

Revision	Description
4/5/2010	Original

SQL-Hero Notifications

Introduction

Notifications in SQL-Hero take events of interest and turn them into messages that can be delivered in a number of ways, to “targeted individuals.” The reasons this is useful include:

- Keeps people abreast of changes, error conditions or other significant events; this is more proactive than say requesting they regularly consult some change log.
- Allows people to focus on their own work without being overwhelmed by a lot of extraneous information that might be produced if one were to say report test failures for objects that the developer had no hand in. SQL-Hero allows for delivery to only developers of interest.

The notification system in SQL-Hero changed significantly in version 0.9.6 to be more flexible, supporting new types of events in the future. As it is now, there are seven different kinds of events that SQL-Hero can send alerts on.

Use of notifications requires that you have SQL-Hero server components installed. Notification meta-data and instance data about the events of interest are stored in the SQL-Hero repository. Consult the whitepaper on “Installing SQL-Hero” for more details.

Creating and Maintaining Notifications

Notifications are managed on the Notification tool window. This window serves two purposes: 1. It allows you to see and/or manage notifications, 2. It shows captured events that are to be delivered using the SQL-Hero client (be it standalone or using Visual Studio integration).

Let’s start by creating a new notification. Click the “New” button on the toolbar:

Description	Databases	Events	Delivery Types	Frequency
Object Changes	Local-Test1, Local-Test2, Local-Test3	Object Changes	E-mail (HTML)	Immediate
Stored trace perf	Local-Test1, Local-Test2, Local-SQLHero, Local-Test3	Stored Trace Alert - Performance Failure	Notify in Application	Immediate
Unit perf failures - SQL Hero	Local-SQLHero	Unit Testing - Error, Unit Testing - Performance Failure	E-mail (HTML)	Hourly
Code gen		Code Gen Invocation	Notify in Application	Immediate
Region Diffs for RD Databases	RD-Development, RD-QA	Region Difference	E-mail (HTML)	Five Minutes

This brings up the notification detail dialog:

Notification Detail...

OK Cancel Delete

Notification Description: [Empty text box]

Frequency: Immediate

Events and Filters

- Event(s): Select at least 1 event type [Configure]
- Database(s): [Empty text box]
- User name regular expression: Optional user name filter (object modifier)
- Object regular expression: Optional object name filter
- Exclude objects changed by self
- Exclude system objects and DML
- New objects only
- Available to all (public)
- Report current object only

Distribution

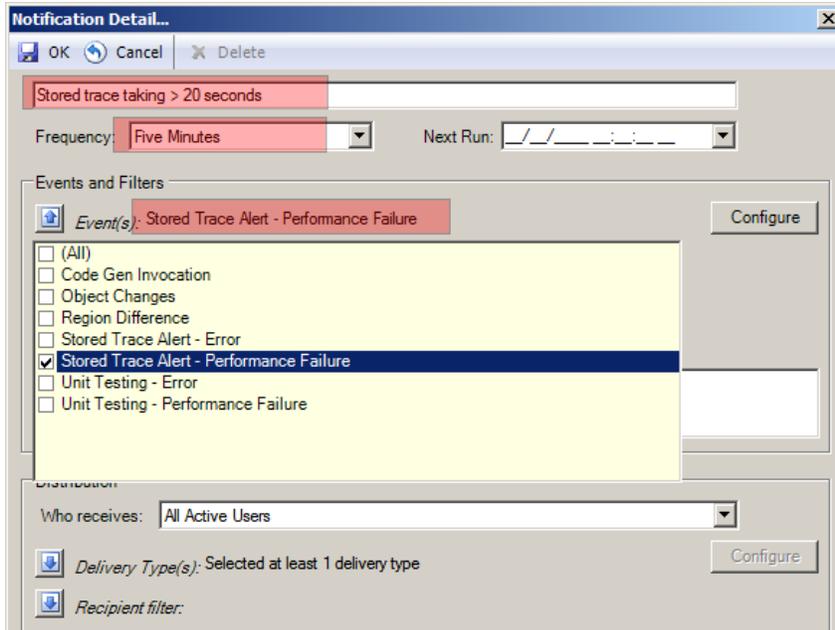
Who receives: All Active Users

- Delivery Type(s): Selected at least 1 delivery type [Configure]
- Recipient filter: [Empty text box]

