Module 05 Notes

Automation

# ****Module Overview****

In this module, you learn about how to automate tasks using SQL Server Agent, SSIS, and Windows Task Scheduler.

# ****Required Software****

Install SQL Developer Edition (or equivalent), Visual Studio 2019 Community edition with the SSIS Extension, and SSMS.

# Assignment

Each week you have to perform an assignment. Let us review this week's assignment.

# SQL Server Agent

"SQL Server Agent uses **SQL Server** to **store job** information. **Jobs contain one or more job steps**. Each **step contains its own task, for example, backing up a database**.

SQL Server Agent can **run a job on a schedule**, **in** **response to a specific event**, **or on demand**. For example, if you want to back up all the company servers every weekday after hours, you can automate this task. Schedule the backup to run after 22:00 Monday through Friday; if the backup encounters a problem, SQL Server Agent can record the event and notify you." (<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent>, 2017)

## Starting SQL Server Agent

SQL Server Agent may need to **start using the SQL Server Configuration Manager** application. If you cannot find this in the start menu using can launch it directly from your drive.

*"Here are the paths to the last five versions when Windows is installed on the C drive…*

*SQL Server 2019 C:\Windows\SysWOW64\SQLServerManager15.msc*

*SQL Server 2017 C:\Windows\SysWOW64\SQLServerManager14.msc*

*SQL Server 2016 C:\Windows\SysWOW64\SQLServerManager13.msc*

*SQL Server 2014 (12.x) C:\Windows\SysWOW64\SQLServerManager12.msc*

*SQL Server 2012 (11.x) C:\Windows\SysWOW64\SQLServerManager11.msc*

*"* <https://docs.microsoft.com/en-us/sql/relational-databases/sql-server-configuration-manager?view=sql-server-ver15>

### Issue: XPs are disabled

On some installations, you will see an error stating that the XPs are disabled. **When this option is not enabled, the SQL Server Agent node is not available** in SQL Server Management Studio Object Explorer.

To fix this, run the following code.

Use Master

Go

-- You verify that eXternal Procedures can run on this server

-- by viewing the "Agent XPs" option with this Sproc.

Exec sp\_configure 'Show advanced options', 1 -- Turn on adv. options

Go

Reconfigure with override -- Make the change

go

exec sp\_configure 'Agent XPs' -- See the setting

Go

Exec sp\_configure 'Agent XPs', 1 -- If Zero set to One

Go

Reconfigure with override -- Make the change

Go

Exec sp\_configure 'Show advanced options', 0 -- Turn on adv. options

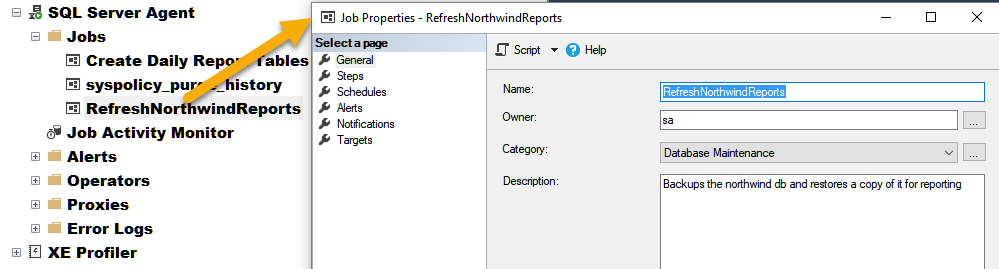
Go

Reconfigure with override -- Make the change

Go

## Jobs

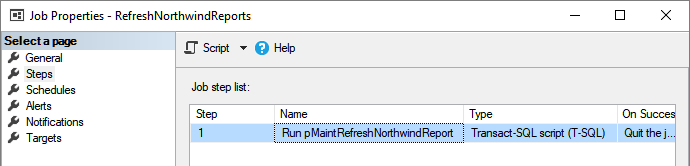
"You can **use SQL Server Agent jobs to automate routine administrative** tasks and run them on a recurring basis, making administration more efficient." (<https://docs.microsoft.com/en-us/sql/ssms/agent/implement-jobs>, 2017)



The properties dialog of a SQL Server Agent job

## Steps

**Jobs** are a set of **one or more steps** that you want to automate. Each **step** will **contain** **one command** (by **default,** this is a batch of **SQL** code.)



