification","params":[]}'
```

The following command registers the advanced JMS filter XML file specified by the **AdvancedFilterConfigFile** property.:

```
$OMNIHOME/bin/ncp_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -
datatype application/JSON -method POST -data
'{"command":"registerNotification","params":[]}'
```

registerNotificationFile

Use the **registerNotificationFile** command to register an alternative JMS filter XML file.

The format of the **-data** option for the **registerNotificationFile** command is:

```
-data '{"command":"registerNotificationFile", "params":  
[{"XML_config_file_path":"alternative_jms_filter"}]}'
```

Where *alternative_jms_filter* is the path to the alternative JMS filter XML file that you want to register.

The following command registers the file `/opt/IBM/tivoli/netcool/omnibus/var/registerNotifcaton.txt` as the alternative JMS filter XML file:

```
$OMNIHOME/bin/ncnco_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -  
datatype application/JSON -method POST -data  
'{"command":"registerNotificationFile", "params":  
[{"XML_config_file_path":"/opt/IBM/tivoli/netcool/omnibus/var/  
registerNotifcaton.txt"}]}'
```

resync

Use the **resync** command to perform a resynchronization using the parameters specified by the following properties in the properties file:

- **XMLretrieveUseInService**
- **SamServerUserName**
- **SamServerPassword**

The format of the **-data** option for the **resync** command is:

```
-data '{"command":"resync", "params": []}'
```

The following command performs a resynchronization:

```
$OMNIHOME/bin/ncnco_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -  
datatype application/JSON -method POST -data '{"command":"resync", "params": []}'
```

shutdownprobe

Use the **shutdownprobe** command to shut down the probe.

The format of the **-data** option for the **shutdownprobe** command is:

```
-data '{"command":"shutdownprobe", "params": []}'
```

The following command shuts down the probe:

```
$OMNIHOME/bin/ncnco_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -  
datatype application/JSON -method POST -data '{"command":"shutdownprobe",  
"params": []}'
```

version

Use the **version** command to print the version of the probe.

The format of the **-data** option for the **version** command is:

```
-data '{"command":"version", "params": []}'
```

The following command returns version information:

```
$OMNIHOME/bin/ncnco_http -uri http://localhost:8080/probes/alcatel_5620_sam_v13 -  
datatype application/JSON -method POST -data '{"command":"version", "params":  
[]}'
```

Messages in the log file

The `nco_http` utility can make extensive entries in the probe's log file indicating the progress of each operation. These messages can help isolate problems with a request, such as a syntax problem in a command.

To obtain the detailed log information, set the probe's **MessageLevel1** property to debug. This enables the logging of the additional information that tracks the progress of a command's execution. For example, the following shows the progress of a **resync** command:

```
Information: I-UNK-104-002: {"response":["Resync successful"],"status":"200"}
```

Storing commands in the `nco_http` properties file

You can use the `nco_http` utility's properties file (`$OMNIHOME/etc/nco_http.props`) to hold frequently used command characteristics.

If you have a particular command that you send to the probe regularly, you can store characteristics of that command in the `nco_http` properties file. Once you have done that, the format of the `nco_http` command line is simplified.

You can use one or more of the following `nco_http` properties to hold default values for the equivalent options on the `nco_http` command line:

Data
DataType
Method
URI

Specify the value of each property in the same way as you would on the command line. Once you have these values in place you do not need to specify the corresponding command line switch unless you want to override the value of the property.

The following is an example of the use of the properties file and the simplification of the `nco_http` command that results. In this example, the `nco_http` properties file contains the following values (note that line breaks appear for presentational purposes only; when editing the properties use one line for each property value):

