Netcool/OMNIbus Probe Extension Package 8.0

Reference Guide
December 14, 2017

SC27-5682-07

Note

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 15.

Edition notice

This edition (SC27-5682-07) applies to version 8.0 of IBM Tivoli Netcool/OMNIbus Probe Extension Package and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces (SC27-5682-06).

Contents

Document control page

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

The IBM Tivoli Netcool/OMNIbus Probe Extension Package documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/#!/SSSHTQ/omnibus/common/kc_welcome-444.html

Table 1. Document modification history

Document version	Publication date	Comments
SC27-5682-00	July 5, 2013	First IBM publication.
SC27-5682-01	January 24, 2014	"Extended rules for the Generic Log File Probe and the Generic Log File Java Probe" on page 5 added to the guide. "Extended rules for the Generic Socket Probe" on page 7 added to the guide.
SC27-5682-02	March 12, 2015	New probe rules files added to the Probe Extension Package to support Generic Event Correlation functionality for Netact 3GPP V6 (CORBA) and Netact (ASCII) "Extended rules for generic event correlation" on page 7 added to the guide.
SC27-5682-03	April 30, 2015	New probe rules files added for the integration of Netcool/OMNIbus with the Internet of things using Node-RED. The following topics added to describe the integration of Netcool/OMNIbus with the Internet of things using Node-RED: • "Extended rules for integrating with the Internet of Things using IBM Node-RED" on page 10 • "Integrating with the Internet of Things using Node-RED" on page 10 • "Mapping Netcool/OMNIbus fields in Node-RED" on page 12 • "Using HTTP/HTTPS with basic authentication or SSL protection" on page 12
SC27-5682-04	August 5, 2015	New probe rules files added to the Probe Extension Package to support Generic Event Correlation functionality for Alcatel-Lucent 5620 SAM v13. "Extended rules for generic event correlation" on page 7 updated. Generic Log File Probe extension files updated to add support for ADVA FSP-NM event sources Versions 8.5.1 and 9.1. See "Extended rules for the Generic Log File Probe and the Generic Log File Java Probe" on page 5.

Table 1. Document modification history (continued)		
Document version	**Publication date**	**Comments**
SC27-5682-05	July 28, 2016	Updated for version 6 of the Probe Extension Package. "Extended rules for event enrichment" on page 10 added. This provides probe enrichment functionality for Alcatel-Lucent 5620 SAM v13/Nokia 5620 SAM V13 and v14 R1.
SC27-5682-06	November 24, 2016	Updated for version 7 of the Probe Extension Package. "Extended rules for event enrichment" on page 10 updated to state the addition of the `alcatel_5529_oad_v6.enrichment.rules` with the Probe Extension Package.
SC27-5682-07	December 14, 2017	Updated for version 8 of the Probe Extension Package. `adva-FSPNM.glf.lookup` file updated for ADVA FSM-NM versions 9.5 and 9.6. `alcatel_5620_sam.enrichment.rules` and `alcatel_5620_sam.itnm39.enrichment.rules` event enrichment files updated. Node-RED `EventMap` module updated.

Chapter 1. Probe Extension Package

The Probe Extension Package is a separate package that provides additional files, such as sql files, xml files, lookup files, and other files that a probe will use when processing data. Each time an updated version of the Probe Extension Package is released, you can use it to update the rules file that your existing probes use when processing data.

This allows you to update the probe to use the latest rules file without having to upgrade the probe currently running on your target device. To update your probe with these additional rules, see "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3.

For details about how to download the most recent version of the Probe Extension Package, see the following Release Notice on the IBM Software Support website:

http://www-01.ibm.com/support/docview.wss?uid=swg21652961

This guide contains the following sections:

What is the Probe Extension Package?

The Probe Extension Package is a collection of rules files written to a common standard, and provides enhanced event correlation and causal analysis for the IBM Tivoli Netcool suite. The IBM Tivoli Probe Extension Package complements the current out-of-the-box event correlation capabilities of IBM Tivoli Netcool/OMNIbus.

How does the Probe Extension Package operate?

Netcool/OMNIbus is a Service Level Management (SLM) system that presents a consistent and consolidated view of the current state of all the Netcool/OMNIbus managed systems to specific users. The Probe Extension Package improves the capability of Netcool/OMNIbus in providing more valuable information.

