IBM® Tivoli® Netcool/OMNIbus Probe for BMC Patrol V9
1.0

Reference Guide
November 8, 2013

SC27-6213-00

Notice

Before using this information and the product it supports, read the information in Appendix A, “Notices and Trademarks,” on page 23.

Edition notice

This edition (SC27-6213-00) applies to version 1.0 of IBM Tivoli Netcool/OMNIbus Probe for BMC Patrol V9 and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Document control page

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

The IBM Tivoli Netcool/OMNIbus Probe for BMC Patrol documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Information Center:

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

Table 1. Document modification history

Document version	Publication date	Comments
SC27-6213-00	November 8, 2013	First IBM publication.

Chapter 1. Probe for BMC Patrol V9

BMC Patrol is a management system based on agent technology. It comprises a console and multiple agents.

Each host to be monitored has a BMC Patrol agent on it, together with the required knowledge modules. A knowledge module is a plug-in device that connects to the agent and is used to monitor events. The agents send events to the BMC Patrol API and the console collects events from the API. Operators can specify on the console what to monitor and which thresholds to set. BMC Patrol allows for up to three threshold ranges: normal operation, threshold 1 exceeded, and threshold 2 exceeded.

The Probe for BMC Patrol V9 connects directly to the BMC Patrol agents and uses the Patrol API to acquire event data using the same method as the BMC Patrol console.

This guide contains the following sections:

Summary

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

The following table provides a summary of the Probe for BMC Patrol V9.

Table 2. Summary	
Probe target	BMC Patrol Agent version 9.0
Probe executable name	`nco_p_patrol_v9` (on UNIX and Linux operating systems) `nco_p_patrol_v9.exe` (on Windows operating systems)
Package version	1.0
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-patrol-v9-`*`version`*
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website:http://www-01.ibm.com/support/docview.wss?uid=swg21653012
Properties file	`$OMNIHOME/probes/`*`arch`*`/patrol_v9.props` (on UNIX and Linux operating systems) `%OMNIHOME%\probes\win32\patrol_v9.props` (on Windows operating systems)

Table 2. Summary (continued)	
Rules file	`$OMNIHOME/probes/arch/patrol_v9.rules` (on UNIX and Linux operating systems) `%OMNIHOME%\probes\win32\patrol_v9.rules` (on Windows operating systems)
Requirements	To run the probe on Windows operating systems, you must have a BMC Patrol Agent installed on the host on which you want to run the probe. See "Prerequisites for Windows operating systems" on page 2. For details of any additional software that this probe requires, refer to the `description.txt` file that is supplied in its download package.
Connection method	Patrol API
Remote connectivity	Available
Multicultural support	Not Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 **Note :** The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus 7.4.0.
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm. For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Prerequisites for Windows operating systems

Before installing the Probe for BMC Patrol V9, you must install a BMC Patrol Agent on the host on which you will be running the probe. This is required even if you have agents installed on other hosts.

Installing an agent on the same host as the probe provides the runtime libraries that the probe requires to run successfully.

You can obtain the BMC Patrol Agent installation files from BMC Software, Inc.

To install the BMC Patrol Agent, use the following steps:

1. Extract the installation files to a temporary directory.
2. Run the following command:

   temp_path`\bmc_products\setup.exe`

   Where *temp_path* is the path to the temporary directory.
3. On the **Select System Roles** panel, select **Managed System**.
4. On the **Select Products and Components to Install** panel, select **PATROL Agent for Microsoft Windows**.

5. On the **Provide the PATROL Default Account Properties** panel, specify your Windows login user name and password.
6. Click **Next** on all other panels to accept the default settings.
7. On the last panel, click **Finish**.

Verifying the %PATH% environment variable

When you install the BMC Patrol Agent, the path to the BMC Patrol Agent installation is set in the %PATH% environment variable.

The default location for the agent installation is `C:\Program Files (x86)\BMC Software \Patrol3\bin`.

