

# **SQL Server 2012: Installation and Configuration**

## **Module 6: Automating Common Maintenance Tasks for SQL Server 2012**

Glenn Berry

Glenn@SQLskills.com



**pluralsight**  
hardcore developer training

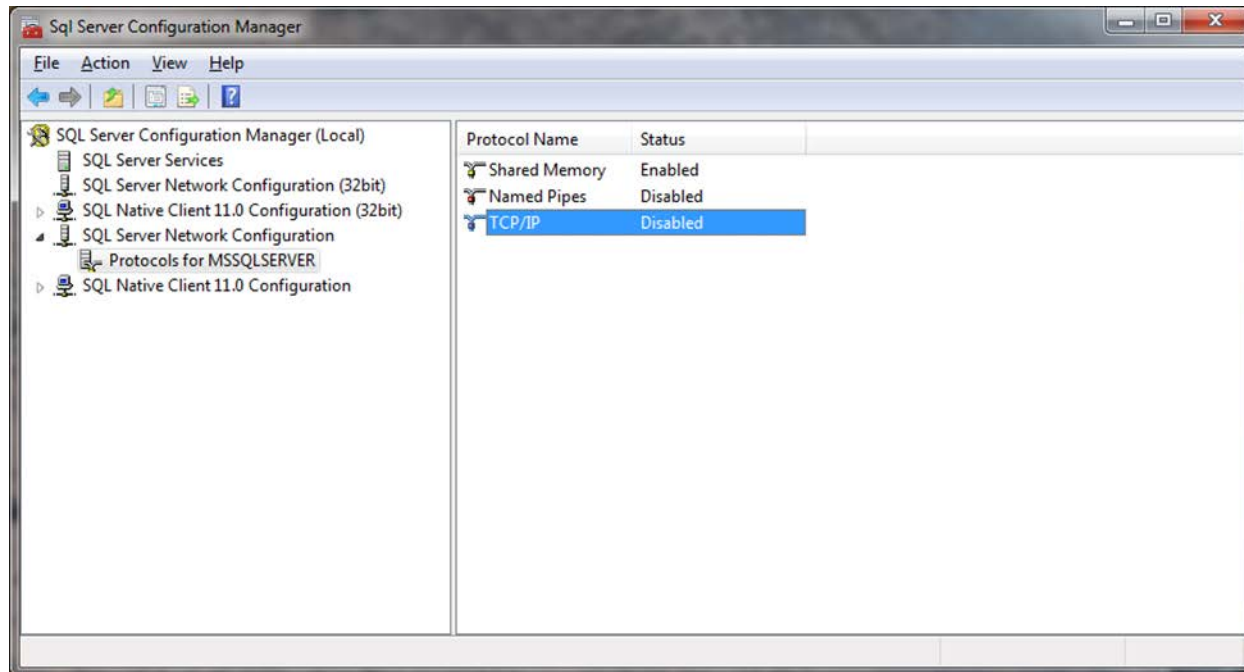
# Introduction

- **Confirming network connectivity**
- **Confirming SQL Server connectivity**
- **Enabling Database Mail**
- **Creating a New SQL operator**
- **Creating SQL Server Agent Alerts**
- **Ola Hallengren's Maintenance Solution**
- **Adding Schedules and Notifications to SQL Agent Jobs**

# Confirming Network Connectivity

- **You need to confirm that other machines can connect to your SQL Server instance over the network**
  - Use SQL Server Configuration Manager
    - Select SQL Server Network Configuration
      - Protocols for MSSQLSERVER (or your instance name)
    - Ensure that the appropriate network protocols are enabled
      - Usually you will want TCP/IP to be enabled
      - It may already be enabled (depending on what SQL Server Edition)
      - You should always double-check whether it is enabled
    - Any change to the status of these protocols will require a restart of that instance of SQL Server
      - There will be a warning prompt to remind you of this
  - You can use Ping (from a command prompt) on another machine to confirm basic network connectivity
    - You may have to open ports in your Windows firewalls
      - Ports 1433 and 1434 are the default ports for a default instance of SQL Server

# SQL Server Configuration Manager



# Confirming SQL Server Connectivity

- **Use a Microsoft Data Link to check SQL Server connectivity**
  - This does not require any special tools on the remote machine
- **Use these steps to create and use a data link file**
  - Create a blank text file on the desktop of the machine
  - Change the file extension from .txt to .udl
    - You may have to unhide file extensions in Windows Explorer
  - Double-click on the .udl file to open a Data Link Properties dialog
  - Enter the appropriate server name and logon information
  - Click on the “Test Connection” button
- **This test will confirm several items**
  - The SQL Server Service is running and a network protocol is enabled
  - The appropriate ports are open between the two machines
  - Your logon credentials can connect to the instance
  - Your logon credentials can connect to a particular database