```
Data : [example required]  
DataType : 'application/JSON'  
Method : 'POST'
```

Properties and command line options

You use properties to specify how the probe interacts with the device. You can override the default values by using the properties file or the command line options.

The following table describes the properties and command line options specific to this probe. For more information about generic Netcool/OMNIbus properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Table 7. Properties and command line options

Property name	Command line option	Description
AdvancedFilterConfigFile <i>string</i>	-advancedfilter configfile <i>string</i>	Use this property to specify the path of an advanced JMS filter XML file. The Alcatel-Lucent 5620 SAM server uses this file to determine which events to send to the probe when the JmsTopic property is set to 5620-SAM-topic-xml-filtered. For details about how to dynamically change the advanced JMS filter while the probes is running, see “Subscribing to the 5620-SAM-topic-xml-filtered topic” on page 16. On UNIX and Linux operating systems, the default is \$OMNIHOME/var/registerNotification.txt. On Windows operating systems, the default is %OMNIHOME%\var\registerNotification.txt.
CertificateStore <i>string</i>	-certificatestore <i>string</i>	Use this property to specify the path of the certificate store file. The default is "".
CertificateStorePassword <i>string</i>	-certificatestore password <i>string</i>	Use this property to specify the password required to access the certificate store file. The default is "".
CertificateStoreType <i>string</i>	-certificatestore type <i>string</i>	Use this property to specify the certificate type obtained from the certificate store. The default is JKS.
Durable <i>string</i>	-durable <i>string</i>	Use this property to specify that the probe makes a durable subscription to the JMS. false: JMS subscription is not durable. true: JMS subscription is durable. The default is true.
EJBPort <i>integer</i>	-ejbport <i>integer</i>	Use this property to specify the EJB port to which the probe connects. The default is 1099.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
EnableFailover <i>string</i>	-noenablefailover (This is equivalent to EnableFailover with a value of false.) -enablefailover (This is equivalent to EnableFailover with a value of true.)	Use this property to specify an instance of probe to support redundant SAM servers. The default is false. If EnableFailover is set to true then the RetryCount must be set to a value greater than 0.
Host <i>string</i>	-host <i>string</i>	Use this property to specify the IP address of the host machine on which the Alcatel-Lucent 5620 SAM server is running. The default is localhost.
HTTPPort <i>integer</i>	-httpport <i>integer</i>	Use this property to specify the HTTP port to which the probe connects. The default is 8080.
JMSAcknowledgeMode <i>string</i>	-jmsacknowledge mode <i>string</i>	Use this property to specify the acknowledgment mode in which the probe runs. This property takes the following values: AUTO_ACKNOWLEDGE: The probe runs in automatic acknowledgement (AUTO_ACKNOWLEDGE) mode. DUPS_OK_ACKNOWLEDGE: The probe runs in deduplication allowed (DUPS_OK_ACKNOWLEDGE) mode. The default is DUPS_OK_ACKNOWLEDGE
JmsFilter <i>string</i>	-jmsfilter <i>string</i>	Use this property to specify the filter the probe uses to limit the events received from Alcatel-Lucent 5620 SAM. For details of the format in which to specify this filter, see “Filtering events” on page 17. The default value is ALA_clientId in ('netcool@127.0.0.1', '') and ALA_category not in (\'STATISTICS\', \'ACCOUNTING\').
JmsTopic <i>string</i>	-jmstopic <i>string</i>	Use this property to specify the XML topic that the probe uses to subscribe to events within the JMS. For details about valid values for this property, see Table 5 on page 15. The default is 5620-SAM-topic-xml.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
Optimize <i>integer</i>	-optimize <i>integer</i>	Use this property to specify whether the probe optimizes performance: 0: The probe does not optimize performance. 1: The probe optimizes performance by writing neither events received from the HTTP request nor JMS events to the log file. The default is 1.
ParseSaxDebug <i>integer</i>	-pasesaxdebug <i>integer</i>	Use this property to specify whether the probe outputs extra debug information from the SAX Parser. The default is 0. Note : Use this option for debug purposes only.
PersistentJmsId <i>string</i>	-persistentjmsid <i>string</i>	Use this property to specify the ID of the JMS subscription. The default is netcool@127.0.0.1.
RecoveryFile <i>string</i>	-recoveryfile <i>string</i>	Use this property to specify the name of the recovery file in which the probe stores the timestamp of the last keepAlive event read before disconnecting from the Alcatel-Lucent 5620 SAM server. The probe uses this when reconnecting to the server. On UNIX and Linux operating systems, the default is \$OMNIHOME/var/AlcatelSamRecovery. On Windows operating systems, the default is %OMNIHOME%\var\AlcatelSamRecovery. This recovery file does not exist by default. You must create this file manually when this property is specified for the first time. Note : You should backup or remove the recovery file when switching to a new machine or upgrading to a newer version of the probe. Failure to do so may result in successive resynchronizations but with missing events.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