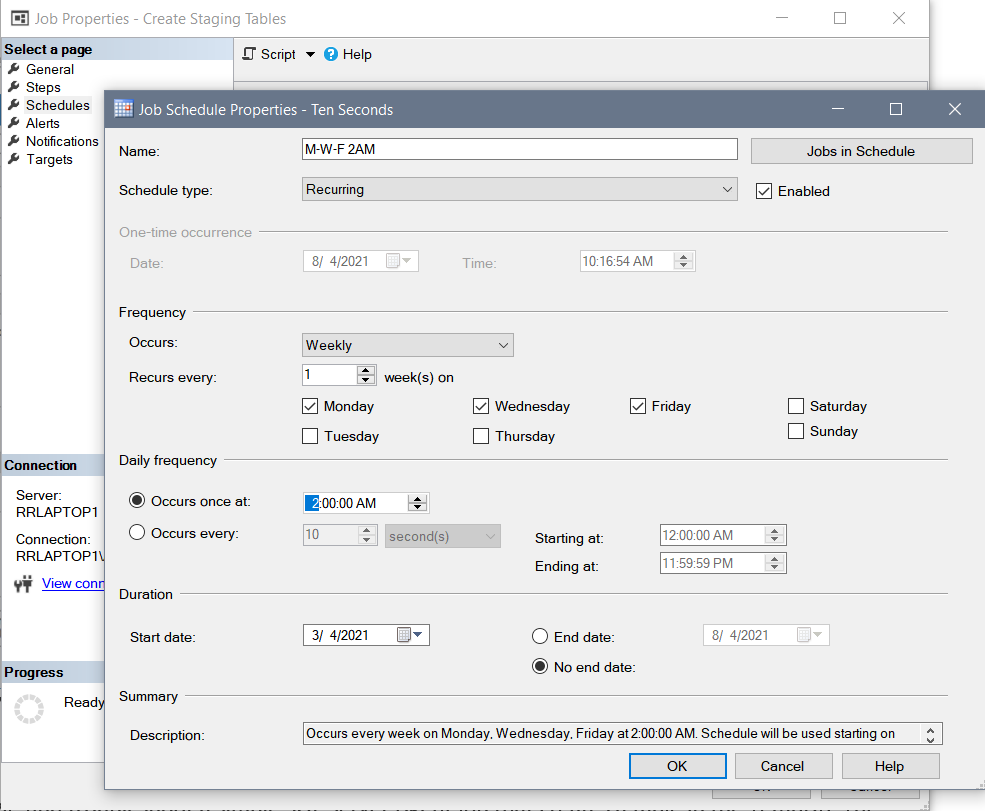
An example step in a SQL Server Agent job

## Schedules

"A schedule specifies when a job runs. More than one job can run on the same schedule, and more than one schedule can apply to the same job. A schedule can define the following conditions for the time when a job runs:

* Whenever SQL Server Agent starts.
* Whenever CPU utilization of the computer is at a level you have defined as idle.
* One time, at a specific date and time.
* On a recurring schedule."

(<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent>, 2017)



## Demo 1: Automate Creating a Table

Let's seehow to create, test, and troubleshoot a simple SQL Server Agent job that creates a table in the TempDB.

Use tempdb;

go

If Exists(Select [Name]

From [Sys].[Tables]

Where [Name] = 'StudentsStaging')

Drop Table StudentsStaging;

Go

Create Table StudentsStaging

([StudentID] nvarchar(100)

,[StudentFirstName] nvarchar(100)

,[StudentLastName] nvarchar(100)

,[StudentEmail] nvarchar(100)

);

