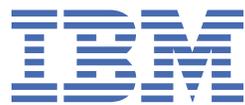


IBM® Tivoli® Netcool/OMNIbus Generic Probe  
for Multi-Technology Operations Systems  
Interface (MTOSI)  
1.0

*Reference Guide*  
*April 12, 2018*



**Notice**

Before using this information and the product it supports, read the information in [Appendix A, “Notices and Trademarks,”](#) on page 21.

**Edition notice**

This edition (SC27-8773-00) applies to version 1.0 of the IBM Tivoli Netcool/OMNIBus Generic Probe for Multi-Technology Operations Systems Interface (MTOSI) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 2018.**

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. Generic Probe for Multi-Technology Operations Systems Interface (MTOSI)..... 1**
  - Summary..... 1
  - Installing probes..... 2
  - Configuring the probe..... 3
    - Configuring SSL connections..... 3
  - MTOSI authentication lock-out..... 4
  - Running the probe..... 4
  - Data acquisition..... 4
    - Specifying filtering operations for resynchronization..... 5
    - Peer-to-peer failover functionality..... 11
  - HTTP/HTTPS command interface..... 11
    - Configuring the command interface..... 12
    - Format of the nco\_http command line..... 12
    - Probe commands..... 12
    - Storing commands in the nco\_http properties file..... 13
  - Properties and command line options..... 13
  - Elements..... 19
  - ProbeWatch messages..... 20
  
- Appendix A. Notices and Trademarks..... 21**
  - Notices..... 21
  - Trademarks..... 22



# 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 Generic Probe for Multi-Technology Operations Systems Interface (MTOSI) documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

<https://www.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/common/Probes.html>

Document version	Publication date	Comments
SC27-8773-00	April 12, 2018	First IBM publication.

## 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.

<b>Operating system</b>	<b>Directory name represented by arch</b>
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. Generic Probe for Multi-Technology Operations Systems Interface (MTOSI)

Multi-Technology Operations System Interface (MTOSI) is a single standard that can connect together services and products from different vendors using different technologies.

The probe acquires XML messages from Webservices interface and stores them as event records in Netcool/OMNIbus ObjectServer. Event notifications and resync requests are both achieved through Webservices call initiated from probe to MTOSI server.

**Note :** The probe is only supported on OMNIbus V8.1 and above.

The probe is described in the following topics:

- [“Summary” on page 1](#)
- <https://www-304.ibm.com/support/docview.wss?uid=swg22014849>
- [“Installing probes” on page 2](#)
- [“Configuring the probe” on page 3](#)
- [“MTOSI authentication lock-out” on page 4](#)
- [“Running the probe” on page 4](#)
- [“Data acquisition” on page 4](#)
- [“Properties and command line options” on page 13](#)
- [“Elements” on page 19](#)
- [“ProbeWatch messages” on page 20](#)

---

## 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 Generic Probe for Multi-Technology Operations Systems Interface (MTOSI).

Probe target	Multi-Technology Operations Systems Interface (MTOSI) Release 2.0
Probe executable name	nco_p_generic_mtosi
Probe installation package	omnibus-arch-probe-nco-p-generic_mtosi-version
Package version	1.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: <a href="https://www-304.ibm.com/support/docview.wss?uid=swg22014849">https://www-304.ibm.com/support/docview.wss?uid=swg22014849</a>
Properties file	\$OMNIHOME/probes/arch/generic_mtosi.props

<i>Table 3. Summary (continued)</i>	
Rules file	\$OMNIHOME/probes/arch/generic_mtosi.rules
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.
Connection method	HTTP
Webservices (XML over HTTP) remote connectivity	The Multi-Technology Operations Systems Interface (MTOSI) can connect to a device on a remote host.
Multicultural support	Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6
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 <a href="http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm">http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm</a> . 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](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](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

---

Before running the probe, you must perform the configuration steps described in this topic.