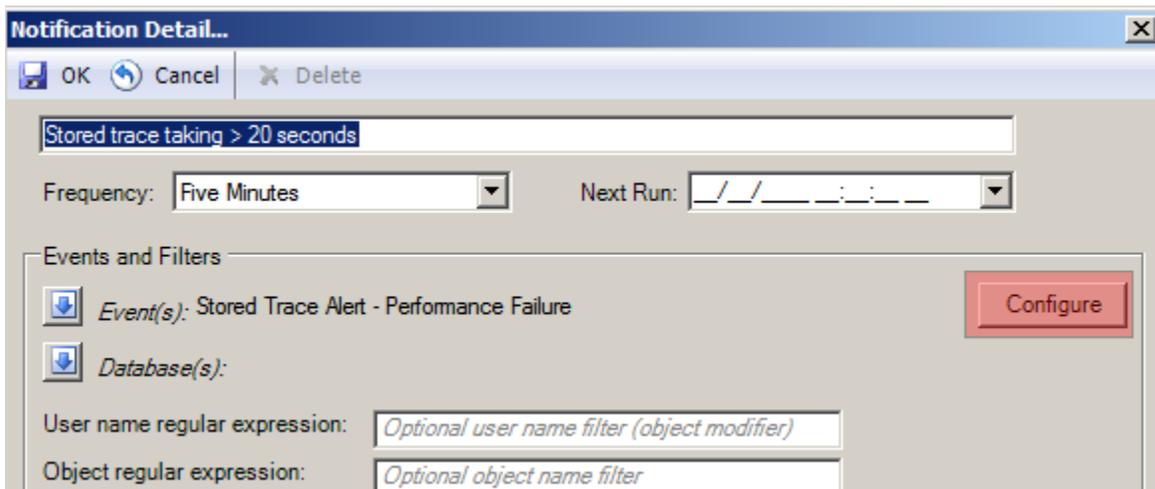
Last Run Info Owner: CODEX07\Administrator

Let’s say we want to be notified about cases where a trace has been collected and uploaded to the SQL-Hero repository for the database RD-QA, where the duration is more than 20 seconds. (For details on what trace uploading is about, see the whitepaper “SQL-Hero Tracing.”) Furthermore let’s say we want to be notified within 5 minutes of the event being captured, and let’s say we’re only interested in mentioning objects to developers that they themselves have changed at some point in time.

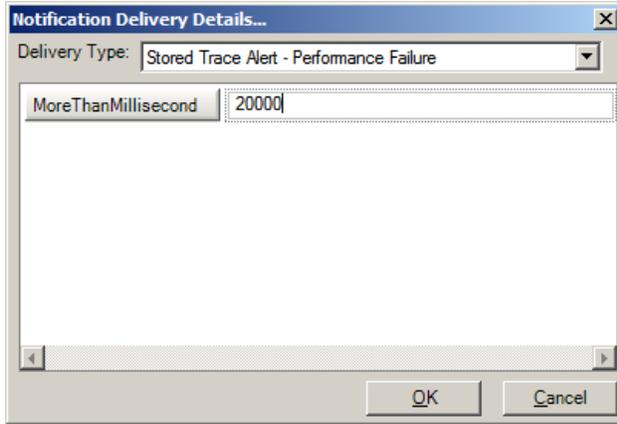
Start by filling out the fields highlighted below:



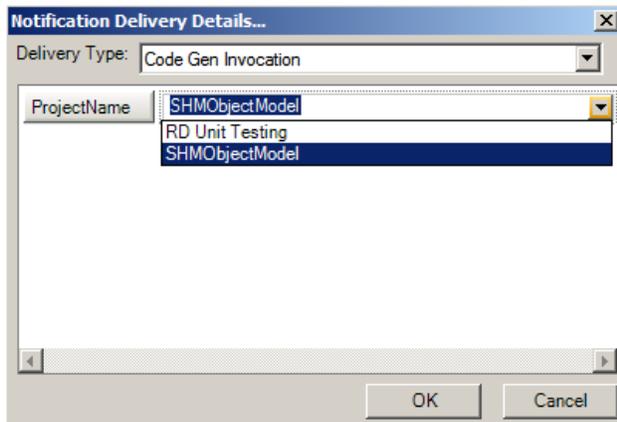
We've picked the event type of "Stored Trace Alert – Performance Failure," but this event type requires some configuration – click the configuration button for it next:



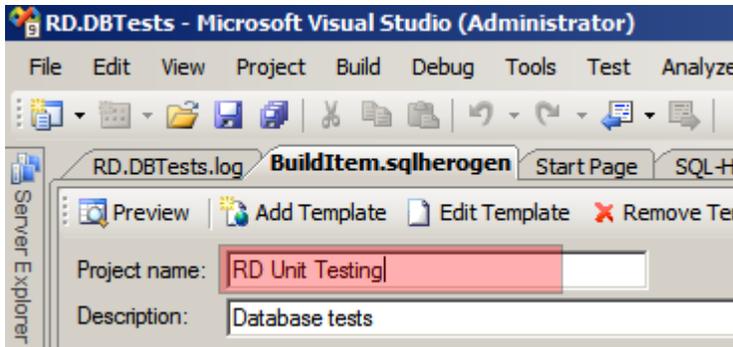
This brings up an event-specific dialog that includes any necessary parameters. Our requirement was to be alerted for collected trace records with duration of more than 20 seconds:



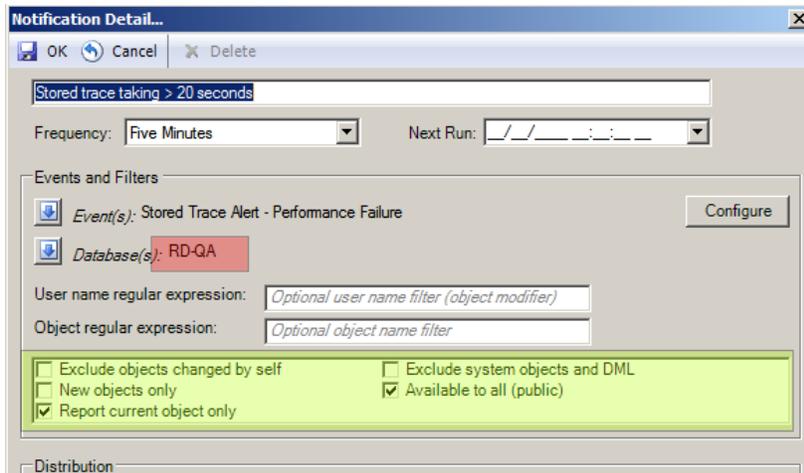
Note that in the case of the “Code Gen Invocation” event type, the project name here:



Should correspond to the project name as seen in the sqlherogen designer:



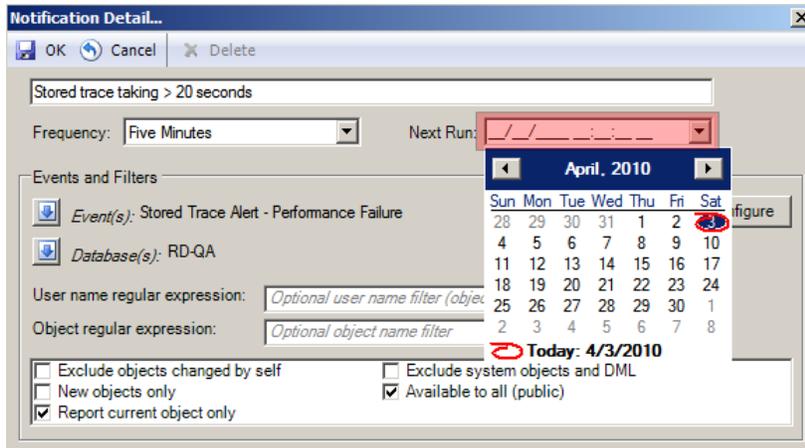
For the next fields, we fill in the desired database (RD-QA), and of the checkbox options we have, the defaults are acceptable.