To verify that the %PATH% environment variable has been set correctly, enter the following command at the command prompt:

```
echo %path%
```

Installing probes

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

The installation process consists of the following steps:

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

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

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html
2. Installing the probe using the installation package.

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

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

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

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.

BMC Patrol comprises a console and multiple agents. The Probe for BMC Patrol V9 connects directly to the BMC Patrol agents and uses the Patrol API to acquire event data using the same method as the BMC Patrol console. The Probe for BMC Patrol V9 can connect to one or more agents.

The console does not have to be running for events to be received by the probe from the agents.

Note : Where a firewall is used, the firewall must allow communication between the probe and the agents.

Data acquisition is described in the following topics:

Definition files

The Probe for BMC Patrol V9 reads the connection information for each agent from a definition (`.def`) file. The probe is supplied with a default definition file (`patrol_v9.def`) that you can modify for your system.

Definition files contain rows of agent connection information in the following format:

```
hostname username password port
```

Where:

- *hostname* is the name of the host on which the agent runs
- *username* is a username to log in to the host
- *password* is the encrypted password for the *username*
- *port* is the port to which you are connecting

You must use the `nco_patrol_v9_write` utility to add agents to the definitions file. The utility prompts you for the agent details. See "Utilities supplied with the Probe for BMC Patrol V9" on page 12.

Note : If an agent specified in the `.def` file is not available, the probe drops it from the list until the next session.

Running the probe

When the probe is run, it opens a connection to each of the agents specified in the definition file and then reads all of the events generated by each agent. If no valid connection details are specified in the definition file, the probe stops. If a connection is lost, the probe attempts to reconnect to the agent (if the **AgentRetries** property or **-interval** command line option is used). For information about properties and command line options, see "Properties and command line options" on page 6.

Every connection event (agent connection opened/lost/retried/closed, etc.) causes the probe to send both a ProbeWatch event and a normal event with the following tokens:

- `$port`
- `$description`
- `$node`
- `$severity`

.

Running the probe as a Windows service

To run the probe as a Windows service, use the following steps:

1. To run the probe on the same host as the ObjectServer, use the following command to register it as a service:

   ```
   %OMNIHOME%\probes\win32\nco_p_patrol_v9 /INSTALL -depend NCOObjectServer
   ```

2. To run the probe on a different host to the ObjectServer, use the following command to register it as a service:

   ```
   %OMNIHOME%\probes\win32\nco_p_patrol_v9 -install
   ```

3. Start the probe using the Microsoft Services Management Console.

Error recovery

The probe recognizes when a BMC Patrol agent loses its connection to the probe and it attempts to reconnect to that agent. The Probe for BMC Patrol V9 caches in memory the `EventID` of the last event read from each agent. If the connection between the agent and the probe is lost, the probe is able to request all new events from the last known `EventID`.

Additionally, the probe writes the last `EventID` from each agent to a recovery file. The probe can read this recovery file on startup and request any events that may have occurred since the probe last ran. For information about the recovery properties, see "Properties and command line options" on page 6.

Probe performance

The performance of the Probe for BMC Patrol V9 depends on many factors (for example, the network latency and the number, location, and configuration of agents). The optimum number of agents configured for each Probe for BMC Patrol V9 depends on these factors.

When the probe starts, it connects to every agent defined in the `.def` file. Consequently, the time taken for the probe to start depends on the number of agents in the file. If the time taken to start seems excessive, create two `.def` files and share the agents between the two. Then run two probes, each using different `.def` files.

Note : IBM Software Support recommends having no more than 100 agents in a `.def` file until implications on probe performance are fully understood for the particular installation.