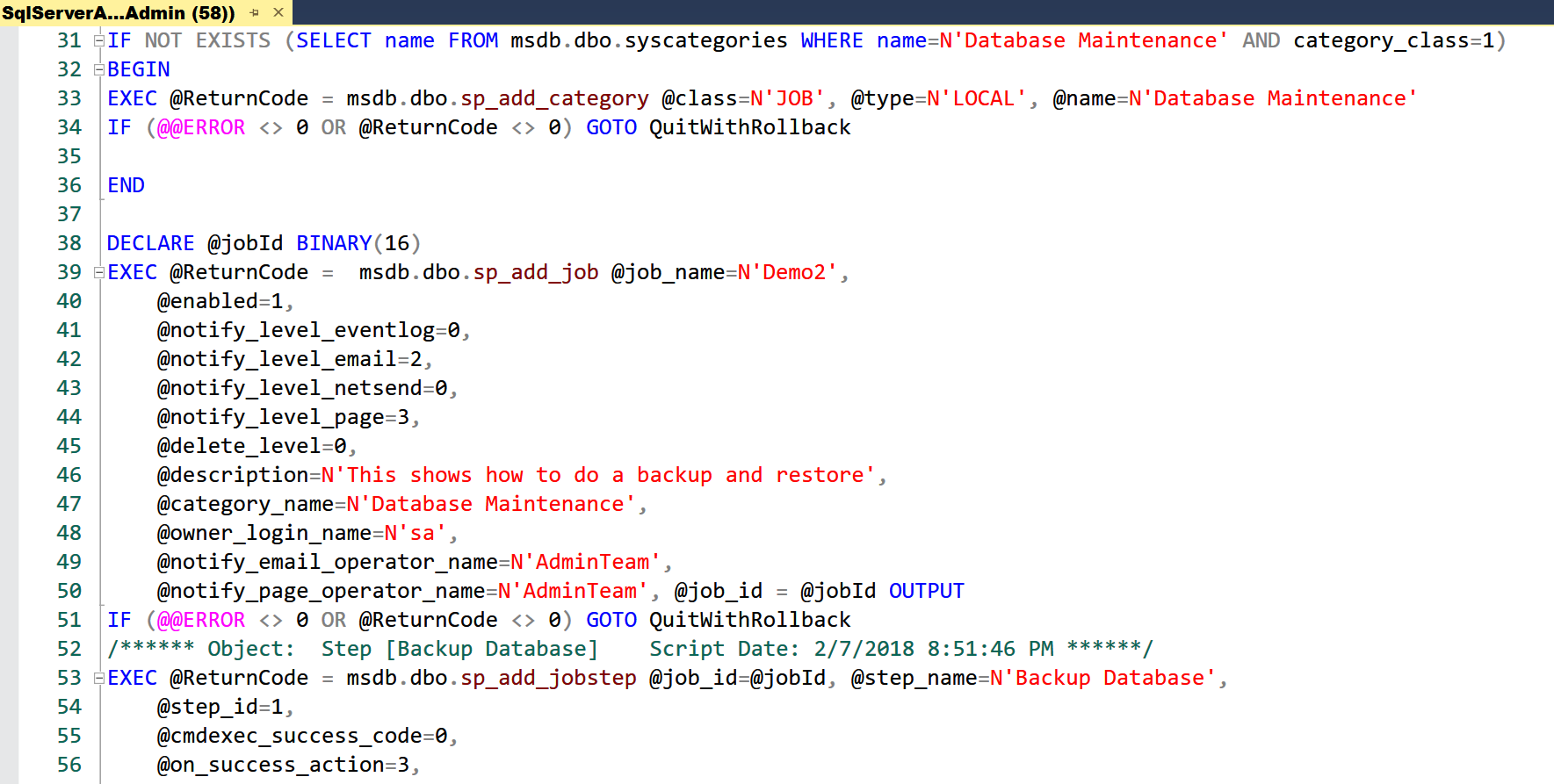
go

Select \* From StudentsStaging;

# Scripting SQL Server Agent Objects

Just like a view or stored procedure the SQL Server agent objects our code in a database. Whenever you create a job or any other objects in SQL Server Agent you are creating entries into a set of tables. These tables are located in the Microsoft database (MSDB).

***"sp\_add\_schedule*** *(Transact-SQL)… Creates a schedule that can be used by any number of jobs."* <https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-add-schedule-transact-sql>?