1. Configuring Probe's Client Component for the following purpose:

- a. Performing subscribe/unsubscribe to EMS by providing user's EMS credentials.

```
HTTPLoginPassword : 'password'
```

```
HTTPLoginUsername : 'username'
```

- b. Performing subscribe for a notification producers.

```
HTTPServiceNotificationProducerURL : 'http://127.0.0.1:8081/nmsnbi/mtosi/NotificationProducer'
```

- c. Performing Resync by calling web service on EMS.

```
HTTPServiceAlarmRetrievalFilterFile : '$OMNIHOME/probes/<arch>/  
generic_mtosi_resync_filter.xml'
```

```
HTTPServiceAlarmRetrievalURL : 'http://127.0.0.1:8081/nmsnbi/mtosi/AlarmRetrieval'
```

2. Configuring Probe's Server Component for receiving incoming event notification:

```
ServerType : 'HTTP'
```

```
HTTPServiceNotificationConsumerIP : '127.0.0.1'
```

```
HTTPServiceNotificationConsumerPath : '/nmsnbi/mtosi/NotificationConsumerService'
```

```
HTTPServiceNotificationConsumerPort : 2000
```

**Note :** Subscribing for notification events is done by the probe's client side component during start up.

## Configuring SSL connections

You can enable Secure Socket Layer (SSL) encryption of data exchanged over HTTP.

To enable SSL encryption, use the following steps:

1. Use the following Java keytool command to generate a key pair for the probe, along with a keystore file and its associated password:

```
keytool -genkey -alias alias_id -keystore $OMNIHOME/java/conf/  
trusted_keystore.jks -storepass password
```

2. When prompted, enter the following details:

- First and Last Name: MTOSI probe at *server\_hostname*
- Organizational Unit: Netcool/OMNIBus Organization IBM
- Organization: IBM
- City: N/A
- State: N/A
- Country: US
- Enter key password: To use the same password that you specified for the keystore password, press ENTER without entering any characters at this prompt.

3. For each server where an instance of the probe is installed, generate an X.509 self-signed certificate for the probe using the following command:

```
keytool -selfcert -alias alias_id -keystore $OMNIHOME/java/conf/  
trusted_keystore.jks -storepass password
```

4. For each server where an instance of the probe is installed, generate and export a certificate file for the probe (*probe\_cert\_filename.cer*) using the following command:

```
keytool -export -alias alias_id -keystore $OMNIHOME/java/conf/  
trusted_keystore.jks -storepass password -file probe_cert_filename.cer
```

Use a unique *alias\_id* value and *probe\_cert\_filename* value for each certificate file.

5. Manually transfer each certificate file generated in step 4 to the MTOSI server.
6. Import each certificate file generated in step 4 to the MTOSI server.
7. Obtain and export the MTOSI system certificate file (*mtosi\_cert\_filename.cer*).
8. For each server where an instance of the probe is installed, import the MTOSI system certificate (*mtosi\_cert\_filename.cer*) using the following command

```
keytool -import -alias alias_id -file mtosi_cert_filename.cer -keystore  
$OMNIHOME/java/conf/trusted_keystore.jks -storepass password
```

9. Specify values for the following probe properties, using the same Java keystore file as both a key store and a trusted certificates store, and using the same password for each file:

- **KeyStore:** Specify the location of the `trusted_keystore.jks` file.
- **KeyStorePassword:** Specify an unencrypted password for the `trusted_keystore.jks` file.
- **TrustStore:** Specify the location of the `trusted_keystore.jks` file.
- **TrustStorePassword:** Specify the same unencrypted password you used for the **KeyStorePassword** property.

**Note :** If the keystore file does not require a password, set the **KeyStorePassword** property to "".

10. For SSL connections over a HTTP interface, the values that you specify for the **HTTPServiceAlarmRetrievalURL** and **HTTPServiceAlarmUpdateURL** properties must begin with `https` instead of the default `http`. For example:

```
https://127.0.0.1:8081/nmsnbi/mtosi/AlarmRetrieval
```

The probe is now enabled to use SSL connections for exchanging data with the MTOSI server over HTTP.