# Microsoft Data Link Properties

The screenshot shows the 'Data Link Properties' dialog box with the 'Connection' tab selected. The dialog is titled 'Data Link Properties' and has a standard Windows XP-style window with a title bar and a close button. The 'Connection' tab is active, showing options for connecting to SQL Server data. The 'Provider' tab is also visible. The 'Advanced' and 'All' tabs are disabled. The main area contains three numbered steps for configuring the connection. Step 1 involves selecting a server name from a dropdown menu, which currently shows 'LABDB05', and a 'Refresh' button. Step 2 involves entering login information, with radio buttons for 'Use Windows NT Integrated security' and 'Use a specific user name and password'. The 'Use a specific user name and password' option is selected, and the 'User name' field contains 'sa'. The 'Password' field is masked with dots. There are checkboxes for 'Blank password' and 'Allow saving password'. Step 3 involves selecting a database on the server from a dropdown menu, which currently shows 'master'. There is also an option to 'Attach a database file as a database name' with a text field and a 'Using the filename' field. At the bottom of the dialog are 'OK', 'Cancel', and 'Help' buttons. A 'Test Connection' button is located near the bottom right of the main configuration area.

**Data Link Properties**

Provider Connection Advanced All

Specify the following to connect to SQL Server data:

1. Select or enter a server name:  
LABDB05 Refresh
2. Enter information to log on to the server:  
☐ Use Windows NT Integrated security  
☒ Use a specific user name and password:  
User name: sa  
Password: .....  
☐ Blank password ☐ Allow saving password
3. ☒ Select the database on the server:  
master  
☐ Attach a database file as a database name:  
Using the filename: .....