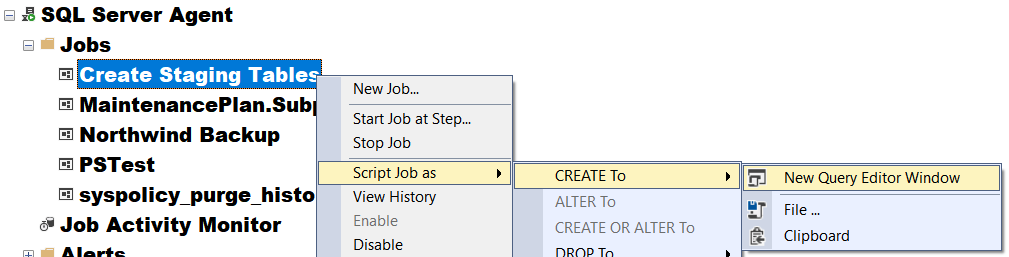


## Bitwise Mapping

"[The Microsoft documentation for the sysschedules table](https://docs.microsoft.com/en-us/sql/relational-databases/system-tables/dbo-sysschedules-transact-sql) assigns values of **1, 2, 4, 8, 16, 32, and 64** progressively for the days within a weekly schedule from Sunday (1), Monday (2), Tuesday (4) through Saturday (64). The & bitwise operator compares the 8-bit byte pattern for a day to the 8-bit byte pattern for the value denoting the days on which a schedule may run. For example, **if a schedule may run on Sunday and Saturday, its byte pattern is 10000010, which corresponds to the decimal value of 65**. The Sunday & bitwise comparison would be for 10000000 versus 10000010, which returns a value of 10000000 for Sunday, and the Saturday comparison would be for 00000010 versus 10000010, which returns a value of 00000010 for Saturday. The 8-bit byte pattern for a schedule that launches jobs on Monday, Tuesday, Wednesday, Thursday, Friday, and Saturday is 01111110, which equals a decimal value of 126. A succession of & operators in the code below deciphers start day byte patterns into the sequence of days for the Days field value in the result set." <https://www.mssqltips.com/sqlservertip/5019/sql-server-agent-job-schedule-reporting/>

## Demo 2: Scripting SQL Agent Objects

In this demo, you will see how to create SQL scripts for SQL agent objects.



Open this link for demo:

<https://docs.microsoft.com/en-us/sql/relational-databases/system-stored-procedures/sp-add-schedule-transact-sql>?

## Alerts

"An alert is an automatic **response to a specific event**. For example, an event can be a job that starts or system resources that reach a specific threshold. You define the conditions under which an alert occurs.

An alert can respond to one of the following conditions:

* **SQL Server** **events**
* SQL Server **performance** **conditions**
* Microsoft Windows Management Instrumentation **(WMI) events** on the computer where SQL Server Agent is running

An alert can perform the following actions:

* **Notify** one or more operators
* **Run** a job"

(<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent>, 2017)

## Operators

"An operator defines contact information for an individual responsible for the maintenance of one or more instances of SQL Server. In some enterprises, operator responsibilities are assigned to one individual. In enterprises with multiple servers, many individuals can share operator responsibilities. An operator does not contain security information and does not define a security principle.

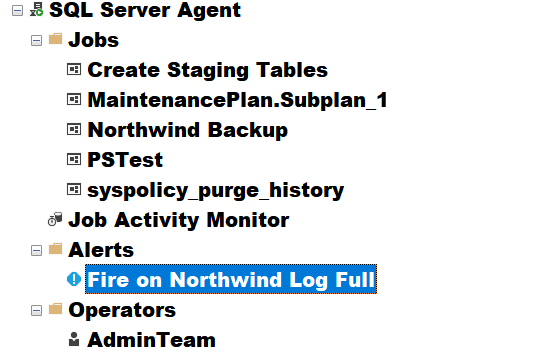
SQL Server can notify operators of alerts through one or more of the following:

* **E-mail**
* Pager (through **e-mail**)
* **net send**"

(<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent>, 2017)

## Demo 3: Alerts and Operators

Let's see how to create a SQL Server Agent Alert and Operator.