## MTOSI authentication lock-out

---

The MTOSI is normally configured to temporarily lock a user account for period of time if incorrect authentication information is repeatedly provided.

If the Generic Probe for MTOSI is configured to automatically retry authentication, and if the incorrect user name or password is specified, the interface will lock the account. When you have corrected the user name and password, the probe will still need to wait for the lock-out period to expire before the connection can be reestablished.

## Running the probe

---

Probes can be run in a variety of ways. The way you chose depends on a number of factors, including your operating system, your environment, and the any high availability considerations that you may have.

For details about how to run the probe, visit the following page on the IBM Tivoli Knowledge Center:

[http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all\\_probes/wip/concept/running\\_probe.html](http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/concept/running_probe.html)

## 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 Generic Probe for Multi-Technology Operations Systems Interface (MTOSI) gathers events from the interface using Webservices connection, and then stores them as event records in the Netcool/OMNIbus ObjectServer.

Data acquisition is further described in the following topics:

- [“Specifying filtering operations for resynchronization” on page 5](#)

- “HTTP/HTTPS command interface” on page 11
- “Peer-to-peer failover functionality” on page 11

## Specifying filtering operations for resynchronization

During resynchronization, the probe can perform various filtering operations, the details of which are defined in the filter file specified by the **ResyncFilterFile** property.

### Simple filters

Simple filters use the following syntax:

```
<v11:filter>
<v11:attributeNameList>
<v13:attributeName>POSSIBLE_VALUE</v13:attributeName>
</v11:attributeNameList>
</v11:filter>
```

Where *attributeName* is one of the event attributes listed in the table below, and *POSSIBLE\_VALUE* is one of the values listed for that attribute.

The following example code shows how to use the filter file to select for resynchronization all those events that have a *perceivedSeverity* of MINOR:

```
<v11:filter>
<v11:perceivedSeverityList>
<v13:perceivedSeverity>MINOR</v13:perceivedSeverity>
</v11:perceivedSeverityList>
</v11:filter>
```

### More complex filters

Complex filters use the following syntax:

```
<v11:filter>
  <v11:source>AlarmSourceType</v11:source>
  <v11:scope>
    <v12:name>
      <v12:rdn>
        <v12:type></v12:type>
        <v12:value></v12:value>
      </v12:rdn>
    </v12:name>
  </v11:scope>

  <v11:attributeNameList>
    <v13:attributeName>POSSIBLE_VALUE</v13:attributeName>
    <v13:attributeName>POSSIBLE_VALUE</v13:attributeName>
  </v11:attributeNameList>

  <v11:probableCauseList>
    <v14:prc extension="?" qualifier="?">
      <v14:ru>>false</v14:ru>
      <v14:contra>>false</v14:contra>
      <v14:probableCause>PROBABLE_CAUSE</v14:probableCause>
    </v14:prc>
  </v11:probableCauseList>

  <v11:acknowledgeIndication>ACKNOWLEDGEMENT_STATUS</v11:acknowledgeIndication>
</v11:filter>
```

The following example code shows how to use the filter file to select for resynchronization only those alarms generated by MTOSI server, that have a *perceivedSeverity* of either WARNING, or MINOR, and whose probable cause is transmission failure.

```
<v11:filter>
  <v11:source>INTERNAL</v11:source>
  <v11:scope>
    <v12:name>
```

```

    <v12:rdn>
      <v12:type>MD</v12:type>
      <v12:value><Object Name of Network Element></v12:value>
    </v12:rdn>
  </v12:name>
</v11:scope>

<v11:perceivedSeverityList>
  <v13:perceivedSeverity>WARNING</v13:perceivedSeverity>
  <v13:perceivedSeverity>MINOR</v13:perceivedSeverity>
</v11:perceivedSeverityList>

<v11:probableCauseList>
  <v14:prc extension="?" qualifier="?">
    <v14:ru>>false</v14:ru>
    <v14:contra>>false</v14:contra>
    <v14:probableCause>TX_FAIL</v14:probableCause>
  </v14:prc>
</v11:probableCauseList>

<v11:acknowledgeIndication>?</v11:AI_EVENT_UNACKNOWLEDGED>

</v11:filter>

```

## Attribute names and their corresponding values

The following table shows all the attributes that you can use within filter files, and their corresponding possible values.