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"
```

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 information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 3. Properties and command line options

Property name	Command line arguments	Description
AgentHeartbeat *integer*	`-heartbeat` *integer*	Use this property to specify the time (in milliseconds) that the probe waits to send a connection heartbeat. The default is `5000`.
AgentReport *integer*	`-agentreport` (This is equivalent to **AgentReport** with a value of 1.)	Use this property to specify whether or not the reporting of agent connections is enabled. This property takes the following values: `0`: The reporting of agent connections is disabled. `1`: The reporting of agent connections is enabled. The default is `0`. When reporting is enabled, the probe counts the number of agent connections that are dead, pending, open, or closed and sends a report as a ProbeWatch message. The probe also pings each agent that it is connected to. The BMC Patrol agents send one of the following responses: • `OK` • `WARN` • `ALARM` • `OFFLINE` • `VOID` • `timeout` If `timeout` is returned, the agent did not respond within the period specified in the **AgentStatusTimeout** property and the probe assumes that the connection is dead and needs to be reestablished.

Table 3. Properties and command line options (continued)

Property name	Command line arguments	Description
AgentReportCheck *integer*	`-agentreportcheck` *integer*	When AgentReport is enabled, this property specifies the number of milliseconds between checking the number of agent connections that are dead, pending, open, or closed. The default is `180000`.
AgentRetries *integer*	`-retries` *integer*	Use this property to specify the number of times the probe attempts to retry the connection. This property takes the following values: `0` (There is no limit on the number of attempts that the probe makes.) 2 through `10`. The default is 4. **Note :** You cannot set this property to 1.
AgentStatusTimeout *integer*	`-pingtimeout` *integer*	Use this property to specify the maximum time (in milliseconds) that the probe waits for an agent to respond with a connection status message. This should be quite small as the probe must wait this amount of time to determine whether a connection is dead. The default is `2000`.
AgentTimeout *integer*	`-timeout` *integer*	Use this property to specify the communication connection timeout in milliseconds. The default is `5000`.
ConnectedOnLogin *integer*	`-connectedonlogin` (This is equivalent to **ConnectedOnLogin** with a value of 1.)	Use this property to specify whether or not the probe sends the `Agent Connected` message when the probe receives the `User Logged In` message from that agent. The default is `0`.
ConnectionCheck *integer*	`-connectioncheck` *integer*	Use this property to specify the interval (in milliseconds) between successive checks of the probe's agent connections. When the connection has been retried the number of times specified by the **RetryInterval** property, the probe issues a new connection request. The default is `60000`.

Table 3. Properties and command line options (continued)

Property name	Command line arguments	Description
DefFileInterval *integer*	`-deffileinterval` *integer*	Use this property to specify the time (in milliseconds) that the probe waits between checking the BMC Patrol definitions file for changes. If the probe detects that the file has been altered, it reloads this file and then opens or closes agent connections, as necessary. The default is `300000`.
DisablePatrolPing *integer*	`-disablepatrolping` *integer*	Use this property to specify whether or not the probe pings the hosts. This property takes the following values: `0`: The probe pings the hosts. `1`: The probe does not ping the hosts. The default is `0`.
ExtraDebug *integer*	`-extradebug` (This is equivalent to **ExtraDebug** with a value of 1.)	When this property is set to 1, the probe logs extra information to the message log file. This extra debugging information shows the status of each agent connection and the number of cycles before the probe attempts to reconnect. The default is `0`. **Note :** This applies only when the message level is set to debug mode.
IgnorePasswordFailures *integer*	`-ignorefailures` (This is equivalent to **IgnorePasswordFailures** with a value of 1.)	When this property is set to 1, the probe ignores the error messages returned when the connection to an agent has failed due to username/password problems. This causes the probe to retry the connection to the agent (instead of ignoring that agent). The probe picks up any username/password changes each time it checks the `patrol_v9.def` file (as defined by the **DefFileInterval**). **Note :** If this property is set to 1 and authentication fails, the probe sends a message that specifies when the probe will retry the connection. The default is `0`.