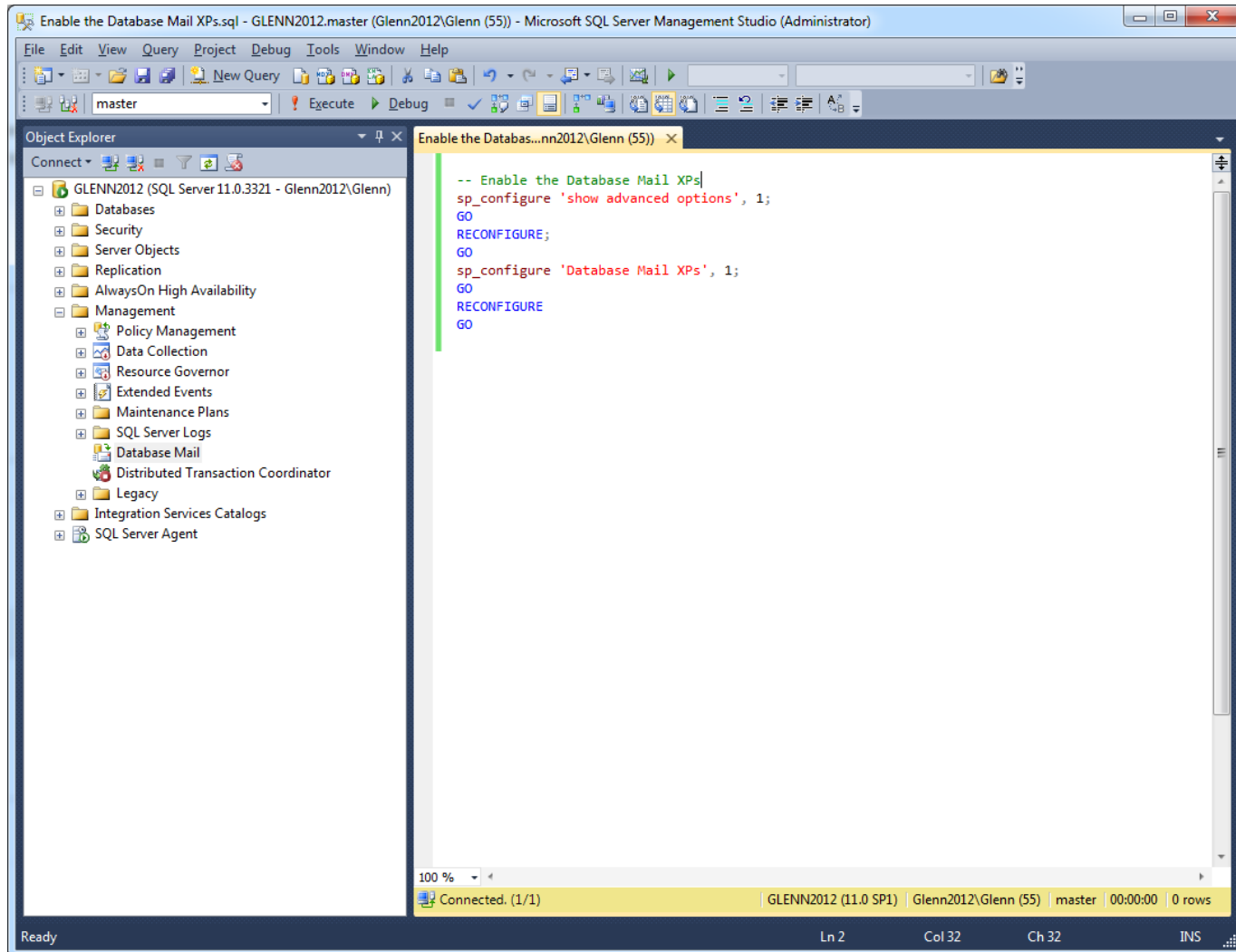
Test Connection

OK Cancel Help

# Enabling Database Mail

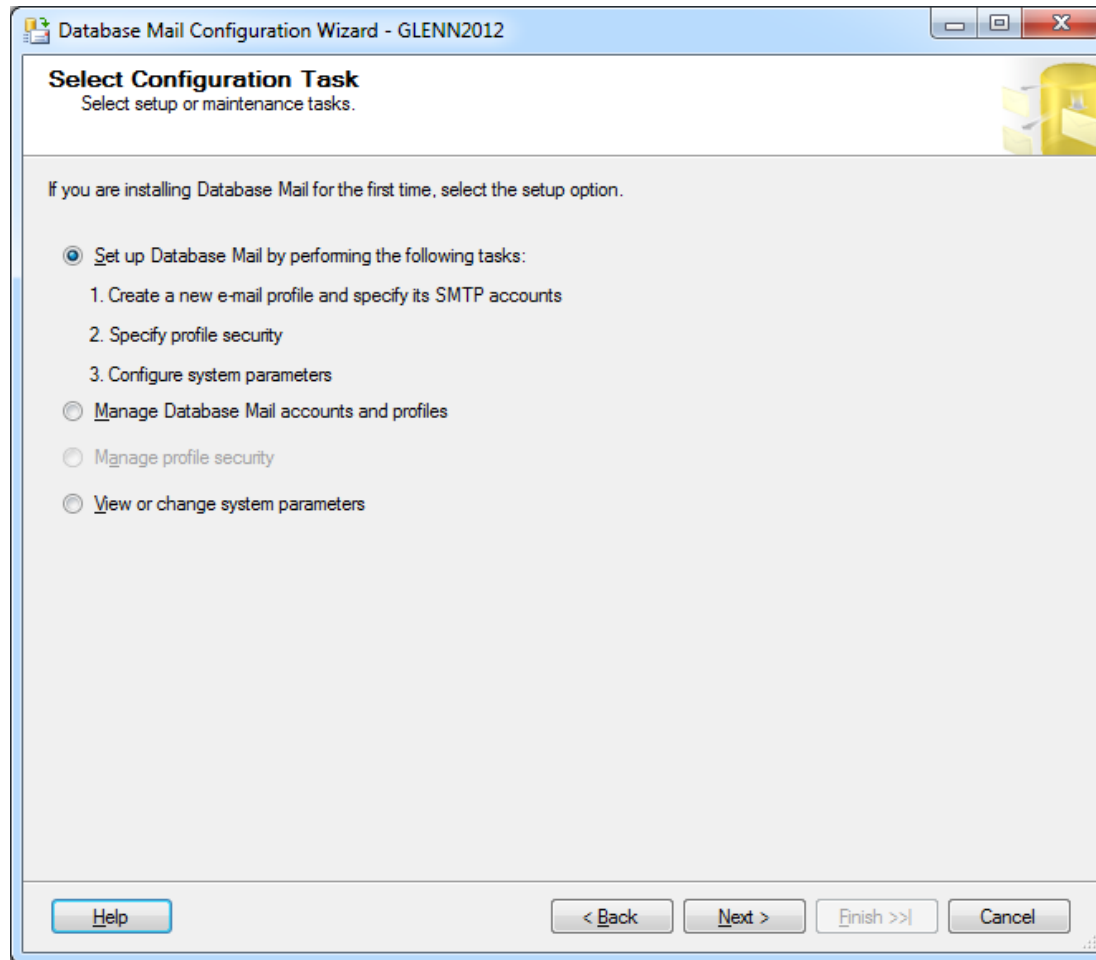
- **This is required so that SQL Server can send mail**
  - It is used for notifications and alerts about errors and failed Agent jobs
- **Follow these steps to enable Database Mail**
  - Enable the Database Mail XPs
  - Then use the Database Mail Configuration Wizard to set properties
  - Create a Database Mail account and profile
    - You must have access to an e-mail server that supports SMTP
    - You will need to know your outgoing mail server (SMTP) information
      - You may need assistance from a network administrator
    - BOL Topic: <http://bit.ly/SEZbKj>

# Enable Database Mail XPs with T-SQL





# Database Mail Configuration Wizard



# Creating a New SQL Operator

- **You need to create at least one SQL operator**
  - Operators are aliases for people or groups that can receive electronic notification when jobs have completed or alerts have been raised
  - The SQL Server Agent service supports the notification of administrators through operators
  - Operators enable the notification and monitoring capabilities of SQL Server Agent

# New Operator Dialog

**New Operator**

Script Help

Select a page

- General
- Notifications

Name: SQLDBGGroup ☒ Enabled

Notification options

E-mail name: ProductionDBGGroup@sqlskills.com

Net send address:

Pager e-mail name: ProductionDBAPager@sqlskills.com

Pager on duty schedule

Day	Workday begin	Workday end
<input checked="" type="checkbox"/> Monday	8:00:00 AM	6:00:00 PM
<input checked="" type="checkbox"/> Tuesday	8:00:00 AM	6:00:00 PM
<input checked="" type="checkbox"/> Wednesday	8:00:00 AM	6:00:00 PM
<input checked="" type="checkbox"/> Thursday	8:00:00 AM	6:00:00 PM
<input checked="" type="checkbox"/> Friday	8:00:00 AM	6:00:00 PM
<input type="checkbox"/> Saturday	8:00:00 AM	6:00:00 PM
<input type="checkbox"/> Sunday	8:00:00 AM	6:00:00 PM

Connection

Server: GLENN2012

Connection: Glenn2012\Glenn

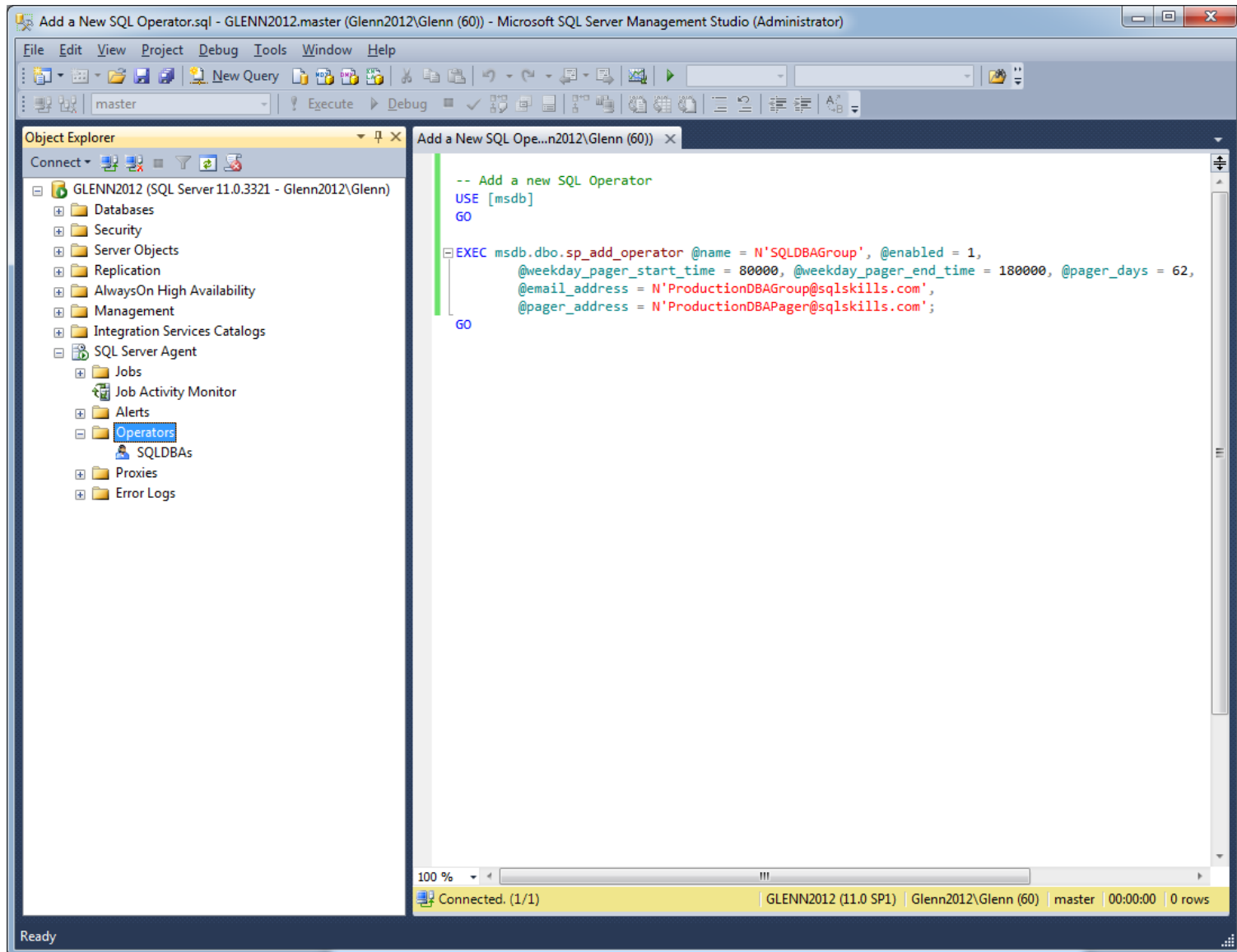
[View connection properties](#)

Progress

Ready

OK Cancel

# Create a New Operator with T-SQL



# Creating SQL Server Agent Alerts

- **You need to create SQL Server Agent alerts for certain critical errors**
  - Severity 19 through Severity 25 errors
  - Error 825 – Read-retry required
  - Read this Paul Randal blog post
    - <http://bit.ly/XuxbMA>
- **You want to know about these errors as soon as possible!**
  - Make sure to configure a response for each alert
- **A SQL Server Agent alert can trigger an e-mail, a net send, or a page**
  - An e-mail to a distribution group is the preferred method
  - Net send and paging are deprecated in SQL Server 2012