<i>Table 4. Filter attributes and associated values</i>	
<b>Attribute name</b>	<b>Possible values</b>
AcknowledgeIndicationType	<ul style="list-style-type: none"> <li>• ACKNOWLEDGED</li> <li>• UNACKNOWLEDGED</li> <li>• NOT_APPLICABLE</li> </ul>
AssignedSeverityEnumType	<ul style="list-style-type: none"> <li>• INDETERMINATE</li> <li>• CRITICAL</li> <li>• MAJOR</li> <li>• MINOR</li> <li>• WARNING</li> <li>• NOT_ALARMED</li> <li>• FREE_CHOICE</li> </ul>
EquipmentProtectionGroupTypeEnumType	<ul style="list-style-type: none"> <li>• MINOR_EXT</li> <li>• M_FOR_N</li> <li>• M:N</li> <li>• EPG_1_PLUS_1</li> <li>• EPG_INBD_1_PLUS_1</li> <li>• EPG_1_FOR_N</li> <li>• BPS</li> </ul>

Table 4. Filter attributes and associated values (continued)

Attribute name	Possible values
EquipmentSwitchReasonEnumType	<ul style="list-style-type: none"> <li>• MINOR_EXT</li> <li>• VENDOR_EXT</li> <li>• NOT_APPLICABLE</li> <li>• FAILURE</li> <li>• MANUAL</li> </ul>
G_774_3_APSfunctionEnumType	<ul style="list-style-type: none"> <li>• MINOR_EXT</li> <li>• VENDOR_EXT</li> <li>• G.783</li> <li>• Legacy</li> </ul>
PerceivedSeverity	<ul style="list-style-type: none"> <li>• INDETERMINATE</li> <li>• CRITICAL</li> <li>• MAJOR</li> <li>• MINOR</li> <li>• WARNING</li> <li>• CLEARED</li> </ul>
ProtectionGroupTypeEnumType	<ul style="list-style-type: none"> <li>• MINOR_EXT</li> <li>• PGT_MSP_1_PLUS_1</li> <li>• PGT_MSP_1_FOR_N</li> <li>• PGT_2_FIBER_BLSR</li> <li>• PGT_4_FIBER_BLSR</li> <li>• PGT_1VN</li> <li>• PGT_1P1</li> <li>• PGT_UNKNOWN</li> </ul>
ProtectionTypeEnumType	<ul style="list-style-type: none"> <li>• MINOR_EXT</li> <li>• VENDOR_EXT</li> <li>• MSP_APS</li> <li>• PGT_MSP_1_PLUS_1</li> <li>• PGT_MSP_1_FOR_N</li> <li>• PGT_2_FIBER_BLSR</li> <li>• PGT_4_FIBER_BLSR</li> <li>• PT_SNCP</li> <li>• PT_UNKNOWN</li> </ul>
ServiceAffectingType	<ul style="list-style-type: none"> <li>• UNKNOWN</li> <li>• SERVICE_AFFECTING</li> <li>• NON_SERVICE_AFFECTING</li> </ul>

<i>Table 4. Filter attributes and associated values (continued)</i>	
<b>Attribute name</b>	<b>Possible values</b>
SwitchReasonType	<ul style="list-style-type: none"> <li>• NOT_APPLICABLE</li> <li>• RESTORED</li> <li>• SIGNAL_FAIL</li> <li>• SIGNAL_MISMATCH</li> <li>• SIGNAL_DEGRADE</li> <li>• AUTOMATIC_SWITCH</li> <li>• MANUAL</li> </ul>
ReversionModeType	<ul style="list-style-type: none"> <li>• UNKNOWN</li> <li>• NON_REVERTIVE</li> <li>• REVERTIVE</li> </ul>

The following table shows a list of attributes that you can use within complex filter files, and their corresponding possible values.