EXEC msdb.dbo.sp\_add\_alert @name=N'Fire on Northwind Log Full',

@message\_id=9002,

@severity=0,

@enabled=1,

@delay\_between\_responses=0,

@include\_event\_description\_in=0,

@database\_name=N'Northwind',

@category\_name=N'[Uncategorized]',

@job\_id=N'00000000-0000-0000-0000-000000000000'

GO

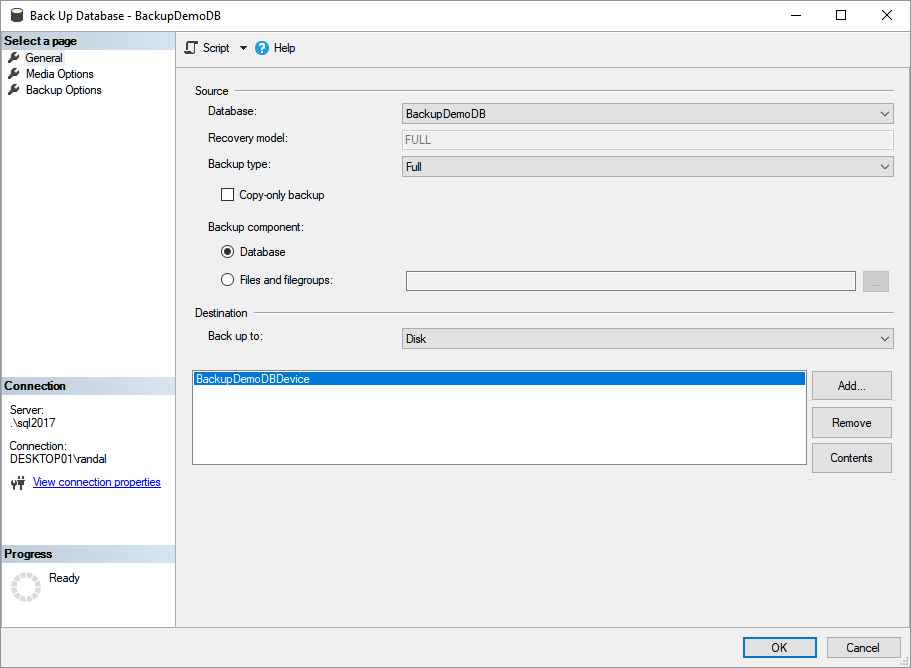
-- <https://docs.microsoft.com/en-us/sql/relational-databases/system-tables/dbo-sysnotifications-transact-sql>?

## Automating Backups

A common type of automated job is backing up a database. **Backups of your data should always be a top concern** in any organization. In addition, there **should always be a restoration and testing process for each backup** process created!

### Backup Basics

"To **minimize** the **risk** of catastrophic data loss, you need to back up your databases to **preserve** modifications to your **data** on a regular basis. A well-planned backup and restore strategy helps **protect** databases against **data loss caused by a variety of failures**. **Test** your strategy **by restoring** a set of backups and then recovering your database to prepare you to respond effectively to a disaster." ( <https://docs.microsoft.com/en-us/sql/relational-databases/backup-restore/back-up-and-restore-of-sql-server-databases>, 2017)



## Demo 4: Automating a Backup

Let's see how to create, test, troubleshoot, and automate a simple SQL Server Agent job that backs up the Northwind database. The code for this demo is found in the "Demo and Lab Files\Backup and Restore Demo Script.sql" file.

## Error Logs

" SQL Server maintains up to nine SQL Server Agent error logs. Each archived log has an extension that indicates the relative age of the log. For example, an extension of .1 indicates the newest archived error log, and an extension of .9 indicates the oldest archived error log."

(<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent-error-log>, 2017)