Table 3. Properties and command line options (continued)		
Property name	**Command line arguments**	**Description**
InactivityTimeout *integer*	`-inactivitytimeout` *integer*	Use this property to specify the time (in seconds) that the probe allows an agent to be inactive before disconnecting from the agent and reopening the connection. This prompts the agent to send new alarms to the probe. The default is `3600`. **Note :** The minimum value for this property is `300` seconds.
LocalPort *integer*	`-localport` *integer*	Use this property to specify the local port number to communicate with a BMC Patrol agent across a firewall. The default is `0`.
MaxPingTimeouts *integer*	`-maxpingtimeouts` *integer*	Use this property to specify the number of consecutive timed-out ping responses that the probe allows before it attempts to reconnect to the agent. The default is `0`.
PatrolDef *string*	`-def` *string*	Use this property to specify the name of the file that contains the definitions of all agents to which the probe connects. The default is `$OMNIHOME/probes/`*`arch`*`/patrol_v9.def`.
RecoveryFileName *string*	`-recoveryfile` *string*	Use this property to specify the file to which recovery data is written. The recovery data is stored in an ASCII format as a sequence of *`ipnumber-EventID:`* `pairs`. The default is `$OMNIHOME/probes/`*`arch`*`/patrol_v9.reco`.
RecoveryInterval *integer*	`-recoveryinterval` *integer*	Use this property to specify the interval (in milliseconds) between successive recovery file writes. Depending on the number of agents that the probe is connected to, raise or lower this value so that writing the recovery data does not affect the performance of the probe. The default is `60000`.

Table 3. Properties and command line options (continued)

Property name	Command line arguments	Description
RecoveryMaxEvents *integer*	`-recoverymax` *integer*	Use this property to specify the maximum number of events sent by an agent in response to a recovery query. When the probe is recovering from a previous session, it requests that no more than the maximum number of events are sent to it per agent. The default is `1000`.
RecoveryMode *integer*	`-recoverymode` *integer*	Use this property to specify how the probe recovers if the connection between the agent and the probe is lost. This property takes the following values: `0`: NORMAL recovery mode. The probe does not read the recovery file on startup, but immediately begins listening for new events from the agents. `1`: RECOVERY mode. Forces the probe to read the recovery file and send requests to agents for events that may have occurred since the last event was received. You can use the **RecoveryStatusMaskFilter** and **RecoveryTypeMaskFilter** properties in conjunction with this mode. `2`: RESYNCH mode. The probe ignores the recovery file on start up, but requests events from all agents that match the settings indicated in the **RecoveryStatusMaskFilter** and **RecoveryTypeMaskFilter** properties, regardless of when they were received. The default filter properties select ALL events. Using these in RESYNCH mode requests ALL events from the agent, which may not be advisable. The default is `0`.

Table 3. Properties and command line options (continued)		
Property name	**Command line arguments**	**Description**
RecoveryStatusMask Filter *string*	`-recoverystatusmask` *string*	This property works in exactly the same way as **StatusMaskFilter**; however, it only applies to the query sent to agents when recovering from a previous session or when resynchronizing in RESYNCH mode. See the **StatusMaskFilter** property in this table for valid values and the default setting. When in RESYNCH mode, set the recovery filters to request only active alarms from the agents. See the **RecoveryMode** property in this table for more information. The default is `O,A,E,C,D`.
RecoveryTypeMaskFilter *string*	`-recoverytypemask` *string*	This property works in exactly the same way as **TypeMaskFilter**; however, it only applies to the query sent to agents when recovering from a previous session or when resynchronizing in RESYNCH mode. See the **TypeMaskFilter** property in this table for valid values and the default setting. When in RESYNCH mode, set the recovery filters to request only active alarms from the agents. See the **RecoveryMode** property in this table for more information. The default is `I,S,E,W,A,R`.
ResynchCheck *integer*	`-resynchcheck` *integer*	Use this property to specify the interval (in milliseconds) with which the probe checks whether a resynch request needs to be sent to one of the agents. It is only possible to send one request at a time, so this interval allows the probe to send resynch requests to the agents sequentially. The default is `30000`.