<i>Table 5. Complex filter attributes and associated values</i>			
<b>Name</b>	<b>Type</b>	<b>Description</b>	<b>Example</b>
source	<b>AlarmSourceType</b>	<p>Use this attribute to filter alarms based on alarm source type. The attribute can be set to one of the following values:</p> <p>INTERNAL: The probe filters internal alarms from target system based on alarm source type.</p> <p>NOT_APPLICABLE or omitted: The probe does not filter alarm based on alarm source type.</p> <p><b>Note :</b> When query internal alarm from a specific target system, source parameter should set to INTERNAL.</p>	<pre>&lt;v11:source&gt;INTERNAL&lt;/v11:source&gt;</pre>

Table 5. Complex filter attributes and associated values (continued)

Name	Type	Description	Example
scope	<b>NamingAttribute ListType</b>	<p>Use this attribute to filter the alarms based on specific managed elements. If no managed element is specified, alarms from all managed elements are selected.</p> <p><b>Note :</b> The specified value must be the MTOSI object name of a network element.</p> <p>When a network element name is specified, it indicates the equipment from which alarms are queried.</p>	<pre>&lt;v11:scope&gt;   &lt;v12:name&gt;     &lt;v12:rdn&gt;       &lt;v12:type&gt;MD&lt;/v12:type&gt;     &lt;v12:value&gt;       &lt;Object Name of Network Element&gt;     &lt;/v12:value&gt;   &lt;/v12:rdn&gt; &lt;/v12:name&gt; &lt;/v11:scope&gt;</pre> <p>For Example:</p> <pre>&lt;v11:scope&gt;   &lt;v12:name&gt;     &lt;v12:rdn&gt;       &lt;v12:type&gt;MD&lt;/ v12:type&gt;     &lt;v12:value&gt;       &lt;Object Name of Network Element&gt;     &lt;/v12:value&gt;   &lt;/v12:rdn&gt; &lt;/v12:name&gt; &lt;/v11:scope&gt;</pre>
perceived SeverityList	<b>PerceivedSeverity ListType</b>	<p>Use this attribute to filter the alarms based on a specified level of severity. This can be set to one of the following values:</p> <ul style="list-style-type: none"> <li>• CRITICAL</li> <li>• MAJOR</li> <li>• MINOR</li> <li>• WARNING</li> </ul>	<pre>&lt;v11:perceivedSeverityList&gt; &lt;v13:perceivedSeverity&gt;WARNING &lt;v13:perceivedSeverity&gt; &lt;v13:perceivedSeverity&gt;MINOR&lt;v1 3:perceivedSeverity&gt; &lt;/v11:perceivedSeverityList&gt;</pre>
probableCause List	<b>ProbableCause List Type</b>	<p>Use this attribute to filter the alarms based on a list of probable causes. If no probable causes are specified, all alarms will be selected for resynchronization.</p> <p><b>Note :</b> You can specify any of the standard MTOSI probable cause descriptions. For details, see the TM Forum MTOSI web site.</p>	<pre>&lt;v11:probableCauseList&gt;   &lt;v14:prc&gt;     &lt;v14:ru&gt;&gt;false&lt;/v14:ru&gt;   &lt;v14:contra&gt;&gt;false&lt;/v14:contra   &lt;v14:probableCause&gt;TX_FAIL&lt;/v14: probableCause&gt;&lt;/   v14:prc&gt; &lt;/v11:probableCauseList&gt;</pre>

Table 5. Complex filter attributes and associated values (continued)