Probes, rules files, and the Probe Extension Package

The probes used by Netcool/OMNIbus collect and interpret information from disparate managed objects in a network. A probe parses the collected information, and sends the parsed data to the ObjectServer in a format described by the rules file and is compatible with the ObjectServer fields.

The default rules file necessary for the execution of a probe only performs generic grouping of data. Using a rules file enhanced to accommodate events from a specific device provides sharpened event enrichment and causal analysis. The Probe Extension Package is a collection of such rules files, fine tuned to specific managed objects that send events, for example, the enhanced Tivoli EIF rules file.

When the device sends the events, the probe uses the device specific rules file in the Probe Extension Package specified by the **`RulesFile`** property.

Note : If you do not specify the device specific rules file of the Probe Extension Package, the probe will use its default rules file.

ObjectServer and the Probe Extension Package

The IBM Tivoli Netcool/OMNIbus ObjectServer currently uses two types of automation to help reduce the number of events that require operator intervention. Generic Clear automations are designed to correlate and delete any matching pair of problem and resolution alerts, whereas deduplication automation eliminates duplicate alerts while maintaining an 'occurrence' count.

The Probe Extension Package increases the ability of Tivoli Netcool/OMNIbus ObjectServer automations to correlate alarms and identify root causes by mapping alarms and events to the relevant ObjectServer fields.

Prerequisites for using the Probe Extension Package

This section describes the software requirements needed to use the IBM Netcool/OMNIbus Probe Extension Package.

The Probe Extension Package provides enhanced rules to process data from a specific target. As such, the Probe Extension Package requires a prior installation of a currently supported version of IBM Tivoli Netcool/OMNIbus and also an IBM Netcool/OMNIbus probe.

Other Probe Extension Package requirements

The following IBM Tivoli Netcool/OMNIbus specification requirements apply to this IBM Tivoli Probe Extension Package installation:

- Supported operating system platforms
- Java Runtime Environment (JRE) requirements
- User interface requirements
- Disk space requirements
- Browser requirements on Windows platforms

These specifications are documented in the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Licensing

For the IBM Tivoli Probe Extension Package, license keys are not required beyond those relevant to the associated IBM Tivoli Netcool products, which use the IBM Tivoli software licensing process.

Accompanying publications

To efficiently utilize the IBM Tivoli Probe Extension Package and realize the benefits delivered, you must be familiar with the underlying principles of IBM Tivoli Netcool/OMNIbus including the following:

- The IBM Tivoli Netcool/OMNIbus components:
  - The ObjectServer (including the database tables and columns)
  - Probes (including editing probe properties, and stopping and restarting probes)
  - Desktop tools (including event lists, filters, and views)
  - Administration tools (including the IBM Tivoli Netcool/OMNIbus Administrator, the SQL interactive interface, and process control)
- The IBM Tivoli Netcool/OMNIbus directory structure and necessary configuration files.
- Basic rules file syntax including the use of lookup tables (both inline as well as separate files).
- Permissions, conversions, and automations.

The documents listed below provide reference information on the related concepts of the IBM Tivoli Netcool/OMNIbus and IBM Tivoli Probe Extension Package.

IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide

This book provides instructions for installing and deploying IBM Tivoli Netcool/OMNIbus, and includes details of the supported platforms and requirements.

IBM Tivoli Netcool/OMNIbus Administration Guide

This book describes how to perform administrative tasks using the IBM Tivoli Netcool/OMNIbus Administrator GUI, command line tools, and process control.

IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide

This book provides general introductory and reference information on probes and gateways. Documentation on the specific probes discussed within this guide is available from the IBM Tivoli Netcool Knowledge Center.

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/common/kc_welcome-444.html

About the Probe Extension Package installation

The Probe Extension Package installation incorporates a series of automatic and manual processes to add enhanced functionality to your existing IBM Tivoli Netcool/OMNIbus installation.

Some target systems require new columns in the `alerts.status` table. Some target systems may also require additional triggers, or modifications to existing triggers.

You must configure the relevant probes to use these updated rules files. The Probe Extension Package provides the necessary SQL files to perform these changes. If you do not use the SQL files provided you must manually carry out the changes using the Administrator GUI. The updated rules files enhance the ability of the probe to identify and flag events according to their causal relevance.