# SQL Agent Alert General Properties

'GLENN2012 Alert - Error 825: Read-Retry Required' alert properties

Select a page

- General
- Response
- Options
- History

Script Help

Name: GLENN2012 Alert - Error 825: Read-Retry Required ☒ Enable

Type: SQL Server event alert

Event alert definition

Database name: <all databases>

Alerts will be raised based on:

☒ Error number: 825

☐ Severity: 001 - Miscellaneous System Information

☐ Raise alert when message contains:

Message text:

Connection

Server: GLENN2012

Connection: Glenn2012\Glenn

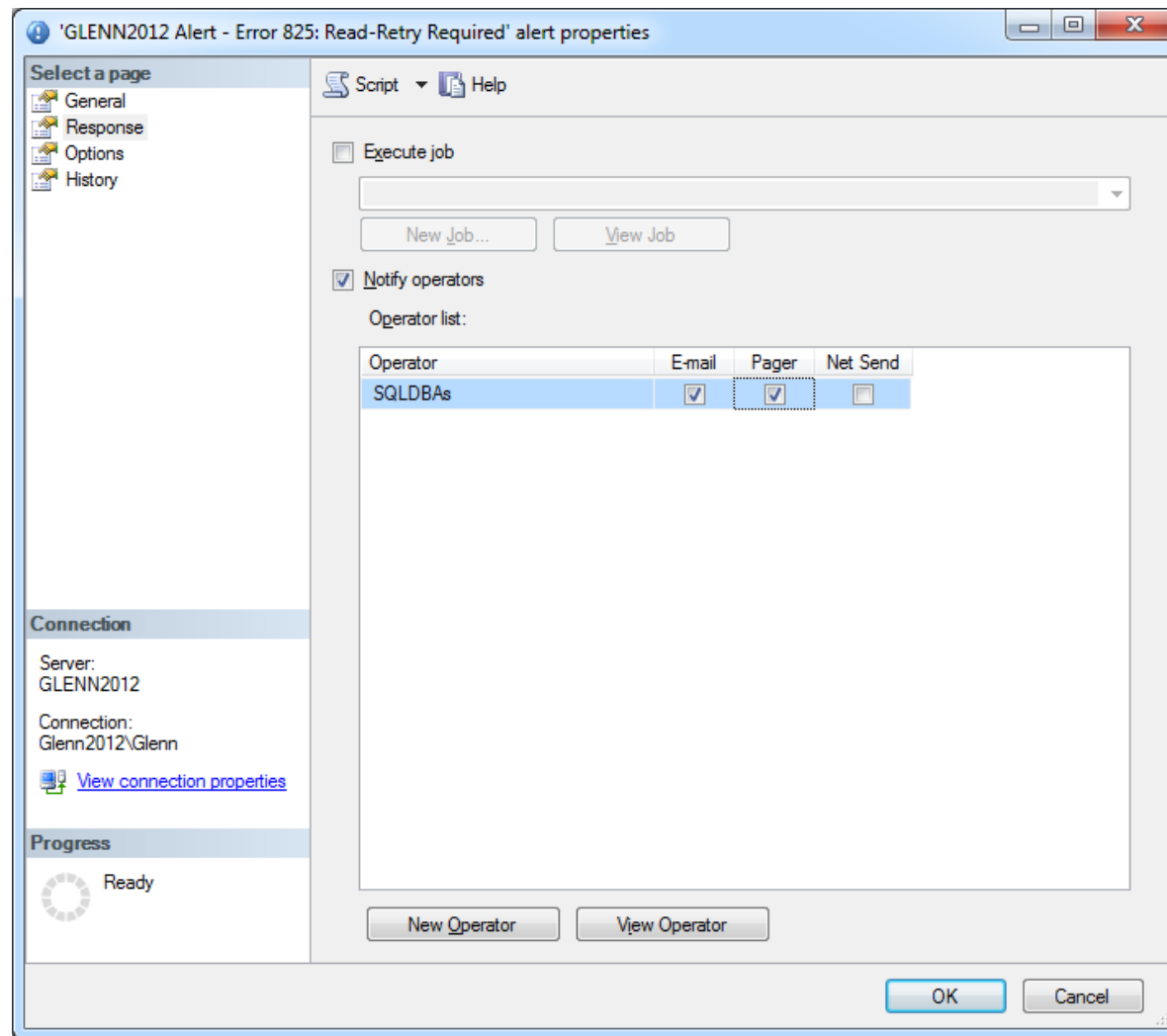
[View connection properties](#)

