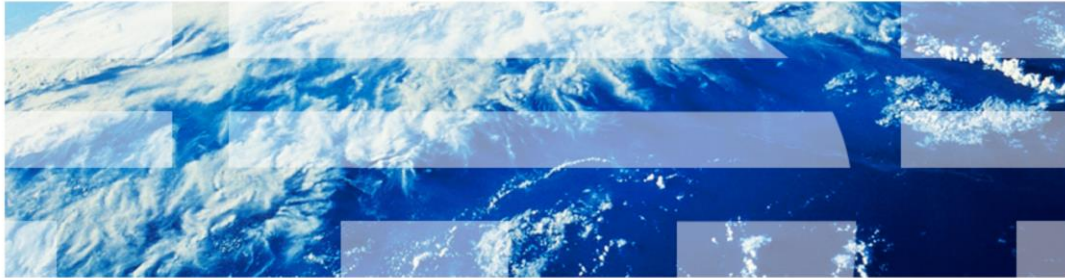


Tivoli Netcool OMNIbus V7.3.1

Introduction to the generic_clear trigger



Tivoli Netcool OMNIbus V7.3.1, Introduction to the generic_clear trigger

Objectives

After you complete this module, you can perform these tasks:

- Describe how to access the settings and source code of the generic_clear trigger
- Describe the algorithm of the generic_clear trigger
- Troubleshoot any generic_clear trigger related issues

After you complete this module, you can perform these tasks:

- Describe how to access the settings and source code of the generic_clear trigger.
- Describe the algorithm of the generic_clear trigger.
- Troubleshoot any generic_clear trigger related issues.

Agenda

- Introduction
- Settings and source code of the generic_clear trigger
- Algorithm of the generic_clear trigger
- An example
- Summary

This module covers a few important topics:

- The introduction section covers some basic information about OMNibus triggers and the generic_clear trigger.
- In the next section, you learn more about how to configure the settings and locate the source code of the generic_clear trigger.
- Finally, the algorithm of the generic_clear trigger is explained.

Introduction

- There are three types of triggers: database triggers, signal triggers, and temporal triggers.
- The generic_clear trigger is a type of temporal trigger. A temporal trigger fires repeatedly based on a specified frequency.
- The generic_clear trigger is used to clear an event when a resolution event is received.

There are three types of triggers: database triggers, signal triggers, and temporal triggers. The generic_clear trigger is a type of temporal trigger. A temporal trigger fires repeatedly based on a specified frequency. By default, the generic_clear trigger fires every five seconds. The generic_clear trigger is used to clear an event by setting the severity to 0 when a resolution event is received.

(1 of 3) Settings and source code of the generic_clear trigger

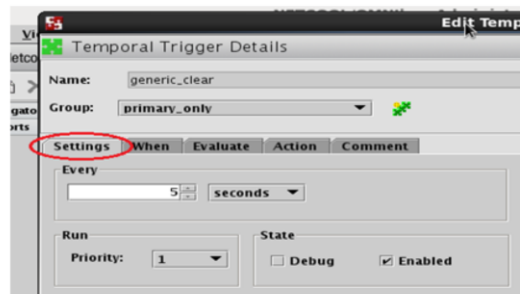
Use the `nco_config` command to access the settings and source code of the generic_clear trigger