Extracting the Probe Extension Package and configuring a probe to use the additional rules

To use the Probe Extension Package files, follow these steps:

1. Extract the contents of the Probe Extension Package to `$OMNIHOME/probes/ProbeExtensions` or a location of your choice.

   Note : The Probe Extension Package must be installed on the same operating system as the probe installation it is going to operate with.
2. Set the PROBE_EXT environment variable to the location where you extracted the Probe Extension Package files.
3. Configure the default probe rules file to include the rules provided within the Probe Extension Package.
   a. Navigate to the location `$OMNIHOME/probes/`*arch*, where *arch* is the name of the platform on which the probe was installed. For example,

      `/opt/IBM/tivoli/netcool/omnibus/probes/solaris2`

   b. Edit the probe rules file to contain the `include` statement, and the path to the Probe Extension Package rules file you want the probe to use.

      For example, edit the rules file for the Probe for Tivoli EIF (`tivoli_eif.rules`) to include the additional rules for IBM Systems Director Events, which is packaged in the Probe Extension Package in the following location:

      ```
      include "$PROBE_EXT/eif/IBM_ISD/IBM_Systems_Director_Events.rules"
      ```

Chapter 2. The Probe Extension Package rules files

This guide contains the following probe extension packages:

Extended rules for the Generic Log File Probe and the Generic Log File Java Probe

This section lists the files in the Probe Extension Package available for the Generic Log File Probe and the Generic Log File Java Probe.

Table 2. Generic Log File Probe and Generic Log File Java Probe extension files.	
ADVA FSP-NM event sources Version 8.1, 8.5.1, 9.1, 9.5, and 9.6	
`adva-FSPNM.glf.lookup` - **updated**	`adva-FSPNM.glf.rules` - **updated**
	`adva-FSPNM.glf_java.compat.rules` - **updated**

Configuring the Generic Log File Probe or Generic Log File Java Probe and ObjectServer for ADVA FSP-NM

Additional changes are required to the Generic Log File Probe properties file and the ObjectServer for these additional rules to be used by the Generic Log File Probe.

Use the following steps to configure the Generic Log File Probe to use these additional rules:

1. Ensure you have followed the instructions in "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3.
2. Stop the Generic Log File Probe or Generic Log File Java Probe.
3. Set the following property values in the Generic Log File Probe properties file:
   - `RulesFile : '$PROBE_EXT/glf/ADVA_FSPNM/adva-FSPNM.glf.rules'`
   - `LogFileName : '`*`path_to_logfile`*`/eventlog.csv'`
   - `LineSeparator : '\n'`
   - `ValueSeparator : '|'`
   - `IgnoreNullFields : 0`
   - `QuoteCharacter : 034`

   The default properties file is `$OMNIHOME/probes/`*`arch`*`/glf.props`. Where *arch* is the operating system on which you have installed the Generic Log File Probe.
4. To use the new Generic Log File Java Probe to parse the `ADVA FSP-NM CSV` NBI file, set the following probe properties in the `$OMNIHOME/probes/<arch>/glf_java.props` file:

```
RulesFile               : '$PROBE_EXT/glf/ADVA_FSPNM/adva-FSPNM.glf.rules'
CleanStart              : 'false'
```

```
LogFileName             : 'path_to_logfile/eventlog.csv'
ParserElementDelimiter  : '\\|'
ParserIgnoreEmptyFields : 'false'
RecoveryFile            :
                '/home/netcool/IBM/tivoli/netcool/omnibus/var/glf_java.reco'
```

5. Create a new class in the ObjectServer with the following values:
   - `Class Number = 40565`
   - `Class Description = ADVA FSP-NM`

   Resynchronize the ObjectServer classes in your Netcool/OMNIbus Event List.

   Note : For more information about creating a new class see, *Creating and editing classes* in the *Netcool/OMNIbus Administration Guide*.
6. Restart the Generic Log File Probe or Generic Log File Java Probe.

Extended rules for the Probe for Tivoli EIF

This section lists the files in the Probe Extension Package available for the Probe for Tivoli EIF.