RetryInterval *integer*	`-interval` *integer*	Use this property to specify the number of `ConnectionCheck` intervals for which the probe waits before retrying a connection to an agent. The default is 3.

Table 3. Properties and command line options (continued)		
Property name	**Command line arguments**	**Description**
StatusMaskFilter *string*	`-statusmask` *string*	Use this property to specify the status mask filter that the probe uses to select events. For this property you must specify a string containing one or more status tags separated by commas. This property takes the following values: `O`: open `A`: acknowledged `C`: closed `E`: escalated `D`: deleted The default is `O,A,C,E,D` (this selects events of any status).
TypeMaskFilter *string*	`-typemask` *string*	Use this property to specify the type mask filter that the probe uses to select events. For this property you must specify a string containing one or more event type tags separated by commas. This property takes the following values: `I`: information `S`: state change `E`: error `W`: warning `A`: alarm `R`: response The default is `I,S,E,W,A,R` (this selects events of any type).

Utilities supplied with the Probe for BMC Patrol V9

Utilities are provided for adding agents to a definition file or removing existing agents from a definition file. These are simpler to use than editing the definition file and also prevent syntax errors.

The Probe for BMC Patrol V9 is supplied with the following command line utilities:

- `nco_patrol_v9_encrypt`
- `nco_patrol_v9_connections`
- `nco_patrol_v9_ping`
- `nco_patrol_v9_remove`
- `nco_patrol_v9_write`

The following table describes the utilities specific to this probe. For information about generic utilities, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 4. Utilities

Utility name	Command line arguments	Description
`nco_patrol_v9_encrypt`	N/A	This utility encrypts a password into the BMC Patrol format. The utility prompts you for a password, then asks for confirmation. **Note :** The encrypted password is for use in the BMC Patrol definition file. However, the encryption only works for a single entry of a single target host in BMC Patrol definition file. For multiple entries of a single target host (that is, a single target host but with multiple BMC Patrol agents), use nco_patrol_v9_write. To create an encrypted password, run the following command: `$OMNIHOME/probes/`*`arch`*`/nco_patrol_v9_encrypt` The utility prompts you to enter and verify a password and then displays the password in encrypted format. This is the password that you use in the definition file.
`nco_patrol_v9_connections`	`filename` *timeout* (optional)	This utility sends a BMC Patrol ping to each agent listed in the BMC Patrol definition file. The optional timeout is the time (in milliseconds) that the probe waits for a response before moving to the next agent.
`nco_patrol_v9_ping`	`hostname port-number timeout` (optional)	This utility sends a BMC Patrol ping to an agent on the specified host and port number. The optional timeout is the time (in milliseconds) that the probe waits for a response.
`nco_patrol_v9_remove`	`filename hostname` *portnumber*	This utility removes the agent on the host and port from the specified BMC Patrol definition file.

Table 4. Utilities (continued)		
Utility name	**Command line arguments**	**Description**
`nco_patrol_v9_write`	`filename`	This utility adds an agent to the specified BMC Patrol definition file. When the command is executed, you are prompted for the agent details. **Note :** If the `nco_patrol_v9_write` utility fails to add an agent to the definition file, you can add it manually, encrypting the password using the `nco_patrol_v9_encrypt` utility. When uisng `nco_patrol_v9_encrypt`, encryption only works for a single entry of a single target host in BMC Patrol definition file. When an existing definition file is used, there will be the following option output on the prompt: "Do you want to append to this file [Y]:" If you enter any character other than y, the file will be overwritten.

Note : All connection messages for the utilities are written to `stdout`; the message level cannot be set.

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 BMC Patrol V9 generates. Not all the elements described are generated for each event; the elements that the probe generates depends upon the event type.

Table 5. Elements	
Element name	**Element description**
`$args`	This element displays a list of arguments from the event received from the BMC Patrol agent.
`$catalog`	This element displays the event catalog name.
`$CauseCode`	This element indicates what caused the event.
`$class`	This element displays the class name of the event.