ResyncBatchSize <i>integer</i>	-resyncbatchsize <i>integer</i>	Use this property to specify the maximum number of alarms that the probe retrieves in each batch of resynchronization alarms. The default is 100.
SamServerPassword <i>string</i>	-samserverpassword <i>string</i>	Use this property to specify the password required with the SamServerUserName property to log in to the JMS. The default is " ".
SamServerUserName <i>string</i>	-samserverusername <i>string</i>	Use this property to specify the user name with which the probe logs in to the JMS. The default is " ".
SecondaryEJBPort <i>integer</i>	-secejbport <i>integer</i>	Use this property to specify the EJB port of secondary host machine to which the probe connects. The default is 1099.
SecondaryHost <i>string</i>	-sechost <i>string</i>	Use this property to specify the IP address of the secondary host machine on which the Alcatel-Lucent 5620 SAM server is running.
SecondaryHTTPPort <i>integer</i>	-sechttpport <i>integer</i>	Use this property to specify the HTTP port of the secondary host machine to which the probe connects. The default is 8080.
StoreEvents <i>integer</i>	-storeevents <i>integer</i>	Use this property to specify whether Alcatel SAM continues to store events after the probe has been stopped by a CTRL - C command. This property takes the following values: 0: Alcatel SAM does not store events after a CTRL - C stop. 1: Alcatel SAM stores events after a CTRL - C stop. The default is 1.
TrustStore <i>string</i>	-truststore <i>string</i>	Use this property to specify the file path of the truststore file. The default is " ".

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
TrustStorePassword <i>string</i>	- truststorepassword <i>string</i>	Use this property to specify the password required to access the truststore file containing the trusted certificates. The default is " ".
UseSSL <i>string</i>	- usessl <i>string</i>	Use this property to enable SSL authentication when connecting to the Alcatel-Lucent 5620 SAM server. This property takes the following values: false: SSL is not used when connecting to the Alcatel-Lucent 5620 SAM server. true: SSL is used when connecting to the Alcatel-Lucent 5620 SAM server. The default is false. Note : If UseSSL is set to true, the HTTPPort property must be configured to use the Alcatel SAM HTTPS port. The default Alcatel SAM HTTPS port is 8443. You must also configure the following properties for the Probe for Alcatel-Lucent 5620 SAM v13: <ul style="list-style-type: none"> • CertificateStoreType • CertificateStore • CertificateStorePassword • TrustStore • TrustStorePassword
XMLFile <i>string</i>	- xmlfile <i>string</i>	Use this property to specify the XML file that the probe reads to check the SAX parser. The default is " " . Note : Use this option for debug purposes only.
XMLFileRead <i>integer</i>	- xmlread <i>integer</i>	Use this property to specify whether the probe reads an XML file to check the SAX parser. This property takes the following values: 0: The probe does not read an XML file. 1: The probe reads the XML file specified by the XMLFile property. The default is 0. Note : Use this option for debug purposes only.

Table 7. Properties and command line options (continued)

Property name	Command line option	Description
XMLretrieveUseInService <i>integer</i>	-xmlretrieveusein service <i>integer</i>	Use this property to specify whether the probe receives alarms from the nodes that are in either <code>inService</code> or <code>inMaintenance</code> mode. This property takes the following values: 0: The probe receives alarms from all the nodes. 1: The probe receives alarms only from the nodes that are in the <code>inService</code> mode. The default is 0.

Properties and command line options provided by the Java Probe Integration Library (probe-sdk-java) version 9.0

All probes can be configured by a combination of generic properties and properties specific to the probe.

The following table describes the properties and command line options that are provided by the Java Probe Integration Library (probe-sdk-java) version 9.0.

Note : Some of the properties listed may not be applicable to your probe.

Table 8. Properties and command line options

Property name	Command line option	Description
CommandPort <i>integer</i>	-commandport <i>integer</i>	Use this property to specify the port to which users can Telnet to communicate with the probe using the Command Line Interface (CLI) supplied. The default is 6970.
CommandPortLimit <i>integer</i>	-commandportlimit <i>integer</i>	Use this property to specify the maximum number of Telnet connections that can be made to the probe. The default is 10.
DataBackupFile <i>string</i>	-databackupfile <i>string</i>	Use this property to specify the path to the file that stores data between probe sessions. The default is "". Note : Specify the path relative to <code>\$OMNIHOME/var</code> .
DisconnectionTimeout <i>integer</i>	-disconnectiontimeout <i>integer</i>	Use this property to specify the maximum time, in seconds, for probe disconnection before shutting down the probe forcefully. The default is 15.