Name	Type	Description	Example
ru	<b>ProbableCauseList Type</b>	<p>The following sub-attribute is defined for each probable cause:</p> <p>ru: The Remote Unit (RU) parameter is used to indicate the location of the network resource associated with the alarm probable cause:</p> <p>false: The alarm is associated with a local network resource.</p> <p>true: The alarm is associated with a remote network resource.</p>	
contra	<b>ProbableCauseList Type</b>	<p>The following sub-attribute is defined for each probable cause:</p> <p>contra: This parameter is used to distinguish the direction of the signal related alarm:</p> <p>false: The alarm is detected on the signal related to the sink atomic function.</p> <p>true: The alarm is detected on the signal related to the source atomic function.</p>	
acknowledge Indication	<b>Acknowledge IndicationType</b>	<p>Use this attribute to filter the alarms based on the acknowledgement status. The attribute can be set to one of the following values:</p> <p>AI_EVENT_ACKNOWLEDGED: Acknowledged.</p> <p>AI_EVENT_UNACKNOWLEDGED: Unacknowledged.</p> <p>NOT_APPLICABLE: unknown.</p>	<pre>&lt;v11:acknowledgeIndication&gt;AI_EVENT_UNACKNOWLEDGED&lt;/v11:acknowledgeIndication&gt;</pre>

## Peer-to-peer failover functionality

The probe 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 does not forward events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master and any events it receives thereafter are forwarded to the ObjectServer on behalf of the master probe. When the master probe is running again, the slave probe continues to receive events, but no longer sends them to the ObjectServer.

### Example property file settings for peer-to-peer failover

You set the peer-to-peer failover mode in the properties files of the master and slave probes. The settings differ for a master probe and slave probe.

**Note :** In the examples, make sure to use the full path for the property value. In other words replace \$OMNIHOME with the full path. For example: /opt/IBM/tivoli/netcool.

The following example shows the peer-to-peer settings from the properties file of a master probe:

```
Server      : "NCOMS"
RulesFile   : "master_rules_file"
MessageLog  : "master_log_file"
PeerHost    : "slave_hostname"
PeerPort    : 6789 # [communication port between master and slave probe]
Mode       : "master"
PidFile     : "master_pid_file"
```

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

```
Server      : "NCOMS"
RulesFile   : "slave_rules_file"
MessageLog  : "slave_log_file"
PeerHost    : "master_hostname"
PeerPort    : 6789 # [communication port between master and slave probe]
Mode       : "slave"
PidFile     : "slave_pid_file"
```

## 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_mtosi -datatype application/json -method post -data '{"command":"commandname","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**  
**resync**  
**name**

- *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 **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/nco_http -uri http://localhost:8080/probes/generic_mtosi -datatype application/JSON -method POST -data '{"command":"help","params": []}'
```

### resync

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

The format of the `-data` option for the **resync** command is: `-data '{"command":"version","params":[]}'`

The following command returns version information:

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

## name

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

The format of the `-data` option of the **name** command is `-data '{"command":"version","params":[]}'`

The following command returns version information:

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

## 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 : '{"command":"help","params":[]}'  
DataType : 'application/JSON'  
Method : 'POST'
```

To use this set of values use the following **nco\_http** command:

```
$OMNIHOME/bin/nco_http -uri http://test1.example.com:6789
```