`$description`	This element displays the description of the event.
`$diary`	This element contains the BMC Patrol event diary entry. This contains information added by users to annotate, clarify, or further describe the event.

Table 5. Elements (continued)

Element name	Element description
$eventId	This element displays the event identifier. Event IDs are assigned sequentially by the BMC Patrol Event Manager as events are generated.
$expectancy	This element displays the life expectancy of the event. Typical values are: STORED, DEL_IF_CLOSED, DEL_IF_INFO, and DO_NOT_STORE.
$extdesc	This element contains an extended description of the event.
$fqhostname	This element displays the fully qualified domain name (FQDN) of the agent host.
$handler	This element displays the User ID of the last person to modify the event's diary, or of the last person to perform an ACKNOWLEDGE, CLOSE, or DELETE action on the event.
$ipaddr	This element contains the IP address of the connected agent.
$node	This element displays the host name of the node that generated the event.
$origin	This element displays the BMC Patrol application name of the source of the event.
$owner	This element displays the User ID of the owner of the event. The user ID is usually a user account.
$port	This element displays the port number of the host where the agent is connected.
$ProbeHost	This element displays the name of the host on which the probe is running.
$severity	This element displays the event severity. This can be an integer in the range 1 to 5, where 5 is the highest severity.
$sourceId	This element contains a source identifier of the event.
$status	This element displays the status of the event: 1: open 2: acknowledged 3: closed 4: escalated 5: deleted

Table 5. Elements (continued)	
Element name	**Element description**
$type	This element displays the type of the event: 1: information 2: state change 3: error 4: warning 5: alarm 6: response

Note : If the probe fails to connect to an agent or if a connection with an agent is lost, the probe forwards an error event to the ObjectServer. This event comprises the $node, $description, and $security elements. For details about these messages, see "ProbeWatch messages" on page 19.

Error messages

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

The following table describes the error messages specific to this probe. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 6. Error messages		
Error	**Description**	**Action**
All agent connections are dead - aborting	The connections to agents have ceased (and store and forward is not enabled in the probe).	Check that the BMC Patrol system is available.
Failed to create/set: *token name*	The probe is unable to create an element.	Contact IBM Software Support.
Failed to install SIGUSR1 handler	The probe was unable to install a signal handler for either a QUIT, INT or TERM. The probe will try to continue, however any further signal handling will result in an error on exit.	Contact IBM Software Support.
Failed to parse host name on line *host name* Failed to parse password on line *password* Failed to parse port on line *line number* Failed to parse username on line *username*	The probe was unable to parse the data specified.	Contact IBM Software Support.

Table 6. Error messages (continued)

Error	Description	Action
`Failed to perform initial stat of PatrolDefFile!`	The probe was unable to perform the initial `stat` on the `patrol_v9.def` file.	Check that the file exists and that the permissions are correct.
`Failed to process arguments`	Internal error.	Contact IBM Software Support.
`FAILED TO RECOVER`	The recovery file could not be found or could not be accessed and so the tailing begins from the current line value.	Check that the **`RecoveryFileName`** property is set correctly and that the user running the probe has read and write permissions over the file.
`Failed to send alert`	The probe was unable to send an alert (usually an internal alert) to the ObjectServer.	Check that the ObjectServer is available.
`Failed to set FirstOccurrence` `Failed to set LastOccurrence`	Some application has affected the operation of the probe.	Rerun the probe.
`Failed to stat PatrolDefFile` `Failed to perform initial stat of PatrolDefFile`	The probe cannot obtain the status of the file specified.	Check that the file exists and that the permissions are set correctly.
`Failed to stat PatrolDefFile!`	The probe was unable to perform a `stat` on the `patrol_v9.def` file.	Check that the file exists and that the permissions are correct.

`gethostbyaddr(): failed to find host` *`host name`*
`gethostbyname(): failed to find agent host host name`
`Hostname resolution problems`
	The probe was unable to log on to the host.	Check that the properties are set correctly.

`gethostbyaddr(): failed to find host` *`hostname`*	The probe was unable to find the host specified in the `patrol_v9.def` file.	Check that the specified host is running and reachable.
`gethostbyname(): failed to find host HostName`	The probe was unable to find the host specified in the `patrol_v9.def` file.	Check that the specified host is running and reachable.
`Incomplete final line in` *`line number`*	A line from the alert file is not in the correct format.	Check that the BMC Patrol is running correctly.

Table 6. Error messages (continued)

Error	Description	Action
`Invalid port on line` `Patrol Agent host is unreachable`	There is a problem with your machine, the network, or the probe configuration.	Check that there are no problems with your network and that you configured the correct port number in the probe properties file.
`Memory problems` `Out of memory`	A memory allocation problem occurred.	Make more memory available.

`No agents configured in Patrol Def file`
`No patrol agents configured - aborting`
`Patrol Def file errors`
	The BMC Patrol definition file does not contain the agent details.	Add the agent details in the BMC Patrol definition file.

`No patrol agents configured - aborting`	The agents have not been configured or have been incorrectly configured.	Check that the definition file exists and that the agents to which the probe connects have been correctly configured.
`No patrol agents configured - aborting`	There are no BMC Patrol Agents defined in the `patrol_v9.def` file.	Add some BMC Patrol Agents to your `patrol_v9.def` file.
`Patrol API problems`	Internal errors.	Contact IBM Software Support.
`Probe has been given` *`Unknown`* `RecoveryMode` *`Recovery Mode Value`*`. Defaulting to NORMAL MODE` `Unknown RecoveryMode Recovery Mode Value. Defaulting to NORMAL MODE`	As the given value for the **RecoveryMode** property is incorrect, the probe reverts back to the `NORMAL` mode.	Correct the **RecoveryMode** value in the properties file.
`PropertyName` `is greater than the maximum` *`PropertyValue of Number`* `Milliseconds - using maximum`	The property specified in the properties file or at the command line is greater than the maximum permitted. The maximum permitted value is used.	Correct the entry in the properties file.
`PropertyName` `is less than the minimum` *`PropertyValue`* `of` *`Number`* `Milliseconds - using minimum`	The property specified in the properties file or at the command line is lower than the minimum permitted. The minimum permitted value is used.	Correct the entry in the properties file.

Table 6. Error messages (continued)		
Error	**Description**	**Action**
`Rules processing failed: errordescription`	There is an error in the rules file format or syntax.	Check the rules file and correct the problem.
`Session create failed – aborting`	Internal errors.	Contact IBM Software Support.
`The number of agents is greater than the recommended absolute maximum of maximum limit` `Too many agents configured in Patrol Def file`	The number of agents specified in the BMC Patrol definition file is more than the default limit.	Reduce the number of agents in the BMC Patrol definition file to the limit shown in the error message.
`Unable to ping Host: HostName Port: PortValue Responded With: errordescription`	The probe lost its connection to the server.	Check that the BMC Patrol is running correctly on the socket shown in the error message.
`Unable to read recovery file: filename` `Unable to write recovery file: filename`	The probe cannot open, read, or reset the recovery file.	Check that the file exists and that the permissions are set correctly.

ProbeWatch messages

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

The following table describes the raw ProbeWatch error messages that the probe generates. For information about generic ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*, (SC14-7530).

Table 7. ProbeWatch messages		
ProbeWatch message	**Description**	**Triggers/causes**
`Agent on host Hostname has exceeded the InactivityTimeout period of seconds seconds`	The agent on the specified host has exceeded the specified inactivity timeout period.	The connection between the agent on the specified host and the probe has been inactive.
`All agent connections are dead - abortintg`	This is an information only message.	There are no live connections.

Table 7. ProbeWatch messages (continued)

ProbeWatch message	Description	Triggers/causes
Closed connection to PatrolAgent on host *host*, port *port* due to username/password problems. This agent will NOT be contacted again.	The user name and password supplied were not valid for the host.	Check the user name and password details and ensure that they are valid for the host. Also, ensure that the permissions on the specified host permit the user to log in. **Note :** The probe only displays this message if there is a problem with the username and password specified and the **IgnorePasswordFailures** property is set to 0.
Closed connection to PatrolAgent on host *host*, port *port* due to username/password problems. Will retry in *number* timer ticks.	The user name and password supplied were not valid for the host.	Check the user name and password details and ensure that they are valid for the host. Also, ensure that the permissions on the specified host permit the user to log in. **Note :** The probe only displays this message if there is a problem with the username and password specified and the **IgnorePasswordFailures** property is set to 1.
Closed connection to PatrolAgent on host *Hostname*, port Portnumber. Failed to connect *Number* times. This agent will NOT be contacted again.	The probe could not contact the specified host.	Check that the specified host is running and can be reached.
Connection report: Number pending, Number open, Number closed, *Number* dead	This is an information message that gives a report on the number of connections.	No action required.
Connection request to Agent on host Hostname has exceeded the AgentTimeout period of TimeoutPeriod milliseconds!	The probe exceeded the set waiting period for connection to the agent.	The host is not running or is unreachable.
Connection to the PatrolAgent on host *host*, port *host* closed, no longer required.	The probe was connected to the specified host, but when it read the patrol_v9.def file, the agent was not found and the connection was closed.	Check the reference to the agent in the patrol_v9.def file.

Table 7. ProbeWatch messages (continued)

ProbeWatch message	Description	Triggers/causes
Connection to the PatrolAgent on Host *Hostname*, port *Portnumber* closed, no longer required.	This is an information message. The probe has disconnected from an agent.	This message can be used to correlate the connection state of all BMC Patrol agents monitored by one or more probes.
Connection to the PatrolAgent on host *Hostname*, port *Portnumber* failed. Will retry in *Number* timer ticks. (*Number* tries left).	There is a problem connecting to the specified host.	Check that the specified host is running and can be reached.
Failed to reconnect to PatrolAgent on host *host*, port *port* due to username/password problems. This agent will NOT be contacted again.	The user name and password supplied were not valid for the host.	Check the user name and password details and ensure that they are valid for the host. Also, ensure that the permissions on the specified host permit the user to log in. **Note :** The probe only displays this message if there is a problem with the username and password specified and the **IgnorePasswordFailures** property is set to 0.
Failed to reconnect to PatrolAgent on host *host*, port *port* due to username/password problems. Will retry in *number* timer ticks.	The user name and password supplied were not valid for the host.	Check the user name and password details and ensure that they are valid for the host. Also, ensure that the permissions on the specified host permit the user to log in. **Note :** The probe only displays this message if there is a problem with the username and password specified and the **IgnorePasswordFailures** property is set to 1.
Host: *Hostname* Port: *Portnumber* Responded With: *String*.	This message is sent when the probe tests a connection. The final string originates from BMC Patrol. It may be one of the following: OK, WARN, ALARM, OFFLINE, VOID, or *timeout*. Consult your BMC Patrol documentation for further information.	This message can be used to correlate the connection state of all BMC Patrol agents monitored by one or more probes.

Table 7. ProbeWatch messages (continued)		
ProbeWatch message	**Description**	**Triggers/causes**
`Unable to read recovery file. CANNOT RECOVER! Unable to write recovery file. This may cause data loss between two restarts of the probe.`	The probe could not read the recovery file.	The recovery file is corrupt or it has wrong permission settings or an incorrect value is set in the properties file.

BMC Patrol API messages

The following table describes the API messages specific to this probe. For information about generic API messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 8. Properties and command line options		
Property name	**Command line option**	**Description**
`BMC: Message`	The message comes directly from the BMC Patrol API.	Refer to the BMC Patrol documentation for information.
`Unknown message from PEM: Message`	The message comes directly from the BMC Patrol API, but was not classified `INFO`, `WARN`, or `ERROR`.	Refer to the BMC Patrol documentation for information.

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.

SC27-6213-00