Table 3. IBM System Director probe extension files.	
IBM System Director event sources	
`IBM_Systems_Director_Events.rules`	`isd_db_update.sql`

Table 4. IBM Tivoli Enterprise Console probe extension rules.	
IBM Tivoli Enterprise Console event sources	
`tec_db_update.sql`	`tivoli_eif_tec.rules`

Table 5. IBM Tivoli Productivity Center probe extension rules.	
IBM Tivoli Productivity Center event sources	
`tivoli_eif_tpc.rules`	

Table 6. IBM Tivoli Storage Manager probe extension rules.	
IBM Tivoli Storage Manager event sources	
`tivoli_eif_tsm.rules`	

Configuring the Probe for Tivoli EIF and ObjectServer

Additional changes are required to the Probe for Tivoli EIF properties file and the ObjectServer for these additional rules to be used by the probe.

Use the following steps to configure the Probe for Tivoli EIF to use these additional rules:

1. Ensure you have followed the instructions in "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3.
2. Create the required Netcool/OMNIbus table columns using the `isd_db_update.sql` file.
   - On UNIX operating systems, run the following command:

     ```
     $OMNIHOME/bin/nco_sql -server "objectserver_name" -user "username" -
     password "password" < $PROBE_EXT/eif/IBM_ISD/isd_db_update.sql
     ```
   - On Windows operating systems, run the following command:

```
%OMNIHOME%\bin\isql.exe -S objectserver_name -U username -P password -i
%PROBE_EXT%\eif\IBM_ISD\isd_db_update.sql
```

Where:

- OMNIHOME is the location of your IBM Tivoli Netcool/OMNIbus installation.
- objectserver_name is the name assigned to your ObjectServer.
- *username* and *password* are your ObjectServer login details.

3. Stop and restart the ObjectServer and probe using the stop and start instructions provided in the Probe for Tivoli EIF Reference Guide.

Extended rules for the Generic Socket Probe

This section lists the files in the Probe Extension Package available for the Generic Socket Probe.

Table 7. NSN Netact 7 (ASCII NBI) probe extension files.	
NSN Netact 7 (ASCII NBI) event sources	
netact.map.rules	netact.socket.lookup
netact.socket.rules	NetActConfigParser.jar

Configuring the Generic Socket Probe to use NetAct 7 ASCII NBI Events

Additional changes are required to the Generic Socket Probe properties file and the ObjectServer for these additional rules to be used by the probe.

To configure the Generic Socket Probe to use the additional rules provided by the Probe Extension Package, follow these steps:

1. Set the following properties in the Generic Socket Probe properties file:
   - Header : "#S#"
   - Footer : "#E#"
   - ParseAsLines : 1
2. Obtain the ASCII alarm format file used by the NetAct system. The default format file is stored in /opt/oss/NSN-fmascii/smx/fmascii-format/
3. Backup the netact.map.rules file.
4. Use the NetActConfigParser utility program to generate a mapping rules file for your alarm format.

   For example, java -jar NetActConfigParser.jar -input *filename*.xml

   Where *filename* is the name of the alarm format file obtained in step 2.
5. Start the probe.

Note : For more information on the ASCII alarm format file, refer to *NetAct ASCII Alarm Forwarding Northbound Interface* documentation. For users who are using the old alarm format file, you can modify the default netact.map.rules file so that the alarm is parsed correctly by the probe.

Extended rules for generic event correlation

This section lists the files in the Probe Extension Package available for Generic Event Correlation.

Table 8. Generic Event Correlation probe extension files
Alcatel 5620 SAM v13 rules files

Table 8. Generic Event Correlation probe extension files (continued)	
`alcatel_5620_sam.genericcorr.include.rules`	`alcatel_5620_sam.genericcorr.user.include .rules`
NetAct 3GPPv6.4 rules files	
`netact3gpp.genericcorr.include.rules`	`netact3gpp.genericcorr.user.include.rules`
NMS2000 rules file	
`netact.genericcorr.include.rules`	`netact.genericcorr.user.include.rules`
Common rules file included by all vendor specific generic rules files	
`genericcorr.common.include.rules`	

Generic event correlation

Generic event correlation allows you to create event containers that correspond to incidents in networks and systems so you can determine the priority of the problems to be worked. The feature creates a one-to-one relationship between incidents and event containers so that first line operators can create trouble tickets from the head container events without duplication.