## 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 6. Properties and command line options

Property name	Command line option	Description
<b>EncodingStandard</b> <i>string</i>	-encodingstandard <i>integer</i>	Use this property to specify the Encoding Standard Default ISO-8859-1 to include the Latin alphabet 1. The default is ISO-8859-1.
<b>HttpConnectionRead Timeout</b> <i>integer</i>	-httpconnectionreadtimeout <i>integer</i>	Use this property to specify the time (in seconds) following which the HTTP connection will timeout on read operation. The default is 300.
<b>HttpConnectionTimeout</b> <i>integer</i>	-httpconnectionconnect timeout <i>integer</i>	Use this property to specify the time (in seconds) following which the HTTP connection will timeout on connect operation. The default is 20.
<b>HttpLoginPassword</b> <i>string</i>	-httploginpassword <i>string</i>	Use this property to specify the password used to authenticate for HTTP operations. The default is password.
<b>HttpLoginUsername</b> <i>string</i>	-httploginusername <i>string</i>	Use this property to specify the username used to authenticate for HTTP operations. The default is username.
<b>HttpServiceAlarm RetrievalFilterFile</b> <i>string</i>	-httpalarmretrieval filterfile <i>string</i>	Use this property to specify the file containing the resync filter for alarm retrieval. The default is "".
<b>HttpServiceAlarm RetrievalURL</b> <i>string</i>	-httpservicealarm retrievalurl <i>string</i>	Use this property to specify the HTTP URL pointing to alarm retrieval service on target system The default is http://127.0.0.1:8081/nmsnbi/mtosi/AlarmRetrieval.
<b>HttpServiceNotification ConsumerIP</b> <i>string</i>	-http servicenotification consumerip <i>string</i>	Use this property to specify the local ip address for the NotificationConsumer service end point. The default is 127.0.0.1.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
<b>HTTPServiceNotificationConsumerPath</b> <i>string</i>	-http servicenotification consumerpath <i>string</i>	Use this property to specify the local path for NotificationConsumer service end point.  The default is '/nmsnbi/mtosi/NotificationConsumerService'
<b>HttpServiceNotificationPort</b> <i>string</i>	-http servicenotification consumerport <i>integer</i>	Use this property to specify the local port for the NotificationConsumer service end point.  The default is 2000.
<b>HttpServiceNotificationProducerURL</b> <i>string</i>	-http servicenotification producerurl <i>string</i>	Use this property to specify the HTTP URL pointing to the notification producer service on the target system. (This is used for subscribing/unsubscribing to alarm notification scenario).  The default is http://127.0.0.1:8081/nmsnbi/mtosi/NotificationProducer.
<b>KeyStore</b> <i>string</i>	-keystore <i>string</i>	Use this property to specify the keystore's location and filename.  The default is "".
<b>KeyStorePassword</b> <i>integer</i>	-keystorepassword <i>integer</i>	Use this property to specify the time (in seconds) that the probe waits before resynchronizing with the MTOSI server.  The default is password.
<b>ResyncBatchSize</b> <i>integer</i>	-resyncbatchsize <i>integer</i>	Use this property to specify incoming events batch size during re-sync.  The default is 100.
<b>SecurityProtocol</b> <i>string</i>	-securityprotocol <i>string</i>	Use this property to specify which security protocol the probe uses when SSL encryption is enabled.  If no value is set for the <b>SecurityProtocol</b> property, the probe uses the default protocol for the JRE.  The default is TLSv1.2.

Table 6. Properties and command line options (continued)

Property name	Command line option	Description
<b>ServerType</b> <i>string</i>	-servertype <i>string</i>	Use this property to specify the type of server used for listening for notification events. Valid value is HTTP only. The default is HTTP.
<b>TransformerFile</b> <i>string</i>	-transformerfile <i>string</i>	Use this property to specify the file that contains the transformer properties. The default is: \$(OMNIHOME)/java/conf/transformers.xml (UNIX). %OMNIHOME%\java\conf\transformers.xml (Windows).
<b>TransportFile</b> <i>string</i>	-transportfile <i>string</i>	Use this property to specify the file that contains the transport properties. The default is \$(OMNIHOME)/java/conf/jmsTransport.properties (UNIX) %OMNIHOME%\java\conf\jmsTransport.properties (Windows).
<b>TransportType</b> <i>string</i>	-transporttype <i>string</i>	Use this property to specify the transport method being used on the secondary MTOSI server. The default is JMS.
<b>TrustStore</b> <i>string</i>	-truststore <i>string</i>	Use this property to specify the file path of the truststore file. The default is "".
<b>TrustStorePassword</b> <i>string</i>	-trustedcertsstorepassword <i>string</i>	Use this property to specify the password required to access the keystore file containing the trusted certificates. The default is password.

Table 7. Generic Probe Framework Properties

Property name	Command line option	Description
<b>DataBackupFile</b> <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 " ". <b>Note :</b> Specify the path relative to \$OMNIHOME/var.
<b>HeartbeatInterval</b> <i>integer</i>	-heartbeatinterval <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 60.
<b>Inactivity</b> <i>integer</i>	-inactivity <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).
<b>InitialResync</b> <i>string</i>	-initialresync <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: false: The probe does not request resynchronization on startup. true: The probe requests resynchronization on startup. For most probes, the default value for this property is false.
<b>MaxEventQueueSize</b> <i>integer</i>	-maxeventqueuesize <i>integer</i>	Use this property to specify the maximum number of events that can be queued between the non native process and the ObjectServer. The default is 10000. <b>Note :</b> You can increase this number to increase the event throughput when a large number of events is generated.

Table 7. Generic Probe Framework Properties (continued)

Property name	Command line option	Description
<b>ResyncInterval</b> <i>integer</i>	-resyncinterval <i>integer</i>	<p>Use this property to specify the time (in seconds) that the probe waits before resynchronizing with the MTOSI server.</p> <p>The default is 0.</p>
<b>RetryCount</b> <i>integer</i>	-retrycount <i>integer</i>	<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>
<b>RetryInterval</b> <i>integer</i>	-retryinterval <i>integer</i>	<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>
<b>RotateEndpoint</b> <i>string</i>	-rotateendpoint <i>string</i>	<p>Use this property to specify whether the probe attempts to connect to another endpoint if the connection to the first endpoint fails.</p> <p>This property takes the following values:</p> <p>false: The probe does not attempt to connect to another endpoint if the connection to the first endpoint fails.</p> <p>true: The probe attempts to connect to another endpoint if the connection to the first endpoint fails.</p> <p>The default is false.</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.

The following table describes the elements that the Probe for Generic MTOSI generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

<b>Element name</b>	<b>Element description</b>
\$acknowledgeIndication	This element indicates the acknowledgement status of the alarm. The possible values are: <ul style="list-style-type: none"><li>• ACKNOWLEDGED</li><li>• UNACKNOWLEDGED</li></ul>
\$additionalText	This element contains a brief description of the problem being reported by the alarm.
\$alias	This element contains the name of the object reporting the alarm as given in the EMS user interface.
\$isClearable	This element indicates whether the alarm can be cleared. The possible values are: <ul style="list-style-type: none"><li>• false</li><li>• true</li></ul>
\$layerRate	This element indicates the layer to which the alarm applies.
\$nativeProbableCause	This element indicates the probable cause as given in the EMS user interface.
\$notificationId	This element contains the unique identifier of the alarm. This is derived from the serial number of the alarm as used by the EMS.
\$objectType	This element indicates the type of object reporting the alarm.
\$osTime	This element indicates the time at which the alarm was reported by the EMS.
\$perceivedSeverity	This element indicates the perceived severity of the alarm.
\$probableCause	This element contains the probable cause of the alarm.
\$probableCauseQualifier	This element contains the qualifier used to classify the alarm type.

<i>Table 8. Elements (continued)</i>	
<b>Element name</b>	<b>Element description</b>
\$rdn	This element indicates the Relative Distinguished Name of the object reporting the event.
\$serviceAffecting	This element indicates whether the alarm has affected the service.
\$sourceTime	This elements indicates the time at which the error occurred in the network element.
\$type	This element indicates the type of the alarm.
\$X733_AdditionalInformation	This element indicates the X733 additional information of the event.
\$X733_EventType	This element indicates the X733 event type.
\$X733_SpecificProblems	This element indicates the X733 specific problem of the event.

## 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*.

<i>Table 9. ProbeWatch messages</i>		
<b>ProbeWatch message</b>	<b>Description</b>	<b>Triggers/causes</b>
START SYNCHRONIZATION	The probe is synchronizing the events.	The probe has started receiving alarms from the alarm list.
FINISH SYNCHRONIZATION	The probe is ending the synchronization process. .	The probe has finished receiving alarms from the alarm 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.







SC27-8773-00