Table 8. Properties and command line options (continued)

Property name	Command line option	Description
HeartbeatInterval <i>integer</i>	<code>-heartbeatinterval</code> <i>integer</i>	Use this property to specify the frequency (in seconds) with which the probe checks the status of the host server. The default is 1.
Inactivity <i>integer</i>	<code>-inactivity</code> <i>integer</i>	Use this property to specify the length of time (in seconds) that the probe allows the port to receive no incoming data before disconnecting. The default is 0 (which instructs the probe to not disconnect during periods of inactivity).
InactivityAction <i>string</i>	<code>-inactivityaction</code> <i>string</i>	Use this property to specify the action the probe takes when inactivity timeout is reached. SHUTDOWN: Sends a ProbeWatch message to notify user and shuts down the probe. CONTINUE: Sends a ProbeWatch message to notify user and do not shut down the probe. The default is SHUTDOWN.
InitialResync <i>string</i>	<code>-initialresync</code> <i>string</i>	Use this property to specify whether the probe requests all active alarms from the host server on startup. This property takes the following values: <code>false</code> : The probe does not request resynchronization on startup. <code>true</code> : The probe requests resynchronization on startup. For most probes, the default value for this property is <code>false</code> . If you are running the JDBC Probe, the default value for the InitialResync property is <code>true</code> . This is because the JDBC Probe only acquires data using the resynchronization process.

Table 8. Properties and command line options (continued)

Property name	Command line option	Description
MaxEventQueueSize <i>integer</i>	<code>-maxeventqueue sizeinteger</code>	<p>Use this property to specify the maximum number of events that can be queued between the non native process and the ObjectServer.</p> <p>The default is 0.</p> <p>Note : You can increase this number to increase the event throughput when a large number of events is generated.</p>
ResyncInterval <i>integer</i>	<code>-resyncinterval integer</code>	<p>Use this property to specify the interval (in seconds) at which the probe makes successive resynchronization requests.</p> <p>For most probes, the default value for this property is 0 (which instructs the probe to not make successive resynchronization requests).</p> <p>If you are running the JDBC Probe, the default value for the ResyncInterval property is 60. This is because the JDBC Probe only acquires data using the resynchronization process.</p>
RetryCount <i>integer</i>	<code>-retrycount integer</code>	<p>Use this property to specify how many times the probe attempts to retry a connection before shutting down.</p> <p>The default is 0 (which instructs the probe to not retry the connection).</p>
RetryInterval <i>integer</i>	<code>-retryinterval integer</code>	<p>Use this property to specify the length of time (in seconds) that the probe waits between successive connection attempts to the target system.</p> <p>The default is 0 (which instructs the probe to use an exponentially increasing period between successive connection attempts, for example, the probe will wait for 1 second, then 2 seconds, then 4 seconds, and so forth).</p>

Elements

The probe breaks event data down into tokens and parses them into elements. Elements are used to assign values to ObjectServer fields; the field values contain the event details in a form that the ObjectServer understands.

This section describes the elements that the Probe for Alcatel-Lucent 5620 SAM v13 JMS generates.

Common elements

The probe generates a common set of elements for each event that it receives.

The following table describes the elements that are generated for all events.

Element name	Element description
\$ALA_allomorphic	This element contains the allomorphic class of the object.
\$ALA_category	This element shows the category of the message, for example, FAULT.
\$ALA_clientId	This element contains the client identifier for the message.
\$correlationId	This element contains the correlation identifier for the related messages.
\$eventName	This element shows the type of event, for example, an attribute value change.
\$MTOSI_NTType	This element shows the notification type of the message, for example, NT_ALARM.
\$MTOSI_objectName	This element contains the object full name associated with the event.
\$MTOSI_objectType	This element indicates the object type for the message, for example, fm.AlarmObject.
\$MTOSI_osTime	This element shows the time that the message was created.

Alarm event elements

The probe generates a standard set of elements for each alarm event that it receives.

The following table describes the elements that are generated for alarm event elements.

Element name	Element description
\$ALA_alarmType	This element indicates the alarm type.
\$ALA_OLC	This element indicates the OLC state of the object associated to the event. A value of zero (0) indicates no state for the object.
\$ALA_application	This element shows the tag associated with the application.
\$ALA_span	This element contains the span of control ID list.
\$ALA_isCorr	This element shows the <code>isCorrelated</code> flag. A value of true indicates that the alarm is correlated at the creation time to another alarm.