To review what each of the checkboxes means:

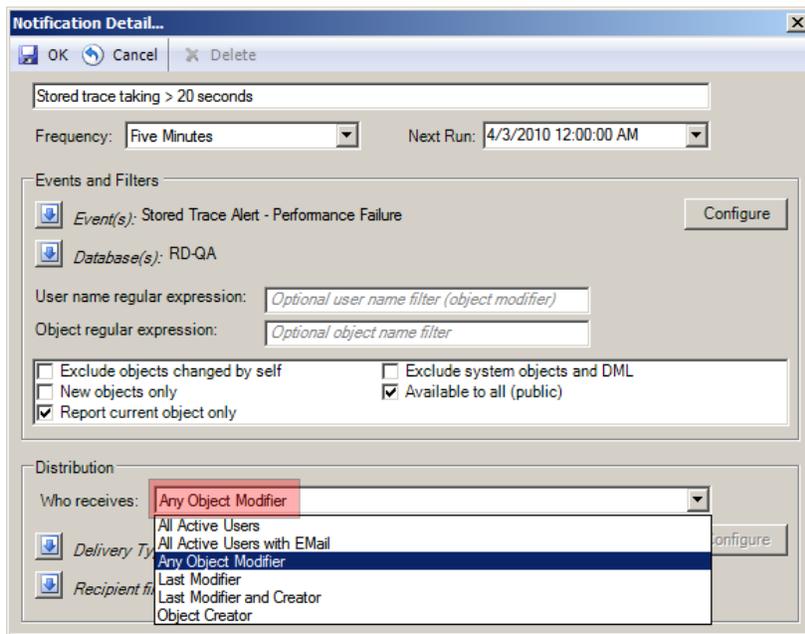
Exclude objects changed by self	If checked, the object is not included in the alert for you if you were the last modifier of the object.
New objects only	If checked, the object is only included if it is new (i.e. has no prior version in history).
Report current object only	If checked and if the object would appear multiple times (e.g. you're using the object change event type and the object was changed several times within the timeframe of the notification), you will only see the last version of the object in the notification.
Exclude system objects and DML	If checked, system objects (identified as those with the SystemGenerated user-defined property set to True) and recorded DML history entries are excluded from the alert.
Available to all (public)	If not checked, the notification will only be delivered to you, no matter what other settings are. Only SQL-Hero administrators can create public notifications.

Because the frequency is not "immediate", we also need to pick the time the notification will run next:



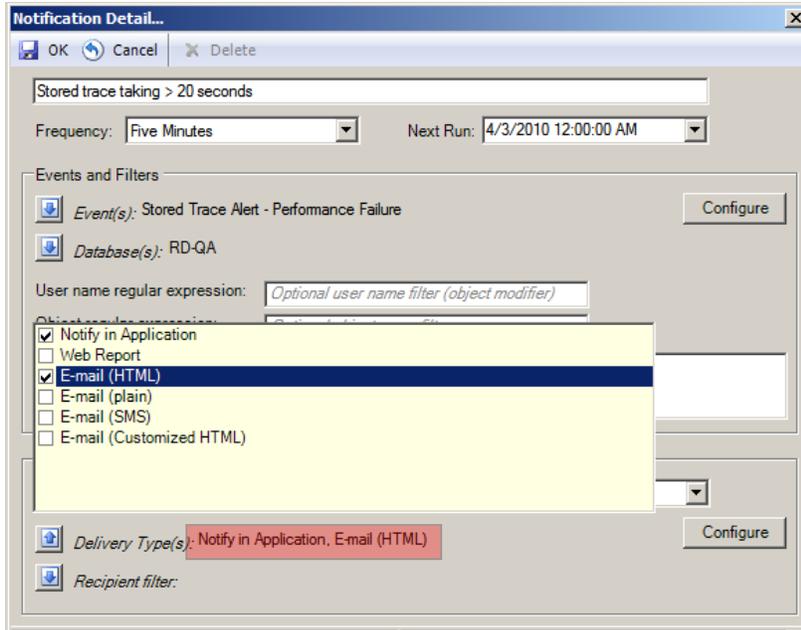
Note that if a notification does not run at the “next run” time, it will run as soon as possible after that and will then be rescheduled for the next time that it would be due to run following the current time.

The next section to fill in relates to *how* and to *whom* the notification should be delivered. Based on our example’s requirements, we want to mention offending objects to any user who modified the object. As such, we can pick “Any Object Modifier” for the “Who Receives” and this will tailor the notification accordingly.

