**Automating SQL Server Agent Notification**

**By** [**Greg Larsen**](http://www.databasejournal.com/feedback/http:/www.databasejournal.com/features/mssql/automating-sql-server-agent-notification.html)

**FROM:** [**http://www.databasejournal.com/features/mssql/automating-sql-server-agent-notification.html**](http://www.databasejournal.com/features/mssql/automating-sql-server-agent-notification.html)

If you are a DBA you probably are responsible for running and monitoring all the SQL Server Agent jobs on one or more SQL Server instances.  How do you monitor these SQL Agent jobs?   There is always the manual method of monitoring the history of all the SQL Server agent jobs.  In this article I will be discussing some automated approaches of monitoring SQL Server agent jobs.

**SQL Server Agent Notifications**

There three different kinds of SQL Server Agent Notifications you can set up: Email, Pager, and NET SEND.  In order to use the Email and Pager option you need to set up Database Mail.  Both the Pager and NET SEND options have been deprecated therefore Microsoft recommends you don’t use these options.   Therefore I will not be discussing these two options.

There are three different types of notification a SQL Server Agent job can have.  You can setup notification when the job succeeds, fails, or completes.  If you only want to be notified when a jobs fails, then you setup your SQL Agent job notification to only perform notifications when the job fails.  If you set up your SQL Agent job to notify you when the job succeeds then you will only get notified when the SQL Agent job runs successfully.   If you want to be notified if the SQL Server Agent job succeeds or fails then you need to set it up to notify you when the job completes.

When considering which option to use for notification you should consider these items:

* If you only get notified when the jobs completes successfully how do you know when the job fails?
* When you get notified when the jobs completes you will get notified for both successful and failed runs, which can lead to lots of notification.   If you get lots of successful notifications you can get conditioned to see only successfully job completions and may not notice the failed completions.
* If you are only concerned about failures, than you should only get notified for failures.

**Detecting SQL Server Agent Jobs that Have No Notification**

**Related Articles**

* [The Format() Function in SQL Server 2012](http://www.databasejournal.com/features/mssql/the-format-function-in-sql-server-2012.html)
* [Sequence Object in SQL Server 2012](http://www.databasejournal.com/features/mssql/sequence-object-in-sql-server-2012.html)
* [Contained Database Authentication in SQL Server 2012](http://www.databasejournal.com/features/mssql/contained-database-authentication-in-sql-server-2012.html)
* [Analysis Services PowerShell Provider (SQLAS) in SQL Server 2012](http://www.databasejournal.com/features/mssql/analysis-services-powershell-provider-sqlas-in-sql-server-2012.html)

You may have lots of SQL Server Agent jobs on your server instance already.  You might want to know how you can tell which ones don’t have notifications set up on them.  It is fairly easy to detect if a SQL Agent job is set up to notify someone via Email, a pager or a NET SEND message.  You can do this by reviewing the notify\_level\_email value in the sysjobs system table in the msdb database.  Here is some code that displays the name of the SQL Server Agent jobs that are not set up to send Email, a pager or a NET SEND messages:

USE msdb;

GO

SELECT name

FROM dbo.sysjobs

WHERE notify\_level\_email = 0;

In this code I verify that the notify\_level\_email value was set to zero (0).  If this column is set to 1, 2, or 3 then the job does send one of the three types of notifications.  If you run this code on a SQL Server instance it will return the name of each SQL Server agent job that isn’t set up with a notification.

If you want to verify each of your SQL Agent jobs have email notification then you can run this code:

USE msdb

SELECT name, notify\_email\_operator\_id

FROM dbo.sysjobs

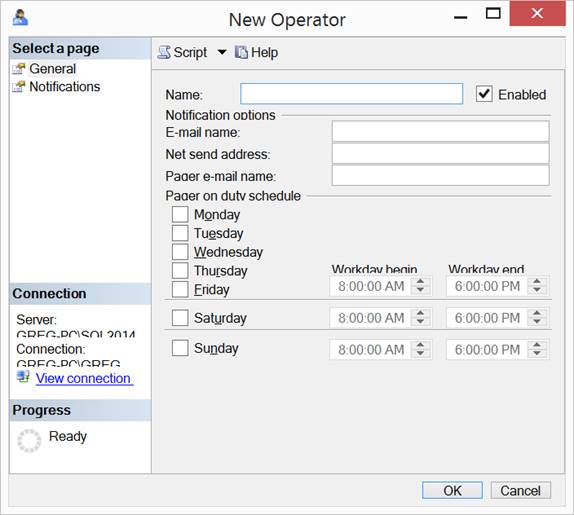
WHERE notify\_email\_operator\_id = 0;

This code will display each SQL Server Agent job that doesn’t have an operator associated with an email notification operator.