<i>Table 10. Alarm event elements (continued)</i>	
Element name	Element description
\$MTOSI_aliasNameList	This element indicates the alias for the alarm name.
\$MTOSI_perceivedSeverity	This element indicates the severity of the alarm.
\$MTOSI_probableCause	This element identifies the probable cause of the alarm.
\$MTOSI_serviceAffecting	This element identifies whether the alarm is service affecting.

Attribute value change event elements

An attribute value change alarm indicates changes to a particular problem alarm that the probe received earlier. Based on the indicated changes, the probe generates a standard set of elements for each attribute value change event that it receives.

The probe performs the following if the attribute value change alarm indicates that the severity field is updated with a value other than cleared in Attribute4:

- Deduplicates the problem alarm.
- Updates severity of the problem alarm with the latest value.
- Increments the count of the problem alarm by 1.

The probe performs the following if the attribute value change alarm indicates that the severity field is updated with a cleared value in Attribute4:

- Makes the attribute value change alarm as a new resolution alarm in the event list. This action is performed as per the Netcool/OMNIBus generic clear standards.
- Clears both the problem alarm and the resolution alarm.

It is possible that the probe may receive another attribute value change alarm before the problem alarm was cleared. If this new attribute value change alarm indicates that the severity field is updated with a value other than cleared in Attribute4, then the probe performs the following:

- Deduplicates the problem alarm.
- Updates severity of the problem alarm with the latest value.
- Increments the count of the problem alarm by 1.

It is The probe performs the following if the attribute value change alarm indicates that attributes other than the severity field are updated in Attribute4:

- Deduplicates the problem alarm.
- Keeps the severity field with its current value.
- Increments the count of the problem alarm by 1.

The following table describes the elements that are generated for attribute value change events.

<i>Table 11. Attribute value change event elements</i>	
Element name	Element description
\$avc_n	This element shows the attribute changed and its associated value.

<i>Table 11. Attribute value change event elements (continued)</i>	
Element name	Element description
\$objectFullName	This element contains the name of the object whose attributes changed.

State change event elements

The probe generates a standard set of elements for each state change event that it receives.

The following table describes the elements that are generated for state change event elements.

<i>Table 12. State change event elements</i>	
Element name	Element description
\$maxAlarmCount	This element shows the maximum alarm count.
\$state	This element indicates the state that changed.
\$sysPrimaryIp	This element shows the primary IP address of the system.
\$sysStandbyIp	This element shows the standby IP address of the system.
\$sysStartTime	This element shows the time that the system started up.

Alarm status change event elements

The probe generates a standard set of elements for each alarm status change event that it receives.

The following table describes the elements that are generated for alarm status change event elements.

<i>Table 13. Alarm status change event elements</i>	
Element name	Element description
\$Attributen	This element contains the attribute changed and its associated value.
\$objectFullName	This element shows the name of the object whose alarm status has changed.
\$olcState	This element indicates whether the alarm is generated when the node is in one of these modes: inService or inMaintenance .
\$spanObjectPointer	This element identifies the span of object pointer.
\$correlatingAlarm	This element indicates whether the alarm has a correlating alarm.
\$isImplicitlyCleared	This element indicates whether the alarm is implicitly cleared.
\$numberOfCorrelatedAlarms	This element shows the number of alarms correlated to the alarm.

Object deletion event elements

The probe generates a single element for each database proxy state change event that it receives.

The following table describes the element that is generated for object deletion event elements.

<i>Table 14. Object deletion event elements</i>	
Element name	Element description
\$objectFullName	This element shows the name of the object deleted.

Relationship change event elements

The probe generates a standard set of elements for each relationship change event that it receives.

The following table describes the elements that are generated for relationship change event elements.

<i>Table 15. Relation change event elements</i>	
Element name	Element description
\$changeType	This element shows the type of change.
\$fromObjectClass	This element contains the class of the original object.
\$fromObjectName	This element identifies the name of the original object.
\$objectFullName	This element identifies the name of the object with a changed relationship.
\$toObjectClass	This element contains the class of the new object.
\$toObjectName	This element shows the name of the new object.