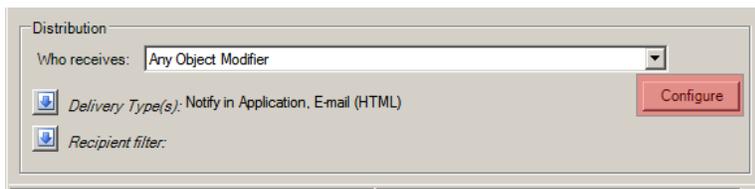


Note that in order for the notification engine to know who was object modifiers, object change tracking information must be present in the SQL-Hero central repository, for the relevant database or databases. The whitepaper “SQL-Hero History Tracking and Region Differences (Compliance)” goes into more detail on setting up change tracking.

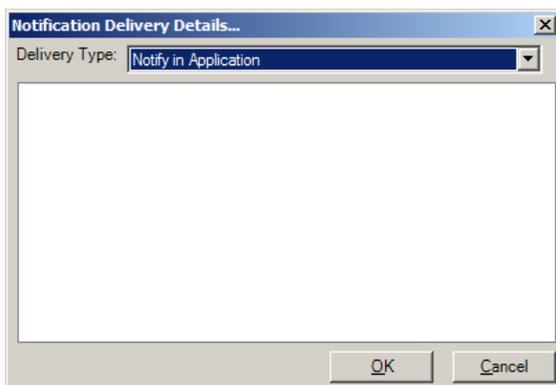
Next we can pick the delivery type or types. In this example, we’ve picked two delivery methods: notifying within the SQL-Hero application, and also by HTML e-mail.



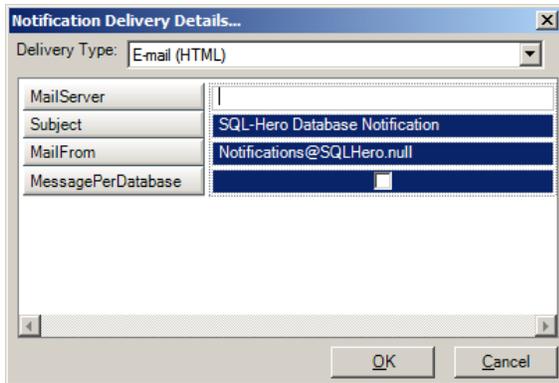
Some delivery types require configuration – such as for the e-mail variety. As such, we’ll click on the Configure button:



We’re brought to a screen that resembles the configuration for the event types, but this time it’s used to capture any parameters specific to the delivery type. The screen comes up initially like this:

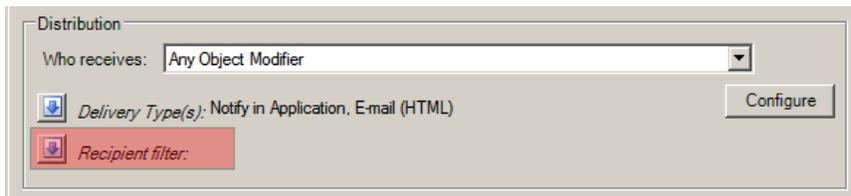


Since the “Notify in Application” delivery type does not have any parameters, we can use the drop-down to pick the e-mail delivery type:



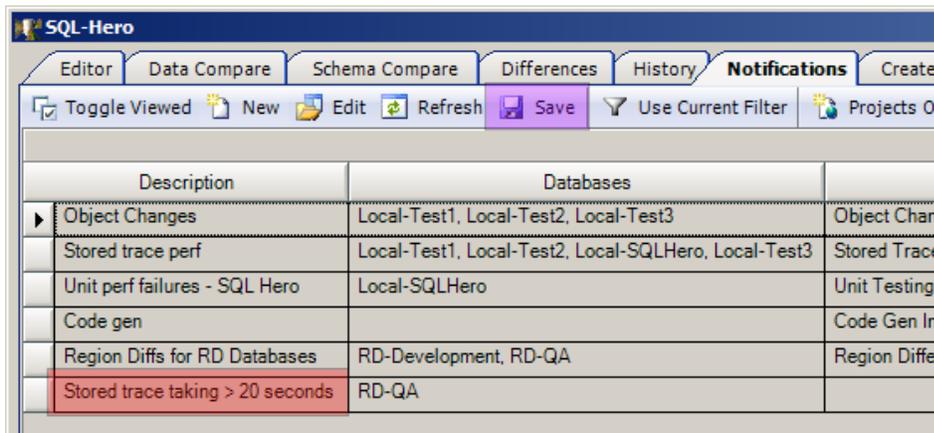
Here we can fill in our SMTP server name, plus tweak the e-mail subject and from address, if we choose. “Message per Database” if checked would split e-mails if the notification spanned multiple databases.