**Setting up an Operator**

In order to send email you need to have an Operator and of course you need to have database mail configured.  Configuring database mail is outside the scope of this article.  To set up an operator you have two choices.

The first choice is to use the GUI tool available in SQL Server Management Studio.  To bring up the GUI tool you can either right click on the SQL Server Agent item and then select “New” from the context menu, followed by “Operator…” from the second context menu.  Or you can expand the SQL Server Agent item and then right click on the Operator item then select the “New Operator…” item from the context menu. Regardless of which way you bring up the GUI the following window will be displayed:

  
New Operator

 On this window you enter the name of the Operator in the “Name:” item.   The email addresses where you want to send the email notifications can be entered in the “E-mail name:” field.

Another way to create an operator is by running a script.  Here is a sample script that creates a single operator:

USE [msdb]

GO

EXEC msdb.dbo.sp\_add\_operator @name=N'DBA',

              @enabled=1,

              @email\_address=N'DBA@DatabaseJournal.com'

GO

In this script I created an operator named “DBA” and associated it with the email address “DBA@DatabaseJournal.com”.

**Automatically Updating SQL Agent Jobs to Have Notification**

If you already have a number of SQL Server agent jobs where notification hasn’t been set up you might want to set up each job to notify someone.   If you have a lot of SQL Agent jobs it might take a while for you to go through each job to define an email notification operator manually through the SQL Agent GUI.  A better way to change a bunch of SQL Agent jobs to send email notifications would be to run a script.  Here is a sample script for defining an operator on each job that doesn’t already have an email operator:

USE [msdb]

GO

SET NOCOUNT ON;

DECLARE @Operator varchar(50) = 'DBA' -- place your operator name here

SELECT 'EXEC sp\_update\_job @job\_name = ''' + j.[name] +

       ''', @notify\_email\_operator\_name = ''' + @Operator  +

       ''', @notify\_level\_email = 2'   -- 1=On Success, 2=On Faulure,3=always

FROM dbo.sysjobs j

WHERE j.enabled = 1

AND j.notify\_level\_email <> 1

GO

This script doesn’t actually add the operators to SQL Agent jobs that don’t have email notification.   But instead it just creates a series of EXEC statements that when executed add the email notifications.  The series of EXEC statements that this script generates will run the sp\_update\_job system stored procedures to add an email operator.  This particular script only adds email notification on a job that doesn’t already have email notification.  Additionally, it only sets up notification to occur when the job execution fails.   If you should happen to want different operators to be notified for different jobs then you will either need to add some special code to place the appropriate operator on each job, or run the above script then modify the operators for each job prior to executing the generated script.

**Daily Monitoring Job**

Getting individual SQL Server Agent jobs notifications for failed jobs is one method of being notified of the jobs you need to look at and fix.  Another method is to get a daily report of all the jobs that failed in the last 24 hours.

There are a number of ways to accomplish this.  One way to do it would be to build a SQL Agent job that runs a query against the msdb databases tables to find failed jobs. Then once you have identified the failed jobs you can email a report to the DBAs.   If you search the web for “sql server agent job that failed in last 24 hours” in your favorite search engine you will get a number of hits that can provide you with examples of code that identifies SQL Server Agent jobs that have failed in the last 24 hours.

I find it best to get the individual job notifications and then also get a single email first thing in the morning that shows me all the SQL Server Agent job failures. This way I have a single report that shows me all the job failures for the last 24 hours.   This is especially nice for quickly identifying all the jobs that failed during our batch window when I’m not normally monitoring those individual emails as they arrive in my inbox.

**Summary**

Being a proactive DBA is better than being a reactive DBA.  One of these proactive steps is to make sure all the SQL Server agent jobs on your server notify someone when they fail.  By being notified when a SQL Agent Job fails you can quickly respond to that failure notification.  Being able to quickly resolve why a SQL Agent job failed is better than having someone else notice that the SQL Agent job has failed.  If you haven’t already set up email notifications on your SQL Server Agent jobs, then the scripts shown in this article should help you identify the SQL Agent jobs that needed email notification, as well as help you actually add email notification to all of your SQL Server Agent jobs.

[**See all articles by Greg Larsen**](http://www.databasejournal.com/article.php/1560691/Gregory-A-Larsen.htm)