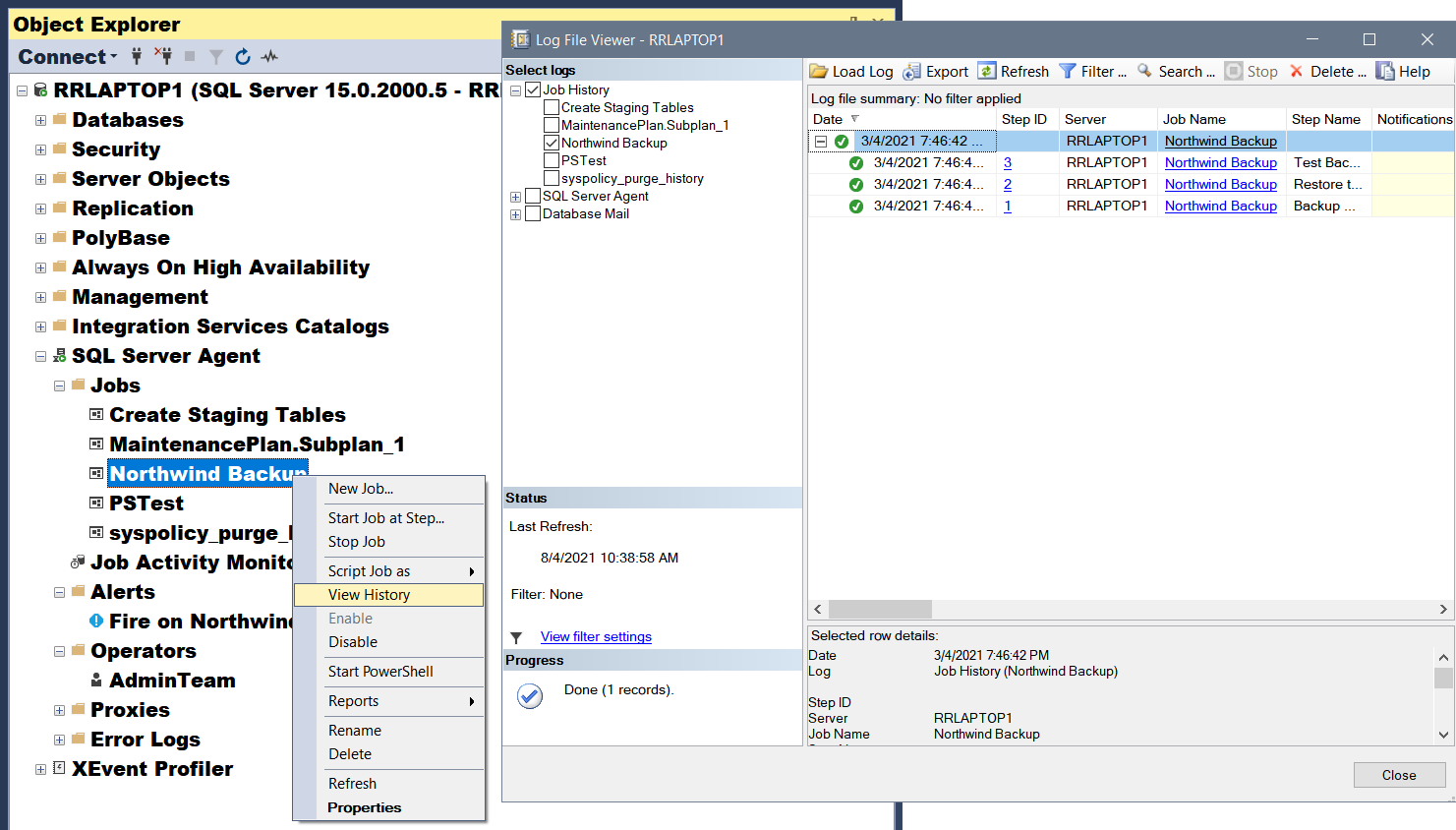
### Automation Reports

A Job includes properties that are configured with either the SSMS UI or with SQL code. These **properties are stored in the MSDB database**. You can use various views and tables to access this information.

*"The topics in this section describe the system tables that store information used by SQL Server Agent. All tables are in the dbo schema in the msdb database." (*<https://docs.microsoft.com/en-us/sql/relational-databases/system-tables/sql-server-agent-tables-transact-sql>)

## Demo 5: SQL Server Agent Error Logs

Let's seehow to view the SQL Server Agent Error Logs.



-- EXEC MSDB.dbo.sp\_purge\_jobhistory;

Select

[JobName] = j.name

,[StepName] = h.step\_name

,[RunDateTime] = msdb.dbo.agent\_datetime(run\_date, run\_time)