Generic event correlation consists of the following stages:

1. **Pre-classification**: This consists of assigning to each alarm a generic alarm type and defining its scope. It is achieved by adding the rules files supplied with the Probe Extension Package to the probe's rules file.
2. **Containerization**: This consists of assigning alarms to containers headed by a synthetic alarm. The containers retain the alarm's diagnostic information and enable easy access to the events grouped within the container. It is achieved by Netcool/OMNIbus automations.
3. **Probable cause and impact analysis**: This consists of setting the probable cause and impact of each alarm. For each alarm, the probable cause is determined by considering the highest weighted cause and impact from all the alarm's children events. It is achieved by Netcool/OMNIbus automations.
4. **Presentation**: This consists of displaying the event correlation information for a given alarm using the Web GUI Event Tables relationships feature.

Generic event correlation is currently available for the following systems:

- Alcatel-Lucent 5620 SAM v13
- Nokia-Siemens NMS2000
- Nokia-Siemens NetAct

Requirements

The generic event correlation feature has the following requirements:

- Correlation trigger. This is supplied by the following SQL files released with Netcool/OMNIbus V8.1 FP2:
  - `$OMNIHOME/extensions/eventgrouping/ootb_event_grouping.sql`
  - `$OMNIHOME/extensions/eventgrouping/ootb_event_grouping_remove.sql`

  The SQL files create the tables, additional columns in the `alerts.status` table, and automation triggers required to perform this functionality.
- Generic Event Correlation rules files. These form a part of the Probe Extension Package.

- Web GUI V8.1 for the creation of a Web GUI relationship whereby the **Parent** column is linked to the **Identifier Key** column.

Configuring the generic event correlation feature

To configure a probe to use the generic event correlation feature, follow these steps:

1. Ensure you have followed the instructions in "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3.
2. Install the `ootb_event_grouping` trigger if it is not already installed on your Netcool/OMNIbus deployment.

   Note : The SQL file is provided with Netcool/OMNIbus V8.1 FP2.
3. Edit the probe's rules file to include at the end of it the new generic correlation rules file written for that probe. See the table in "Extended rules for generic event correlation" on page 7 for a list of the generic correlation rules files that are available.

   Note : You can use Probe Rules Syntax Checker (`nco_p_syntax`) to verify the rules file syntax.
4. In Web GUI, create a new relationship for generic event correlation using the following steps:
   a. On the **Relationships** tab, click the **New relationship** icon.
   b. Specify `GenericParentChild` as the **Display Name** for the new relationship.
   c. Select `OMNIBUS` in the **Data Source** field.
   d. Select `ParentIdentifier` in the **Column** field.
   e. Select `Identifier` in the **Key Column** field.
   f. Click **Create Relationship**.
5. Copy the default view setting to create a new one so that you do not alter the default setting of the system using the following steps:
   a. Select **Views** from the **Administration** tab.
   b. Select **Global Views**.
   c. Select **Default**.
   d. Click the **Copy View** icon.
   e. Select the users who should have access to this view (for example, `Admin user`).
   f. Click **OK**.
6. Assign this relationship to your new view using the following steps:
   a. Select the **Relationships** tab.
   b. Select `GenericParentChild` in the **Relationship** field.
   c. Click **Save and Close**
7. Restart the probe or use the probe reload rules file utility if it has been enabled.

Now when you access the Event Viewer, it will display the following event details:

- The correlation between each site parent and child grouped together under the `SiteNameParent` synthetic event.
- The `SiteNameParent` synthetic events are further grouped under the `ScopeIDParent` synthetic events.
- The `Severity` of each parent event is set to the highest severity of any of its child events.
- The `ScopeIdParent` event (parent of all site parent events) `Summary` displays the number of sites affected.

Extended rules for event enrichment

This section lists the files in the Probe Extension Package available for event enrichment.

Table 9. Event enrichment probe extensions files	
Alcatel-Lucent 5620 SAM v13.0 and 14.0 rules files	
`alcatel_5620_sam.enrichment.rules` - **updated**	`alcatel_5620_sam.itnm39.enrichment.rules` - **updated**
Alcatel-Lucent 5529 OSS Alarm Dispatcher (OAD) v6 rules files	
`alcatel_5529_oad_v6.enrichment.rules`	

Event enrichment

The event enrichment rules files are compatible with Probe for Alcatel-Lucent 5620 SAM V13 and is developed for the IBM Tivoli Network Manager 4.1.1 or newer.

Note : For ITNM 3.9, the *.itnm39.* rules files are used.

To configure the default probe rules files to use the event enrichment rules:

1. See "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3 for instructions on extending default probe rules files.
2. Edit the main rules files and include the enrichment rules at the end of the main probe rules files.
3. Customize this rules files to support more alarm IDs for enrichment.

Extended rules for integrating with the Internet of Things using IBM Node-RED

This section lists the files in the Probe Extension Package available for integrating with the Internet of Things using Node-RED.

Installation directories

Integration with Node-RED functionality is installed in the following Probe Extension directories:

`$PROBE_EXT/iot/nodered/nodes` for all nodes.

`$PROBE_EXT/iot/nodered/flow` for example Node-RED flows.

Table 10. Probe extension files for integrating with the Internet of Things using Node-Red	
Node-Red configuration files	
`nodered.rules`	`nodered.props`

Integrating with the Internet of Things using Node-RED

The Internet of Things is the network of physical objects embedded with electronics, software, sensors, and connectivity to enable the exchanging of data with the manufacturer, operator, and other connected devices. Each thing is uniquely identifiable through its embedded computing system and can interoperate within the existing Internet infrastructure.

Node-RED is a tool for wiring together hardware devices, APIs, and online services. It is built on Node.js, a JavaScript runtime platform for building fast, scalable network applications. Node.js uses an event-driven, non-blocking I/O model.

The Probe Extension Package uses Node-RED to integrate IBM Netcool/OMNIbus with the Internet of Things. It provides two Node-RED nodes that allow you to send events from any data source to any IBM probe that supports the event factory facility. For example, you can configure a Netcool/OMNIbus Socket Probe or a Netcool/OMNIbus Generic Log File Probe to run in passive mode listening to incoming HTTP requests from Node-RED.

Requirements

The integration with the Internet of Things has the following requirements:

- Netcool/OMNIbus V7.4 or later.
- Node.js installed and configured to use the node: `http://nodejs.org`
- Node-RED installed and configured to use the node: `http://nodered.org`
- A Netcool/OMNIbus probe that supports HTTP/HTTPS mode configured to listen to HTTP/HTTPS requests from the Node-RED nodes provided by the Probe Extension Package.

Configuring the probe

To configure the integration with the Internet of Things, follow these steps:

1. Ensure that you have followed the instructions in "Extracting the Probe Extension Package and configuring a probe to use the additional rules" on page 3.
2. Ensure that the prerequisite software is installed and running.
3. Copy the JavaScript and HTML files from the following directory:

   `$PROBE_EXT/iot/nodered/nodes/omnibus`

   to your Node-RED/node installation directory. For example:

   `$NODERED/node/omnibus`
4. Optional step: Copy the example flow (`push2omnibus.js`) file to the NodeRed directory (`$NODERED`).
5. Copy the `nodered.props` file and `nodered.rules` file from the `$PROBE_EXT/iot/nodered` directory to the following directory:

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

   The `nodered.props` file is intended as reference to help you to edit the probe properties file.
6. Set the **`RulesFile`** property of the properties file of the probe that you will be using to listen to requests from the Node-RED to the following path:

   `$OMNIHOME/probes/`*`arch`*`/nodered.rules`
7. Make the following additional changes to the probe properties file:
   a. Set the **`Server`** property to the name of the ObjectServer.
   b. Set the **`Nhttpd.ListeningPort`** property to an open port.
   c. Update any other properties as required.
8. Start the probe.

   The probe will run as a listener to the HTTP port number set in the properties file and is not listening to any target.
9. Start Node-RED.

   In the Node-RED GUI, you will see two new nodes in the left pane: **EventFieldMap** and **EventFactory**.
10. Configure the **EventFactory** node to point to the host name and HTTP port number that the probe is listening to.

    Note : This setting can be made on the **EventFactory** node itself by double clicking on the node.

Mapping Netcool/OMNIbus fields in Node-RED

You can see how the JSON formatted messages received from the Internet of Things are mapped to Necool/OMNIbus fields using the Node-RED GUI.