Elements generated for object creation events

The probe generates a standard set of elements for each object creation event that it receives.

The following table describes the elements generated when alarm information objects are created.

<i>Table 16. Elements for object creation events</i>	
Element name	Element description
\$acknowledgedBy	This element contains the name of the operator that acknowledged the alarm.
\$additionalText	This element contains the additional descriptive information about the alarm.
\$affectedObjectClassIndex	This element indicates the identifier of the managed object class.
\$affectedObjectClassName	This element contains the name of the managed object class to which the object with the alarm belongs.

Table 16. Elements for object creation events (continued)

Element name	Element description
\$affectedObjectDisplayedName	This element contains the text version name of the alarm's affected object.
\$affectedObjectFullName	This element contains the fully distinguished name of the object against which the alarm was raised.
\$affectedObjectInstanceIndex	This element indicates the identifier of the managed object instance.
\$alarmClassTag	This element shows the object class against which the alarm was raised.
\$alarmName	This element contains the alarm name.
\$alarmType	This element shows the vendor-specific and X.733 standards for the type of the alarm.
\$firstTimeDetected	This element shows the date and time the alarm was first raised.
\$highestSeverity	This element identifies the highest severity value of the alarm since it was generated.
\$isAcknowledged	This element indicates whether the alarm is currently acknowledged by an operator.
\$isServiceAffecting	This element indicates whether an alarm has affected a customer's service.
\$lastTimeAcknowledged	This element shows the date and time at which the alarm was acknowledged.
\$lastTimeCleared	This element shows the date and time at which the alarm was cleared.
\$lastTimeDeEscalated	This element shows the date and time at which the alarm was de-escalated.
\$lastTimeDemoted	This element shows the date and time at which the alarm was demoted.
\$lastTimeDetected	This element shows the date and time the alarm was most recently observed.
\$lastTimeEscalated	This element shows the date and time at which the alarm was escalated.
\$lastTimePromoted	This element shows the date and time at which the alarm was promoted.

Table 16. Elements for object creation events (continued)

Element name	Element description
\$lastTimeSeverityChanged	This element shows the date and time at which the alarm was cleared.
\$nodeId	This element contains the IP address of the router that issued the alarm and contains the affected object.
\$nodeName	This element contains the text version name of the router that issued the alarm and contains the affected object.
\$numberOfOccurrences	This element indicates the number of times the alarm has been raised.
\$numberOfOccurrencesSinceAck	This element indicates the number of times the alarm has been raised since it was acknowledged.
\$numberOfOccurrencesSinceClear	This element indicates the number of times the alarm has been raised since it was cleared.
\$operatorAssignedUrgency	<p>This element indicates the urgency setting of the alarm as determined by the user-configured value set for the alarm. The possible values are:</p> <ul style="list-style-type: none"> 0: Unspecified 1: Indeterminate 2: Minor 3: Major 4: Critical
\$originalSeverity	This element shows the severity of the alarm when it was first raised.
\$previousSeverity	This element shows the severity value of the alarm previous to the current value.
\$probableCause	This element shows the vendor-specific and X.733 standards for the probable cause of the alarm.
\$relatedObject	This element contains the list of fully distinguished names that point to other objects that are also affected by a particular alarm.
\$specificProblem	<p>This element indicates the specific problem that the alarm is reporting. It is generated when the node reports vendor specific information in the <code>specificProblem</code> field. The probe populates this element by referring to the <code>alcatel_5620_sam_v13.lookup</code> table supplied with the probe. For details about configuring the lookup table, see “Configuring lookup tables” on page 7.</p>

Table 16. Elements for object creation events (continued)

Element name	Element description
\$severity	This element shows the vendor-specific, TMN, and X.733 standards for severity of the alarm. The possible values are: 0: Unspecified 1: Cleared 2: Indeterminate 3: Info 4: Condition 5: Warning 6: Minor 7: Major 8: Critical
\$urgencyAssignedBy	This element contains the name of the operator that last modified the urgency of the alarm.
\$userText	This element indicates custom user text of the alarm.
\$wasAcknowledged	This element indicates whether the alarm has ever been acknowledged by an operator.

Customizing the rules file to populate the \$specificProblem element

The \$specificProblem element is available for SAM v13 for certain types of alarms, for example NobeB alarms. To populate the \$specificProblem element you need to customize the `alcatel_5620_sam_v13.rules` file.

The probe rules file contains the following lines that define the specificProblem element:

```
#specificProblem is only available starting SAM v12 and above; and applicable
for certain alarm type only (eg: NodeB alarm)
if (exists($specificProblem))
{
    $SpecificProblem_T = lookup($specificProblem, SpecificProblem_t)
    ### Remove hash below to map specific problem with object server field
    ### @X733SpecificProb = $SpecificProblem_T
}
```

To make the \$specificProblem element available, uncomment the following line by removing the hash symbols:

```
@X733SpecificProb = $SpecificProblem_T
```

This will map \$SpecificProblem to the @X733SpecificProb field.

Alternatively, you can create a new field in the ObjectServer using Netcool Administrator and map \$SpecificProblem_T with a new field (for example @NewField).

```
@NewField = $SpecificProblem_T
```

Client session event elements

The probe generates a single element for each client session event that it receives.

The following table describes the element that is generated for client session event elements.

<i>Table 17. Client session event elements</i>	
Element name	Element description
\$clientId	This element shows the identifier of the client in this session.

File available event elements

The probe generates a standard set of elements for file available events that it receives.

The following table describes the elements that are generated for file available event elements.

<i>Table 18. File available event elements</i>	
Element name	Element description
\$fileName	This element contains the name of the file required.
\$requestId	This element shows the identifier of the request.

Database activity event elements

The probe generates a single element for each database activity event that it receives.

The following table describes the element that is generated for database activity event elements.

<i>Table 19. Database activity event elements</i>	
Element name	Element description
\$state	This element shows the state associated with the database activity.

Database connection state change event elements

The probe generates a single element for each database connection state change event that it receives.

The following table describes the element that is generated for database connection state change event elements.

<i>Table 20. Database connection state change event elements</i>	
Element name	Element description
\$state	This element indicates the state of the database connection.

Database error event elements

The probe generates a standard set of elements for each database error event that it receives.

The following table describes the elements that are generated for database error events.

<i>Table 21. Database error event elements</i>	
Element name	Element description
\$error	This element shows the description of the error.
\$state	This element identifies the database error state.

Database proxy state change event elements

The probe generates a single element for each database proxy state change event that it receives.

The following table describes the element that is generated for database proxy state change events.

<i>Table 22. Database proxy state change event elements</i>	
Element name	Element description
\$state	This element indicates the state of the database proxy.

Error messages

Error messages provide information about problems that occur while running the probe. You can use the information that they contain to resolve such problems.

The following table describes the error messages specific to this probe. For information about generic Netcool/OMNIbus error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

<i>Table 23. Error messages</i>		
Error	Description	Action
Failed to get properties	The probe could not access the properties file.	Check the location of and permissions set for the properties file.
Failed to make HTTP connection to <i>host</i>	The probe was unable to connect to the port specified by the HTTPEndPoint property.	Check that you have specified the correct port in the properties file.
JNDI API lookup failed	The probe failed to create a <code>topicConnectionFactory</code> object with which to connect to the JMS.	Contact IBM Software Support.
Please check whether you have configured incorrect <code>CertificateStore/TrustStore</code> password.	The JNDI API lookup failed due to an incorrect certificate or truststore password.	Check that you have configured the CertificateStore and TrustStore passwords correctly.
Please check whether you have configured correct <code>TrustStore</code> & <code>TrustStorePassword</code> in probe.	The JNDI API lookup failed due to missing truststore configuration.	Check that you have configured the TrustStore and TrustStorePassword properties correctly.

Table 23. Error messages (continued)

Error	Description	Action
It looks like the JMS is configured to use SSL, and the probe is not properly configured to use it. Please double-check the probe configuration.	The SAM JMS server has been configured to use SSL, but the probe is not properly configured to use it.	Check the probe configuration: Make sure the UseSSL property is set to <code>true</code> and that the related SSL properties have been configured correctly.
Please ensure you have configure CertificateStore & CertificateStorePassword correctly in probe.	The JNDI API lookup failed due to missing certificate store configuration.	Check that you have configured the CertificateStore and CertificateStorePassword properties correctly.
Please check whether you have Host & EJBPort configured correctly in probe.	The JNDI API lookup failed due to a NoRouteToHost exception.	Check that you have configured the TrustStore and TrustStorePassword properties correctly.

ProbeWatch messages

During normal operations, the probe generates ProbeWatch messages and sends them to the ObjectServer. These messages tell the ObjectServer how the probe is running.

The following table describes the ProbeWatch messages that the probe generates. For information about generic Netcool/OMNIBus ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIBus Probe and Gateway Guide*.

Table 24. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes
Could not find JNDI context topic <i>topic_name</i>	There was an error when the probe attempted to connect to the Alcatel-Lucent 5620 SAM server.	The JMS did not start properly on the Alcatel-Lucent 5620 SAM server.
Failed to establish HTTP connection for resync	The probe has failed to establish an HTTP connection to the Alcatel-Lucent 5620 SAM server.	The probe failed to connect to the port specified by the HTTPPort property.
Going Down	The probe is shutting down.	The probe is shutting down after performing the shutdown routine.
JNDI API lookup failed	There was an error when the probe attempted to connect to the Alcatel-Lucent 5620 SAM server.	The JMS did not start properly on the Alcatel-Lucent 5620 SAM server.
Running ...	The probe is running normally.	The probe has just been started up.

Table 24. ProbeWatch messages (continued)

ProbeWatch message	Description	Triggers/causes
Unsupported Java class version error	An unsupported Java class version exception occurred while the probe was initializing the JMS connection to the EMS.	The server is running a version of Java other than 1.7
xmlFilterChangeEvent received, advanced filtering registration is successful	The probe successfully completed an advanced filtering registration.	The probe received an xmlFilterChangeEvent from the Alcatel-Lucent 5620 SAM server.
Failed to register advanced filtering	This indicates that the advanced filtering failed to register.	The SAM JMS server returned an exception status of 1. Check the advanced XML filter specified by the AdvancedFilterConfigFile property and retry running the probe.
Some classes are missing from the CLASSPATH. Exiting.	The probe could not find the necessary class during connection.	Check that you have copied the necessary JAR files from the Alcatel-Lucent 5620 SAM v13 server into the Netcool/OMNIBus environment.
Unable to parse event due to IOException	The probe could not parse events due to an IO exception. This may cause event loss.	Contact IBM support for assistance.
Unable to parse event due to SAX Parser exception	The probe could not parse events due to a SaxParser exception. This may cause event loss.	Contact IBM support for assistance.
START SYNCHRONIZATION	A resynchronization operation started.	The probe started receiving active alarms from EMS.
STOP SYNCHRONIZATION	A resynchronization operation completed.	The probe stopped receiving active alarms from EMS.
Connecting new primary SAM server	This probe watch is used to notify user about the probe is setting up connection to new primary 5620 SAM server after detected SAM server activity switch in redundant SAM system.	
Connected to primary SAM server	The probe is connected to primary SAM server.	

Troubleshooting

Various issues arise as users work with the probe. Troubleshooting information is provided to help you diagnose and resolve such issues.

Probe is missing some events and how to resolve it?

Part of the value set for the **JmsFilter** property includes the value set for the **PersistentJmsId** property. So, the probe will not work properly if the **JmsFilter** property is not updated after changing the value of the **PersistentJmsId** property. When the probe encounters this issue, the following debug message appears in the logfile:

```
Debug: SAM: Sleeping until receiving the Sam server's data?
```

To resolve this problem, set the JMS filter and the JMS ID to the same IP address or host name, and not to a combination of local host and IP or host name.

Missing SystemInfoEvent message upon initial JMS connection

The Alcatel-Lucent 5620 SAM v13 server is not currently sending the expected `SystemInfoEvent` message when the probe makes a JMS connection. If you have enabled inactivity processing (that is, set the **Inactivity** property in the probe properties file to a value greater than 0), the probe will shut down when it does not receive the `SystemInfoEvent` message within the number of seconds defined in the **Inactivity** property during the check subscription process.

The workaround for this issue is to disable inactivity processing by setting the **Inactivity** property in the `alcatel_5620_sam_v13.props` file to 0.

JmsMissedEvent messages are not being sent when events have been lost

The Alcatel-Lucent 5620 SAM v13 server is not currently sending the expected `JmsMissedEvents` messages when events have been lost. This means that the probe is not able to force a resynchronization in order to recover the missed events.

There is no workaround for this issue because since the `JmsMissedEvents` message is not sent by the 5620 SAM server, the probe has no way of knowing to send a resynchronization request. If you notice that there is event loss, you should request a resynchronization manually through the command port or HTTP interface to recover the lost events.

Known Issues

At the time of release, a number of known issues were reported that you should be aware of when running the probe.

Modifying the default Netcool/OMNIBus deduplication triggers for the Probe for Alcatel 5620 SAM v13

The `ObjectServer Summary` field may change after receiving an `AttributeValueChange` event, but it should not.

To resolve this issue, run the `modify_dedup_trigger.sql` SQL script provided in the probe's installation package. This script modifies the default Netcool/OMNIBus deduplication triggers for the Probe for Alcatel 5620 SAM v13 to prevent the `@Summary` field from being updated when the `ObjectServer` receives an `AttributeValueChange` event.

The script is located in the following directory:

```
$OMNIHOME/probes/arch
```

Run the script using one of the following commands:

On UNIX and Linux operating systems use:

```
$OMNIHOME/bin/nco_sql -server objectserver_name -user username -password password < path_to_file/modifyt_dedup.trigger.sql
```

On Windows operating systems use:

```
%NCHOME%\bin\redist\isql.exe -S objectserver_name -U username -P password -i path_to_file\modify_dedup_trigger.sql
```

Where :

objectserver_name is the name of the ObjectServer against which you want to run the SQL file.

username is the username with which you log in to the ObjectServer

password is the password associated with the username you specified.

path-to_file is the path to the SQL file.

Disabling Automatic Alarm Deletion under Alarm Settings

If the user selects Disable Automatic Alarm Deletion under Alarm Settings and changes the severity of an event to "cleared" on SAM client GUI then in the event list the severity of the event will be changed to indeterminate.

This is to avoid the event from being deleted by delete_clears trigger. However, if the cleared event is deleted on SAM client GUI then the event will be resolved and removed from event list.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.



SC27-6577-02