Progress

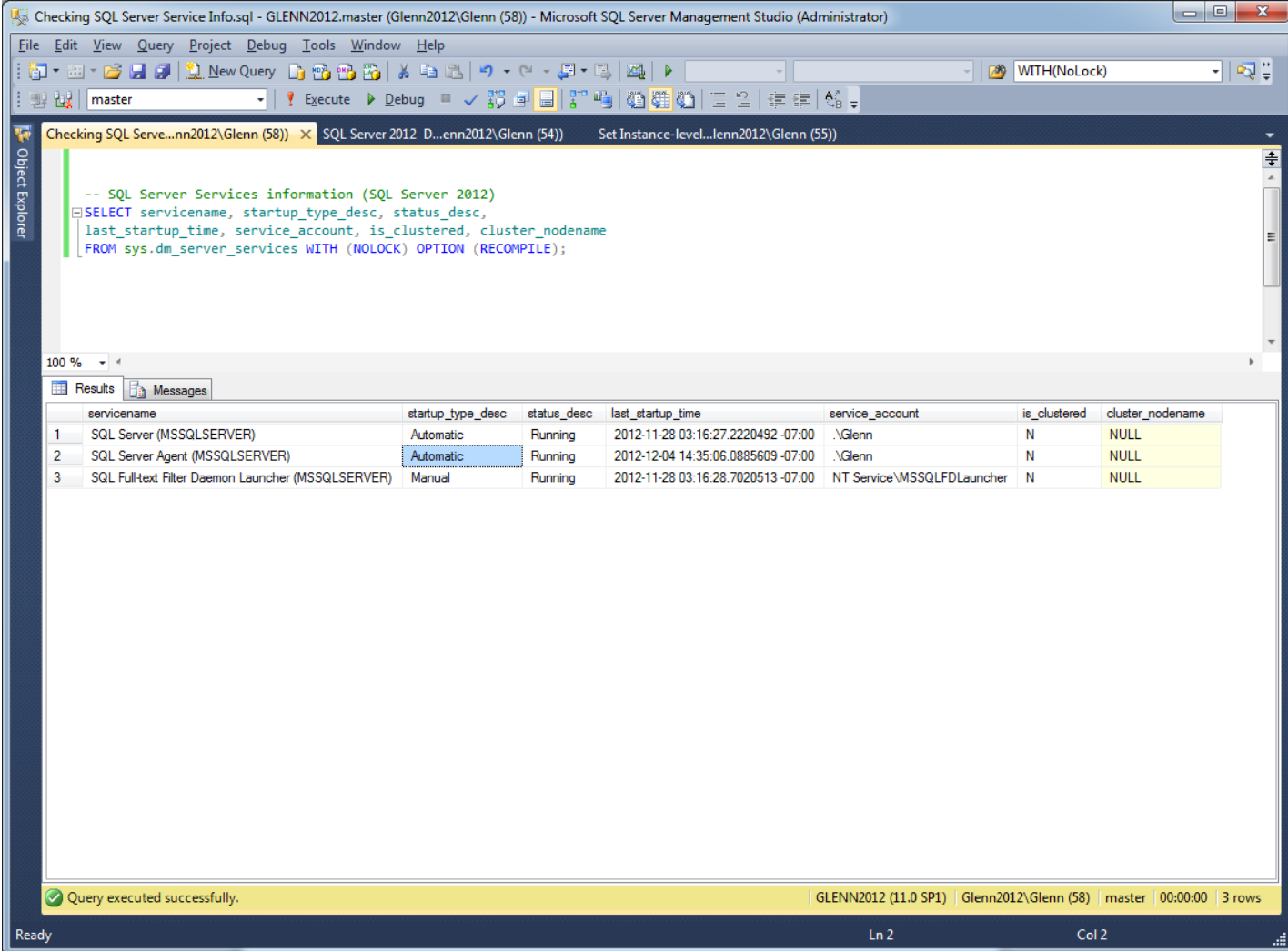
Ready

OK Cancel

# SQL Agent Alert Response Options



# Checking SQL Server Service Properties

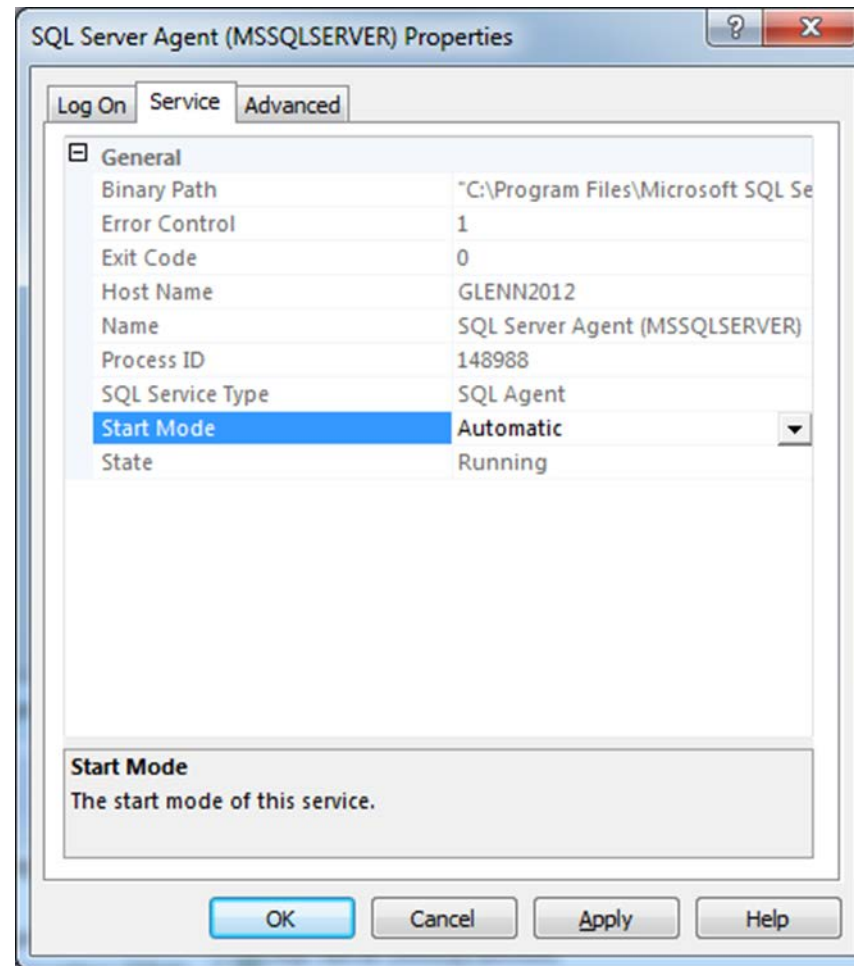


The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The title bar indicates the current session is 'Checking SQL Server Service Info.sql - GLENN2012.master (Glenn2012\Glenn (58)) - Microsoft SQL Server Management Studio (Administrator)'. The menu bar includes File, Edit, View, Query, Project, Debug, Tools, Window, and Help. The toolbar contains various icons for file operations, query execution, and debugging. The 'master' database is selected in the 'Database' dropdown. The 'Object Explorer' on the left shows the 'SQL Server 2012' instance. The 'Query Editor' displays a SQL query:   
`-- SQL Server Services information (SQL Server 2012)`  
`SELECT servicename, startup_type_desc, status_desc,`  
`last_startup_time, service_account, is_clustered, cluster_nodename`  
`FROM sys.dm_server_services WITH (NOLOCK) OPTION (RECOMPILE);`  
The 'Results' pane shows a table with 7 columns: servicename, startup\_type\_desc, status\_desc, last\_startup\_time, service\_account, is\_clustered, and cluster\_nodename. The table contains 3 rows of data. The status bar at the bottom indicates 'Query executed successfully.' and 'GLENN2012 (11.0 SP1) | Glenn2012\Glenn (58) | master | 00:00:00 | 3 rows'.

	servicename	startup_type_desc	status_desc	last_startup_time	service_account	is_clustered	cluster_nodename
1	SQL Server (MSSQLSERVER)	Automatic	Running	2012-11-28 03:16:27.2220492 -07:00	.\Glenn	N	NULL
2	SQL Server Agent (MSSQLSERVER)	Automatic	Running	2012-12-04 14:35:06.0885609 -07:00	.\Glenn	N	NULL
3	SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Manual	Running	2012-11-28 03:16:28.7020513 -07:00	NT Service\MSSQLFDLauncher	N	NULL



# SQL Server Agent Start Mode Property



# Ola Hallengren's Maintenance Solution

- **Ola Hallengren has developed and maintains an excellent script**
  - Handles database backups
    - Full, differential, and log backups for user databases
    - Full backups for system databases
  - Index maintenance
    - Intelligently reorganizes or rebuilds as necessary
    - It also does statistics maintenance
  - Database integrity checks with DBCC CHECKDB
    - User databases
    - System databases
  - Very flexible and configurable
- **In Production use with many organizations around the world**
- **It is a free download available at this link**
  - <http://bit.ly/W528k>

# Implementing the Maintenance Solution Scripts

- **Download the MaintenanceSolution.sql file**
  - Open the MaintenanceSolution.sql script in SSMS
  - Change this line in the script to point to your backup directory, by replacing “C:\Backup” with the path to your directory:  
SET @BackupDirectory = N'M:\Backup'
  - Execute the MaintenanceSolution.sql script in SSMS
    - This will create eleven new SQL Agent jobs
  - Run each job manually to verify that it is working correctly
  - Create a job schedule for each new SQL Agent job
    - This will depend on your RTO/RPO objectives and infrastructure
  - Add a notification to each new SQL Agent job
    - This will let you know when a job fails

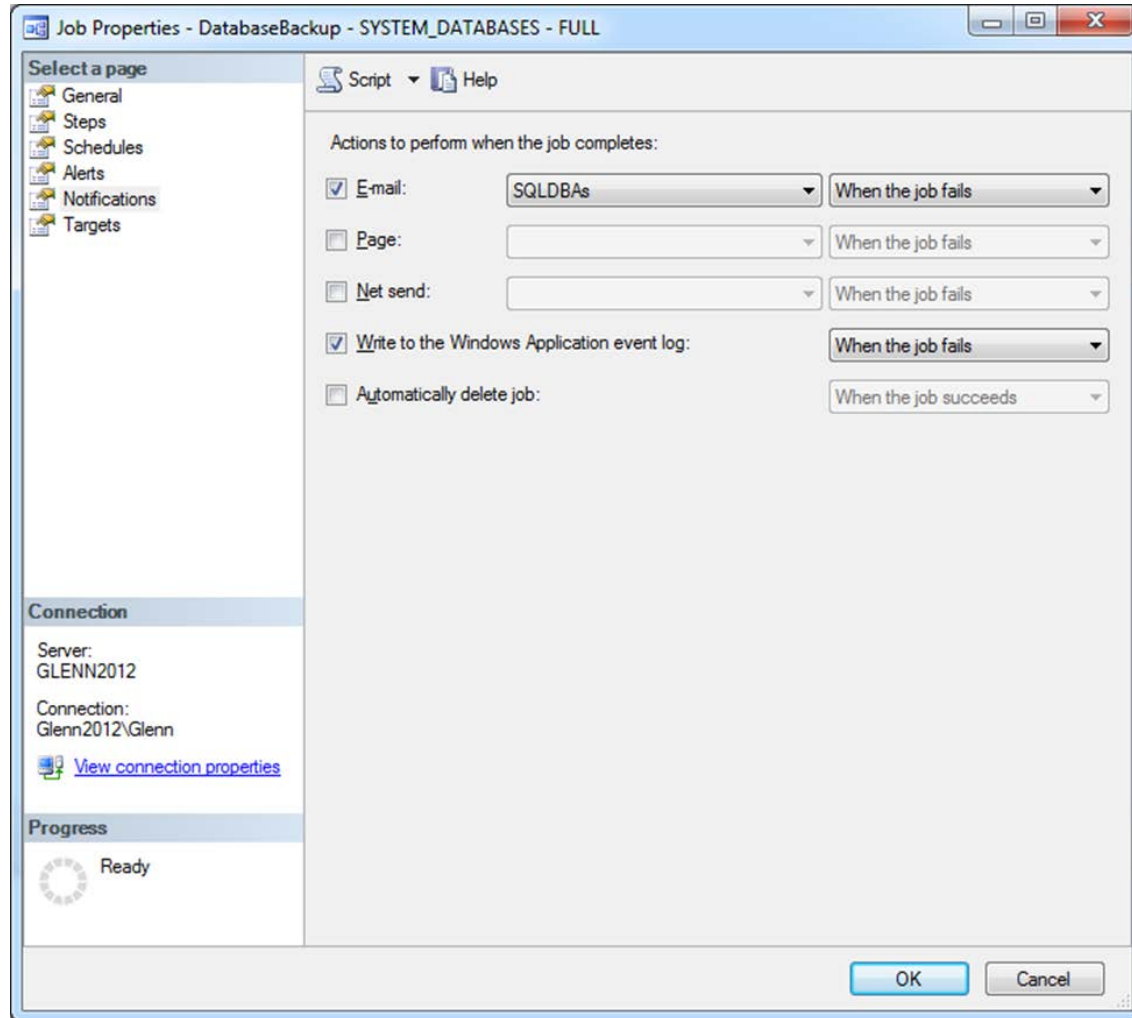
# Adding a Job Schedule to an Agent Job

The screenshot shows the 'New Job Schedule' dialog box with the following configuration:

- Name:** DatabaseBackup - SYSTEM\_DATABASES - FULL
- Schedule type:** Recurring
- Enabled:** ☒
- One-time occurrence:**
  - Date: 9/10/2012
  - Time: 9:28:21 PM
- Frequency:**
  - Occurs: Daily
  - Repeats every: 1 day(s)
- Daily frequency:**
  - ☒ Occurs once at: 12:00:00 AM
  - ☐ Occurs every: 1 hour(s)
  - Starting at: 12:00:00 AM
  - Ending at: 11:59:59 PM
- Duration:**
  - Start date: 9/10/2012
  - ☐ End date: 9/10/2012
  - ☒ No end date:
- Summary:**
  - Description: Occurs every day at 12:00:00 AM. Schedule will be used starting on 9/10/2012.

Buttons at the bottom: OK, Cancel, Help.

# Adding a Notification to an Agent Job



# Suggested Default Maintenance Job Schedule

Job Name	Run Schedule
CommandLog Cleanup	Every Sunday at 12:00AM
DatabaseBackup - SYSTEM_DATABASES – FULL	Every day at 11:55PM
DatabaseBackup - USER_DATABASES – DIFF	Every day at 12:00PM
DatabaseBackup - USER_DATABASES – FULL	Every day at 12:00AM
DatabaseBackup - USER_DATABASES – LOG	Every 30 minutes
DatabaseIntegrityCheck - SYSTEM_DATABASES	Every Saturday at 9:00AM
DatabaseIntegrityCheck - USER_DATABASES	Every Saturday at 9:00AM
IndexOptimize - USER_DATABASES	Every Monday at 3:00AM
Output File Cleanup	Every Sunday at 12:01AM
sp_delete_backuphistory	Every Sunday at 12:02AM
sp_purge_jobhistory	Every Sunday at 12:03AM

# Course Summary

- **There are many steps to a complete SQL Server 2012 installation**
  - Hardware configuration
  - Operating system configuration
  - Proper SQL Server installation
- **There are a number of post-installation configuration tasks**
  - Installing the latest Service Pack and Cumulative Update
  - Setting instance properties
  - Configuring tempdb
- **There are important alerting and maintenance tasks**
  - Enabling Database Mail
  - Creating SQL Agent Alerts
  - Using Ola Hallengren's Maintenance Solution
- **Thanks for watching!**