The **EventFieldMap** contains a list of Netcool OMNIbus fields and a text box that contains the value to which it will be set. The values are set as a key-value pair in the JavaScript object.

Within the GUI, the node is shown as a form with two columns:

- Column one contains the ObjectServer fields.
- Column two contains the values to be set for each ObjectServer field.

  For fields other than the `Severity` field, the value in Column two can be one of the following:

  – A fixed string.

  – A variable specified in mustache format:

  {{field1}}

  where *field1* is the variable to which the field will be set when the probe sends the event to the ObjectServer.

  – For the `Severity` field, map the field to an object within the `msg.payload` and apply conditions to set the final severity value. For details, see"Configuring the Severity field" on page 12.

  Note: If you use double quotes in column two, they must be properly escaped using the backslash character; otherwise it will break JSON format.

Adding mappings

You can add more, optional ObjectServer fields by using the **Add** button at the foot of the EventFieldMap node GUI.

Configuring the Severity field

If you want to assign an element or token and apply further conditions to the `Severity` field to reflect the final severity level to be set, you can do so by updating the value for the Severity field in column 2. This gives better flexibility and node re-usability in a flow without having to duplicate the Event Map node to support different severity events.

The following example allows you to add a severity of Catastrophic:

```
Severity = field2
if == "catastrophic" set Severity = 5 (Critical)
if == "critical" set Severity = 5 (Critical)
if == "major" set Severity = 4 (Major)
if == "minor" set Severity = 2 (Minor)
else , set Severity = 1 (Clear)
```

The following example allows you to apply conditions to the integer values associated with the Severity field:

```
Severity = field2
if > "5" set Severity = 5 (Critical)
if > "3" set Severity = 4 (Major)
if > "2"  set Severity = 2 (Minor)
else , set Severity = 1 (Clear)
```

Using HTTP/HTTPS with basic authentication or SSL protection

You can use basic authentication or SSL-protected communication.

Using HTTP/HTTPS with basic authentication

You can use basic authentication with HTTP and HTTPS. This is enabled by setting a `username:password` credential in the probe's **`Nhttpd.BasicAuth`** property and using this in the EventFactory node by checking the **Use basic authentication** checkbox, and inserting the credentials. For details about enabling HTTP/S basic authentication on the probe, see the following topic in the IBM Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/probegtwy/task/omn_prb_enableremoteauthentication.html:

Using SSL protection

To use SSL communication between the probe and the remote host on which Node-RED is running, the probe must be configured with SSL enabled. For details, see the following topic in the IBM Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/probegtwy/task/omn_prb_enableremotessl.html.

To configure ObjectServer SSL-protected network, see the following topic in the IBM Knowledge Center:

http://www-01.ibm.com/support/knowledgecenter/SSSHTQ_8.1.0/com.ibm.netcool_OMNIbus.doc_8.1.0/omnibus/wip/install/concept/omn_con_ssl_usingssl.html.

When the SSL certificate is installed, update the URL field In the EventFactory node. Configure the URL to use HTTPS and the port that is configured in **`NHttpd.SSLListeningPort`** property; for example: `https://localhost:443/probe/common`

Note : If you are using a self-signed certificate, Node-RED may reject the connection because it is considered unauthorized and Node-RED will show a `SELF_SIGNED_CERT_IN_CHAIN` error on the console. To work around this, set the `NODE_TLS_REJECT_UNAUTHORIZED` environment variable to `0`, and restart Node-RED.

Example for Unix:

```
$ export NODE_TLS_REJECT_UNAUTHORIZED=0
$ node red.js -v push2omnibus.json
```

For Windows, edit this environment variable in System Properties.

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 the Probe Extension Package.

Table 11. Error messages

Error	Description	Action
`Field '`*`Field_Name`*`' not found`	A table or field is not found in the ObjectServer.	Create the table or table column on the ObjectServer.

Table 11. Error messages (continued)

Error	Description	Action
`Failed to open Rules File` `Failed to read rules: could not open rules file`	The problem could be one of the following issues: • The rules file does not exist. • The probe does not have permissions to read the rules file. • An incorrect file path was specified.	Ensure the rules file is readable and verify that the path is correctly defined in the rules file or in the environment variable.

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.

SC27-5682-07