Back on the Notification Detail screen, we have one more thing to think about: an optional recipient filter list:



Adding users here implies that the notification cannot go to anyone else, outside of this list. This list is populated based on the current set of users as seen on the Manage Users screen. (See the whitepaper on Security, User and Connection Management for more details.)

That’s it for setting up this notification. After clicking OK, we’re returned to the Notification tool window. Notice that: 1. We have a new row in the list of notifications, and 2. The Save button is enabled.



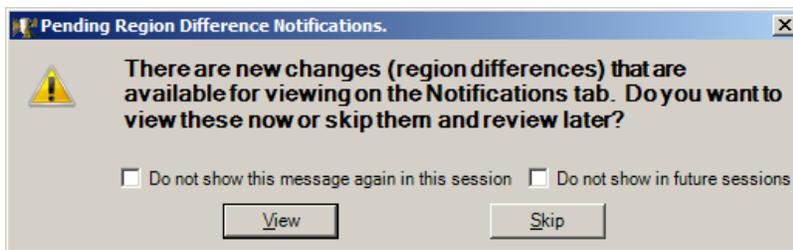
Click the Save button now to commit your new notification. To edit it later, you can select the row and click the Edit button.

Setting up all other notification types uses a similar approach to the example described above.

Notifications Inside of SQL-Hero

The “Notify In Application” delivery type means that the event being published will appear in SQL-Hero, on the Notification tool window. More specifically, it shows up in the lower grid where all event types are combined in a single view.

Let’s assume you have SQL-Hero open, a notification is run in the background and it delivers an event to you. If you are working on a different window than the Notification tool, you’ll receive a message such as:



At some point, you must acknowledge the events that are delivered to you within SQL-Hero. When you go to the Notifications tool, you might see for example:

Description	Databases	Events	Delivery Types	Frequency	
Object Changes	Local-Test1, Local-Test2, Local-Test3	Object Changes	E-mail (HTML)	Immediate	
Stored trace perf	Local-Test1, Local-Test2, Local-SQLHero, Local-Test3	Stored Trace Alert - Performance Failure	Notify in Application	Immediate	
Unit perf failures - SQL Hero	Local-SQLHero	Unit Testing - Error, Unit Testing - Performance Failure	E-mail (HTML)	Hourly	4/4/2
Code gen		Code Gen Invocation	Notify in Application	Immediate	
Region Diffs for RD Databases	RD-Development, RD-QA	Region Difference	E-mail (HTML), Notify in Application	Five Minutes	4/4/2
Stored trace taking > 20 seconds	RD-QA	Stored Trace Alert - Performance Failure	E-mail (HTML), Notify in Application	Five Minutes	4/4/2

Database	Schema	Name	Object Type	Type	Failure	
<input checked="" type="checkbox"/>	RD-QA	dbo	up_SomeChangeProc	Stored Procedure	Region Difference	Object is newer in 'RD-QA' (modified by CODEX07\Administrator, CODEX07), than in 'RD-Development' (modified by CODEX07\Administrator, CODEX07).
<input checked="" type="checkbox"/>	RD-QA	dbo	up_SomeNewProc	Stored Procedure	Region Difference	Object exists only in 'RD-QA' (modified by CODEX07\Administrator, CODEX07), not in 'RD-Development'.

To dismiss these two events and clear your grid, you need to check the checkbox next to each and click on Save. If you wanted to mark all objects as acknowledged, you can also use the “Toggle Viewed” button, highlighted above, then click Save.

Double-clicking events in the event grid generally has an action, as well, be it navigating to the object using the SQL-Hero editor, or reviewing the actions that happened as part of an invoked code gen.

Also, on the global settings screen invoked from the SQL-Hero editor tool, Manage -> Global Settings, you can stipulate how long notification messages are retained on the event grid:

