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 metadata 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:

þ	🛃 SQL-He	ero																
	Edito	Data Con	npare Sc	hema Compare	Differences	History	Notif	ications	reate Da	ta Te	sting F	Region Diffs	Tracing	Monit	oring	Settin	ngs	
	🔂 Toggle Viewed 🎦 New 🛃 Edit 😰 Refresh 🚽 Save 🛛 Use Current Filter 🐴 Projects Only 🗶 Errors Only 🏹 Send To Filter																	
													Noti	fications				
		Description			Databases					Ev	ents		[elivery Ty	pes	Frequ	Jency	Ī
	Objec	t Changes		Local-Test1, Lo	ocal-Test2, Local-	Test3		Object C	hanges				E-m	ail (HTML)	Immed	iate	T
	Store	d trace perf		Local-Test1, Lo	ocal-Test2, Local-	SQLHero,	Local-Tes	st3 Stored Ti	race Alert	- Perforn	nance Fail	lure	Not	fy in Appli	cation	Immed	iate	T
	Unit p	erf failures - S	GQL Hero	Local-SQLHero)			Unit Test	ting - Erro	r, Unit Te	sting - Pe	rformance Fa	ilure E-m	ail (HTML)	Hourly	r	Ī
	Code	gen						Code Ge	n Invocat	ion			Not	fy in Appli	cation	Immed	iate	I
	Regio	on Diffs for RD	Databases	RD-Developme	ent, RD-QA			Region D)ifference				E-m	ail (HTML)	Five M	linutes	T
													0 0	bjects				
		Database	Build Proje	ct Build Errors	Build Warnings	Schema	Name	Object Type	Туре	Failure	Number	Event Date	Processe	d User	Mach	nine :	SQL	,

This brings up the notification detail dialog:

Notification Detail	×
🛃 OK 🕥 Cancel 💥 Delete	
Notification Description	
Frequency: Immediate	
Events and Filters	
Event(s): Select at least 1 event type	
Database(s):	
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	
Who receives: All Active Users	
Delivery Type(s): Selected at least 1 delivery type	
Recipient filter:	
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:

Notification Detail	×
🛃 OK 🕥 Cancel 💥 Delete	
Stored trace taking > 20 seconds	
Frequency: Five Minutes Next Run: /_/:	•
Events and Filters	
Event(s): Stored Trace Alert - Performance Failure	Configure
Object Changes	
Region Difference Stored Trace Alert - Error	
Stored Trace Alert - Performance Failure	
Unit Testing - Error Unit Testing - Performance Failure	
Who receives: JAII Active Users	
Delivery Type(s): Selected at least 1 delivery type	Configure
Recipient filter:	

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:

Notification Detail		×
🛃 OK 🕤 Cancel 💥 Delete		
Stored trace taking > 20 seconds		
Frequency: Five Minutes	Next Run: _/_/:_:	
Events and Filters		
Event(s): Stored Trace Alert	Performance Failure Configure	
Database(s):		
User name regular expression:	Optional user name filter (object modifier)	
Object regular expression:	Optional object name filter	

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:

Notification Delivery D	etails		X
Delivery Type: Stored T	race Alert - Perfo	mance Failure	▼
MoreThanMillisecond	20000		
			Þ
		<u>0</u> K	<u>C</u> ancel

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

Notification Deliv	very Details		×
Delivery Type:	ode Gen Invocation		•
ProjectName	SHMObjectModel RD Unit Testing SHMObjectModel		
		ок	Cancel

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.

Notification Detail	×
🛃 OK 🕥 Cancel 💥 Delete	
Stored trace taking > 20 seconds	
Frequency: Five Minutes Next Run: _/_/:_:	
Events and Filters	
Event(s): Stored Trace Alert - Performance Failure Configure	
Database(s): RD-QA	
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) Available to all (public)	
Distribution	

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:

Notification Detail									X
🛃 OK 🕥 Cancel 💥 Delete									
Stored trace taking > 20 seconds	_		_		_	_			
Frequency: Five Minutes Next Run:	_/.	_/		: :			•		
Events and Filters	•	I	Ар	ril, 2	010		Þ		
Event(s): Stored Trace Alert - Performance Failure	<u>Sun</u> 28	Mon 29	Tue 30	Wed	Thu 1	Fri 2	Sat	ifigure	
Database(s): RD-QA	4	5	6	7	8	9	10		
User name regular expression: Optional user name filter (object	18	19	20	21	22	23	24		
Object regular expression: Optional object name filter	25	3	4	28 5	6	30 7	8		
Exclude objects changed by self Exclude sy: New objects only Available to Available to	stem all (object public	lay: cts ai c)	4/3/ nd DN	2010 1L)			-

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.

Notification Detail	×
🛃 OK 🕥 Cancel 💥 Delete	
Stored trace taking > 20 seconds	
Frequency: Five Minutes Next Run: 4/3/2010 12:00:00 AM	
Events and Filters	
Event(s): Stored Trace Alert - Performance Failure Configure	
Database(s): RD-QA	
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	٦.
Who receives: Any Object Modifier	
All Active Users with EMail onfigure	
Recipient fil Last Modifier and Creator	
Object Creator	

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.

Notification Detail	×
🛃 OK 🕥 Cancel 🐹 Delete	
Stored trace taking > 20 seconds	
Frequency: Five Minutes Next Run: 4/3/2010 12:00:00 AM	•
Events and Filters	
Event(s): Stored Trace Alert - Performance Failure	Configure
Jatabase(s): RD-QA	
User name regular expression: Optional user name filter (object modifier)	
Notify in Application	
Web Report	
E-mail (plain)	
E-mail (SMS)	
E-mail (Customized HTML)	
	-
Delivery Type(s): Notify in Application, E-mail (HTML)	Configure
Recipient filter:	

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

Distribution	
Who receives: Any Object Modifier	•
Delivery Type(s): Notify in Application, E-mail (HTML)	Configure
Recipient filter:	

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:

Notification Delivery Details X Delivery Type: Email (HTML)					
MailServer	SOL-Hero Database Notification				
MailFrom	Notifications@SQLHero.null				
MessagePerDatabase					
4		Þ			
	<u>о</u> к	<u>C</u> ancel			

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:

Distribution		
Who receives:	Any Object Modifier	•
Delivery Ty	/pe(s): Notify in Application, E-mail (HTML)	Configure
Recipient fi	iter:	

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.

	🕂 SQL-Hero					
		Editor Data Compare Sche	ma Compare Differences History Notification	ons Create		
	F	Toggle Viewed 붭 New 🗾 Ec	dit 😰 Refresh 🛃 Save 🛛 Use Current Filter 🕇	🏠 Projects Or		
		Description	Databases			
 Object Changes 		Object Changes	Local-Test1, Local-Test2, Local-Test3	Object Chang		
		Stored trace perf	Local-Test1, Local-Test2, Local-SQLHero, Local-Test3	Stored Trace		
		Unit perf failures - SQL Hero	Local-SQLHero	Unit Testing		
		Code gen		Code Gen Inv		
		Region Diffs for RD Databases	RD-Development, RD-QA	Region Differ		
		Stored trace taking > 20 seconds	RD-QA			

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:

Pending Region Difference Notifications.							
<u> </u>	There are new changes (region differences) that are available for viewing on the Notifications tab. Do you want to view these now or skip them and review later?						
	☐ Do not show this message again in this session ☐ Do not show in future sessions						
	<u>Vi</u> ew <u>S</u> kip						

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:

R	₽ SQL-Hero									
Editor Data Compare Schema Compare Differences History Notification				Schema Compare D	ifferences Histo	y Notification	s Create Data Testing Region Diffs Trac	ing Monitoring Settings		
Ę	Toggle Viewed 🖺 New 🛃 Edit 🕢 Refresh 🛃 Save 🛛 Use Current Filter 📸 Projects Only 🗙 Errors Only 🏹 Send To Filter									
							I and the second se	Notifications		
		Descriptio	n		Databases		Events	Delivery Types	Frequency	
	Obj	ect Changes		Local-Test1, Local-	Test2, Local-Test3		Object Changes	E-mail (HTML)	Immediate	
	Stor	red trace perf		Local-Test1, Local-	Test2, Local-SQLHe	ro, Local-Test3	Stored Trace Alert - Performance Failure	Notify in Application	Immediate	
	Uni	t perf failures - S	SQL Hero	Local-SQLHero			Unit Testing - Error, Unit Testing - Performance Failure	E-mail (HTML)	Hourly	4/4/2
	Coc	le 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			
	Stor	red trace taking	> 20 secon	ds RD-QA			Stored Trace Alert - Performance Failure	E-mail (HTML), Notify in Application	Five Minutes	4/4/2
1										
	2 Objects									
		Database	Schema	Name	Object Type	Туре		Failure		
		RD-QA	dbo	up_SomeChangeProc	Stored Procedure	Region Differenc	e Object is newer in 'RD-QA' (modified by CODEX07V	dministrator, CODEX07), than in 'RD-D	Development' (n	nodifie
RD-QA dbo up_SomeNewProc Stored Procedure Region Difference				Stored Procedure	Region Differenc	ce 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:

🕂 SQL-Hero	
Editor Data Compare Schema Compare Differences Histo	ory Notifications Create Data
Category: RD Database: RD-QA	🔹 🍸 Filter 🚰 🕨 Execute
File • Edit • Tabs • Actions • Object • Favorites • Templates •	Manage 🔹 👫 Find: count
🛛 🖻 🖻 🖀 🗩 🗿 🎝 🖄 🗑 🛠 🗐 🖺 🗙 🍇 🚽	😫 Manage Users
Stored Procedures	Tombstone Settings
E- W up_SomeChangeProc	👌 Source Control Settings 🏾
⊞… up_SomeNewProc	Global Settings
	Hanage UDP's
	JEI Server Logs
	Recorded SQL History

🦞 Global Settings 🗙
Stored Procedure Test Requirements
Must have test UDP and attempt to auto-generate
Maximum wait (milliseconds) 8000
Logging
"Log All" writes to: Both Central Repository and Local File System
Build Settings
Work item provided with forced comment adds to open build; formatting: (0)
Enable "add dependencies" check-box on add to build window
Notifications
Purge "in application" messages after (days):
<u>OK</u> <u>Cancel</u>