

IBM Tivoli Netcool/OMNIBus V7.3

Enabling self monitoring and flood detection in Netcool/OMNIBus probes



© 2011 IBM Corporation

IBM Tivoli® Netcool® OMNIBus 7.3: Enabling self monitoring and flood detection in Netcool/OMNIBus probes.

In this training module, you learn about enabling self monitoring and flood detection in Tivoli Netcool/OMNIBus probes.

Assumptions

- You have a general knowledge of Netcool/OMNIBus

Assumptions.

Before you begin this module, you must have a general knowledge of Netcool/OMNIBus.

Objectives

- After you complete this module, you can enable self monitoring and flood detection in Netcool/OMNibus probes

Objectives.

After you complete this module, you can enable self monitoring and flood detections in Netcool/OMNibus probes.

Self monitoring overview

- You can configure probes to send a ProbeWatch Heartbeat for self monitoring
- Self monitoring provides probe statistical data

Self monitoring overview.

You can configure probes to send a ProbeWatch Heartbeat if events are not sent to the ObjectServer at a designated interval. This self-monitoring mechanism provides probe statistical data that you can use to assess the condition of the probe.

Self monitoring actions

- Send or discard heartbeat events
- Populate the ObjectServer master.probestats table
- Write statistical data to a probe log file
- Log the ObjectServer report data on overall connections and events in addition to probe statistical data
- Modify the rules file to generate an event when the threshold is exceeded
- Send probe statistical data from the ObjectServer to Tivoli Data Warehouse by using the ODBC gateway for Tivoli Common Reporting

Self monitoring actions.

Self monitoring actions include sending or discarding heartbeat events, populating the ObjectServer master.probestats table, and writing statistical data to a probe log file. The ObjectServer reports include statistical data about overall connection and events in addition to probe statistical data. Optionally, you can modify rules files to generate an event. For example, you might modify these files when the threshold is exceeded. Probe statistical data can be sent from the ObjectServer to Tivoli Data Warehouse by using the ODBC gateway for Tivoli Common Reporting.

Probe statistics available

- CPU time
- Rules file processing time
- Memory used
- Number of events received
- Number of events discarded from rules
- Number of events generated by self monitoring
- Data gathered at ProbewatchHeartbeatInterval

Probe statistics available.

The probes statistics that are available are listed on the slide. They include CPU time, rules file processing time, memory used, number of events received, number of events discarded from rules, number of events generated by self monitoring, and data gathered at ProbewatchHeartbeatInterval.

ObjectServer statistics available

- Client connection counts and type
- Event counts
- Number of event inserts
- Data gathered every five minutes

ObjectServer statistics available.

The ObjectServer statistics that are available are client connection counts and type, event counts, number of event inserts, and data that is gathered every five minutes.

Self monitoring configuration

- On the ObjectServer, run **probstats.sql** to add necessary tables and triggers for the report log
- On each probe, modify the **probewatch.include** file to specify whether these actions occur:
 - Heartbeat is sent
 - Probe statistics are sent to ObjectServer
 - Probe statistics are written to the probe log
 - Threshold events are generated
- On each probe, include the probewatch.include file in the current rules file
- On each probe, add the ProbewatchHeartbeatInterval property
- Add the rule logic for generating threshold events (optional)
- Ensure that data is sent to Tivoli Data Warehouse by way of the ODBC gateway and that sample Tivoli Common Reporting reports are provided (optional)

Self monitoring configuration.

When you configure the probes for self monitoring and flood detection, you perform several tasks. On the ObjectServer, you run **probstats.sql** to add the necessary tables and triggers for the report log.

On each probe, you modify the **probewatch.include** file to specify whether the heartbeat is sent and whether the probe statistics are sent to the ObjectServer. You also specify whether the probe statistics are written to the probe log and whether threshold events are generated on each probe.

You include the probewatch.include file in the current rules file and add the **ProbewatchHeartbeatInterval** property on each probe. Optionally, you can add rule logic for generating threshold events. If data is sent to Tivoli Data Warehouse by way of the ODBC gateway, the sample Tivoli Common Reporting reports are provided.

Probe rule example

```
if( match( @Manager, "ProbeWatch" ) )
{
    include "$SOMNIHOME/probes/solaris2/probewatch.include"
}
```

Probe rule example.

In this example, you replace the ProbeWatch section at the beginning of the rules file with an include statement for the **probewatch.include** file.

Probe report example

Probe Log File Info MessageLevel

```
--- ProbeStats report at Heartbeat ---  
Timestamp: 1294869063  
Probe: glf  
Host: hostname.ibm.com  
PID: 29976  
ProbeID: glf@hostname.ibm.com  
NumberEventsProcessed: 8785  
NumberEventsDiscarded: 0  
NumberEventsGenerated: 0  
DiscardPercentage: 0.000000  
TotalRulesTime: 0.785492 sec  
TotalCPUTime: 43.089477 sec  
ProbeUpTime: -1294869003 sec  
AverageEventsPerSecond: 8785  
AverageRulesFileTime: 89 usec  
--- ProbeStats report end ---
```

Probe report example.

In this example, the probe statistics are logged by using the MessageLevel property to specify the message logging level. In this case, the level is **Info**.

ObjectServer report example

\$OMNIHOME/log/<ObjectServer Name>_probstats.log1

- Probe statistics are logged
- These master statistics are logged:
 - master.stats report:
 - NumClients: 4
 - NumRealtime: 2
 - NumProbes: 1
 - NumGateways: 0
 - NumMonitors: 0
 - NumProxys: 0
 - EventCount: 9979
 - JournalCount: 0
 - DetailCount: 0
 - StatusInserts: 9413
 - StatusNewInserts: 9396
 - StatusDedups: 17
 - JournalInserts: 0
 - DetailsInserts: 0

ObjectServer report example.

In this example, the probe statistics and the master statistics are logged in the ObjectServer report.

Flood detection overview

- Detection of event floods in a probe rule is based on a flood rate threshold
- Flood detection can be for anomalous event rates (too low or too high) in the probe

Flood detection overview.

You can configure a probe rule, based on a flood rate threshold. You can also configure a rule for an anomalous event rate. When an event rate is detected, the ObjectServer receives an informational alert that describes the event rate, which might be higher or lower than an average rate.

Flood detection: Remedial actions

If a flood is detected, the following actions can occur:

- Alerts for low or high event rates can be sent to the ObjectServer
- Alerts below a specified severity can be discarded
- Alerts can be diverted to a different ObjectServer

Flood detection remedial actions.

If a flood is detected, alerts for a low or high event rate can be sent to the ObjectServer. Alerts that are below a specified severity can be discarded. Alerts can also be diverted to a different ObjectServer.

Configuring probes to detect event floods

- Edit the **flood.config.rules** file for each probe
 - Sample time windows and maximum size
 - Flood detection startup time, flood threshold, and normal threshold (events per second)
 - Lower and upper event rate threshold multiplier for anomalous events
 - Discard events with low severity
 - Divert events during flood
- Include the **flood.rules** file and **flood.config.rules** file in the current rules file for each probe

Configuring probes to detect event floods.

When you configure the probes to detect event floods, you edit the **flood.config.rules** file for each probe. These edits include the sample time windows and maximum size, the flood detection startup time, the flood threshold, and normal threshold in events per second. The edits also include a lower and upper event rate threshold multiplier for anomalous events, discard events with low severity, and divert events during flood. You must include the **flood.rules** file and the **flood.config.rules** file in the current rules file on each probe.

flood.config.rules and flood.rules example

```
include "$OMNIHOME/probes/solaris2/flood.config.rules"
if( match( @Manager, "ProbeWatch" ) )
{
    include "$OMNIHOME/probes/solaris2/probewatch.include"
}
else
{
    include "$OMNIHOME/probes/solaris2/flood.rules"
}
.....
```

flood.config.rules and flood.rules example.

At the beginning of the rule file, include the **flood.config.rules** file. In the section before normal processing of flood events, include the **flood.rules** file.

Flood detection example

- Normal probe log
Debug: D-UNK-000-000: Current flood mode = normal
- Flood detected probe log
Debug: D-UNK-000-000: Current flood mode = flood
Warning: W-UNK-000-000: Event flood detected
Debug: D-UNK-000-000: Executing genevent() command for target server 'DefaultOS'.
Debug: D-UNK-000-000: genevent() created new alert for target server 'DefaultOS'.
Debug: D-UNK-000-000: genevent() created new alert for target server 'DefaultOS'.
Debug: D-UNK-000-000: genevent() sent new event to target server 'DefaultOS'.
Debug: D-UNK-000-000: Discarding event during event flood
- Flood detected alerts
Critical "An event flood has been detected, event rate is <EPS>"
Major "An event flood has finished, event rate duration: <seconds>, event count during flood: <EPS>"

Flood detection example.

In this example, you see a normal probe log, a flood detected probe log, and flood detected alerts. In the flood detected probe log, you see that an event is sent to the ObjectServer for the detected flood. The event is discarded because of severity. An informational alert is sent to the ObjectServer when the event flood is detected and when it is resolved.

Anomalous event example

- Normal probe log
 - Debug: D-UNK-000-000: Average event rate for anomaly detection is set to 35.017486
 - Debug: D-UNK-000-000: Current event rate for anomaly detection is set to 33.685707
- Anomalous probe log
 - Debug: D-UNK-000-000: Average event rate for anomaly detection is set to 35.017486
 - Debug: D-UNK-000-000: Current event rate for anomaly detection is set to 0.001283
 - Debug: D-UNK-000-000: Executing genevent() command for target server 'DefaultOS'.
 - Debug: D-UNK-000-000: genevent() created new alert for target server 'DefaultOS'.
 - Debug: D-UNK-000-000: genevent() created new alert for target server 'DefaultOS'.
 - Debug: D-UNK-000-000: genevent() sent new event to target server 'DefaultOS'.
 - Debug: D-UNK-000-000: Current event rate is unusually low
- Anomalous event
 - Minor "An unusually low event rate has been detected, event rate is <EPS>"

Anomalous event example.

In the anomalous probe log, the current event rate is much lower than normal. An alert about the low event rate is sent to the ObjectServer.

Additional high availability options

- ActingPrimary ObjectServer property that is set by gateways to determine resynchronization direction and minimize event loss
- Minimal Gate.Resync.Type reduces resynchronization time
- Configuration of controlled shutdown of the ObjectServer to flush events before shutdown
- Automatic load balancing of desktop event list clients
- Performance triggers that calculate TimeToDisplay at the display layer
- Multitier architecture
- Controlled failback

Additional high availability options with Netcool/OMNibus 7.3.

Additional high availability options are available in Netcool/OMNibus 7.3. You can use the ActingPrimary ObjectServer property to determine resynchronization direction and minimize event loss. You can use the Gate.Resync.Type property to specify the type of resynchronization that is required and reduce synchronization time. You can configure a controlled shutdown of any ObjectServer to flush the events before shutdown. You can set up automatic load balancing of desktop event list clients. You can use performance triggers to calculate the TimeToDisplay value at the display layer. You can use multitiered architecture and controlled failback. See the *IBM Tivoli Netcool/OMNibus Administration Guide* for more information about these options.

Summary

- Now that you have completed this module, you can enable self monitoring and flood detection in Netcool/OMNIBus probes

Summary.

Now that you have completed this module, you can enable self monitoring and flood detection in Netcool/OMNIBus probes.

Training roadmap for IBM Tivoli Netcool/OMNibus

www.ibm.com/software/tivoli/education/edu_prd.html

Training roadmap for IBM Tivoli Netcool/OMNibus.

You can see the training roadmap for IBM Tivoli Netcool/OMNibus by going to the web site that is shown on the slide.

Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, Netcool, and Tivoli are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.