Name	Group	Kind	Priority	Debug	Enabled	Group E...
clean_journal_table	default_trig...	Temporal	1	✗ false	✓ true	✓ true
connection_watch...	connection_...	Signal	1	✗ false	✓ true	✓ true
dedup_status_ins...	stats_triggers	Database	20	✗ false	✓ true	✗ false
deduplicate_details	default_trig...	Database	1	✗ false	✓ true	✓ true
deduplicate_iduc...	iduc_triggers	Database	1	✗ false	✓ true	✓ true
deduplication	default_trig...	Database	1	✗ false	✓ true	✓ true
delete_clears	primary_only	Temporal	1	✗ false	✓ true	✓ true
details_inserts	stats_triggers	Database	20	✗ false	✓ true	✗ false
disable_inactive...	security_wa...	Temporal	1	✗ false	✗ false	✓ true
disable_user	security_wa...	Signal	1	✗ false	✓ true	✓ true
disconnect_iduc...	iduc_triggers	Signal	1	✗ false	✓ true	✓ true
escalate_off	default_trig...	Temporal	1	✗ false	✗ false	✓ true
expire	primary_only	Temporal	1	✗ false	✓ true	✓ true
flash_not_ack	default_trig...	Temporal	1	✗ false	✗ false	✓ true
generic_clear	primary_only	Temporal	1	✗ false	✓ true	✓ true
iduc_messages_t...	iduc_triggers	Temporal	1	✗ false	✓ true	✓ true
iduc_stats_insert	iduc_triggers	Signal	1	✗ false	✓ true	✓ true
iduc_stats_update	iduc_triggers	Signal	1	✗ false	✓ true	✓ true
journal_inserts	stats_triggers	Database	20	✗ false	✓ true	✗ false

The easiest way to access the settings and source code of the generic_clear trigger is through the `nco_config` command. When you are connected to the Object Server, click **Automation**. Then, select **Triggers**. You should be able to see the generic_clear trigger in the list.

(2 of 3) Settings and source code of the generic_clear trigger

- To change the settings of the generic_clear trigger, right click the generic_clear trigger and select **Edit Trigger**.
- Under the **Settings** tab, there are configurations that can be changed:
 - (a) Frequency
 - (b) Priority
 - (c) State
- Because the generic_clear trigger is a temporal trigger, one of the more important settings is the frequency. The default is five seconds.



6

Introduction to the generic_clear trigger in Tivoli Netcool OMNibus

© 2012 IBM Corporation

To change the settings of the generic_clear trigger, right click the generic_clear trigger and select the **Edit Trigger** option.

Under the **Settings** tab, there are configurations that can be changed:

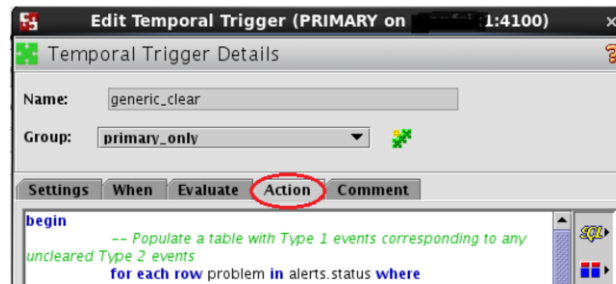
- (a) Frequency
- (b) Priority
- (c) State

...

Because the generic_clear trigger is a temporal trigger, it is important to make sure that the frequency setting meets the requirement of your environment. The default frequency is five seconds.

(3 of 3) Settings and source code of the generic_clear trigger

- The source code of the generic_clear trigger is accessible through the same window under the **Action** tab.
- It is advisable not to change the source code.
- If there is any specific requirement, you can duplicate the trigger and change it accordingly.



The source code of the generic_clear trigger is accessible through the same window under the **Action** tab. It is advisable not to change the source code. If there is any specific requirement, you can duplicate the trigger and change it accordingly.

(1 of 2) Algorithm of the generic_clear trigger

- The generic_clear trigger first goes through the alerts.status table to find all events that are of **Type 1 (Problem)** and **Severity > 0** (any other severity than Clear) and match it to all events in the same table that is of **Type 2 (Resolution)** and **Severity > 0**.
- The matching conditions are in the following list:
 - (a) Matched events have to share the same Node value AND
 - (b) Matched events have to share the same AlertKey value AND
 - (c) Matched events have to share the same AlertGroup value AND
 - (d) Matched events have to share the same Manager value
- All alerts that match the conditions stated here are selected into a temporary table named alerts.problem_events.

The generic_clear trigger first goes through the alerts.status table to find all events that are of **Type 1 (Problem)** and **Severity > 0** (any other severity than Clear) and match it to all events in the same table that is of **Type 2 (Resolution)** and **Severity > 0**.

The matching conditions are in the following list:

- (a) Matched events have to share the same Node value AND
- (b) Matched events have to share the same AlertKey value AND
- (c) Matched events have to share the same AlertGroup value AND
- (d) Matched events have to share the same Manager value

...

All alerts that match the conditions stated here are selected into a temporary table named alerts.problem_events.

(2 of 2) Algorithm of the generic_clear trigger

- Then, the trigger goes through the alerts.status table again and selects all alerts of type **uncleared (Severity > 0) Resolution (Type = 2)**
- For all matched alerts, the trigger completes the following tasks:
 - (a) Clear the alert (by setting Severity = 0)
 - (b) Update the corresponding alert of type Problem in the temporary table (alerts.problem_events) and set it as Resolved
- The matching is done based on these conditions:
 - (a) The (a) to (d) conditions described in the previous page
 - (b) The alert of type Resolution has to occur later than the corresponding alert of type Problem
- The last step of the trigger is to update all those alerts marked as Resolved in the temporary table (alerts.problem_events) to **Clear (Severity = 0)** in the alerts.status table
- Now, all Problem alerts with corresponding Resolution alerts are set to clear

9

Introduction to the generic_clear trigger in Tivoli Netcool OMNibus

© 2012 IBM Corporation

Then, the trigger goes through the alerts.status table again and selects all alerts of type **uncleared Resolution** that have **Severity > 0** and **Type = 2**. For all matched alerts, the trigger completes the following tasks:

- (a) Clear the alert by setting Severity to zero(0).
- (b) Update the corresponding alert of type Problem in the alerts.problem_events table and set it as Resolved.

The matching is done based on these conditions:

- (a) The (a) to (d) conditions described in the previous page.
- (b) The alert of type Resolution has to occur later than the corresponding alert of type Problem.

...

The last step of the trigger is to update all those alerts marked as Resolved in the alerts.problem_events to Clear in the alerts.status table. Now, all Problem alerts with corresponding Resolution alerts are set to clear.

An example

- This is a Problem event:

Severity	Alert Key	Node	Alert Group	Summary
Critical	Unknown alarm location	Unknown Host	Communication Alarm	Unknown alarm location OUT_OF_MEMORY Communication Alarm

- These are the cleared Problem and Resolution events:

Severity	Alert Key	Node	Alert Group	Summary
Clear	Unknown alarm location	Unknown Host	Communication Alarm	Unknown alarm location OUT_OF_MEMORY Communication Alarm
Clear	Unknown alarm location	Unknown Host	Communication Alarm	Unknown alarm location OUT_OF_MEMORY Communication Alarm

In this slide, you can see from the example how a Resolution event is used to clear a Problem event.

Summary

- The generic_clear trigger is used to correlate a Problem and a Resolution
- When a Resolution for a particular event is detected, the Problem event is then cleared
- The trigger is easy to improvise to cater for a more complex correlation methodology
- For more information, see page 319 of the *OMNibus Administration Guide 7.3.1*

Now that you have completed this module, you can perform these tasks:

- Describe how to access the settings and source code of the generic_clear trigger
- Describe the algorithm of the generic_clear trigger
- Troubleshoot any generic_clear trigger related issues

The generic_clear trigger is used to correlate a Problem and a Resolution. When a Resolution for a particular event is detected, the Problem event is then cleared. The trigger is easy to improvise to cater for a more complex correlation methodology. For more information, you can see page 319 of the *OMNibus Administration Guide 7.3.1*. Now that you have completed this module, you can perform these tasks:

- Describe how to access the settings and source code of the generic_clear trigger.
- Describe the algorithm of the generic_clear trigger.
- Troubleshoot any generic_clear trigger related issues.

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 2012. All rights reserved.