,[RunDurationSeconds] = h.run\_duration

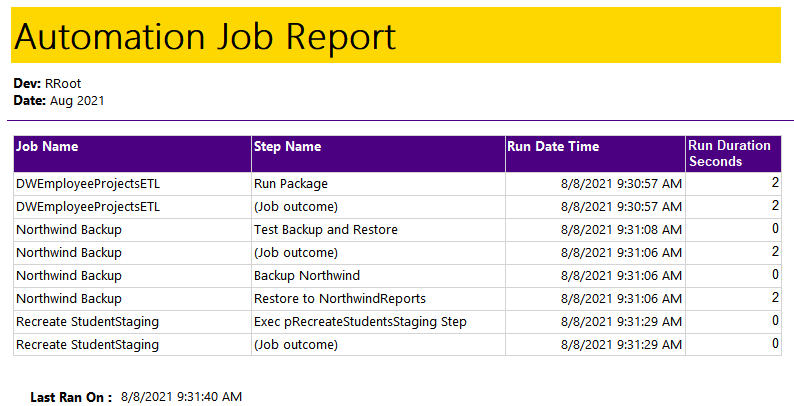
From msdb.dbo.sysjobs as j

Inner Join msdb.dbo.sysjobhistory as h

ON j.job\_id = h.job\_id

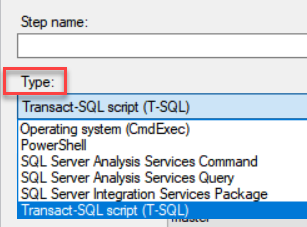
Where j.enabled = 1 --Only Enabled Jobs

Order by JobName, RunDateTime desc



## Proxies

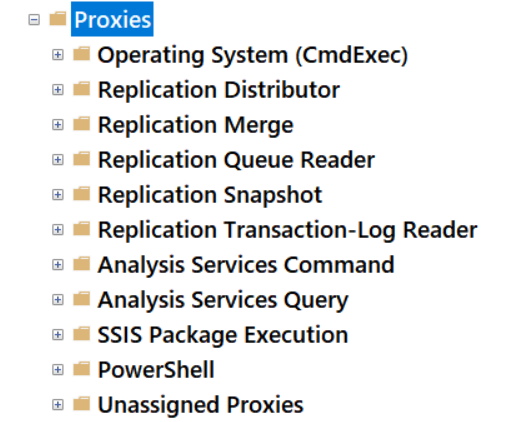
SQL Server Agent has several other types of tasks besides Transact-SQL script tasks, but using them often requires a SQL Agent Proxy.



"SQL Server Agent uses **proxies to manage security contexts**. A proxy can be used in more than one job step. Members of the sysadmin fixed server role can create proxies.

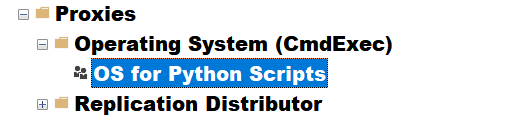
Each proxy corresponds to a security credential. Each proxy **can be associated with a set of subsystems and a set of logins**. The proxy can be used only for job steps that use a subsystem associated with the proxy. To create a job step that uses a specific proxy, the job owner must either use a login associated with that proxy or be a member of a role with unrestricted access to proxies. Members of the sysadmin fixed server role have unrestricted access to proxies. Members of SQLAgentUserRole, SQLAgentReaderRole, or SQLAgentOperatorRole can only use proxies to which they have been granted specific access. Each user that is a member of any of these SQL Server Agent fixed database roles must be granted access to specific proxies so that the user can create job steps that use those proxies."

(<https://docs.microsoft.com/en-us/sql/ssms/agent/sql-server-agent>, 2017)



## Demo 6: SQL Server Agent Proxy

Let's learnhow to create a SQL Server Agent Proxy.



USE [master]

GO

CREATE LOGIN [RRLAPTOP1\Admin]

FROM WINDOWS

WITH DEFAULT\_DATABASE=[master], DEFAULT\_LANGUAGE=[us\_english]

GO

ALTER SERVER ROLE [sysadmin] ADD MEMBER [RRLAPTOP1\Admin]

GO

CREATE CREDENTIAL [Credential for Python Scripts]

WITH IDENTITY = N'RSLaptop\Admin'

GO

EXEC msdb.dbo.sp\_add\_proxy

@proxy\_name=N'OS for Python Scripts'

,@credential\_name=N'Credential for Python Scripts'

,@enabled=1

GO

EXEC msdb.dbo.sp\_grant\_proxy\_to\_subsystem

@proxy\_name=N'OS for Python Scripts'

,@subsystem\_id=3

GO

EXEC msdb.dbo.sp\_grant\_proxy\_to\_subsystem

@proxy\_name=N'OS for Python Scripts'

,@subsystem\_id=12

GO

## Integration Services (SSIS) Tasks

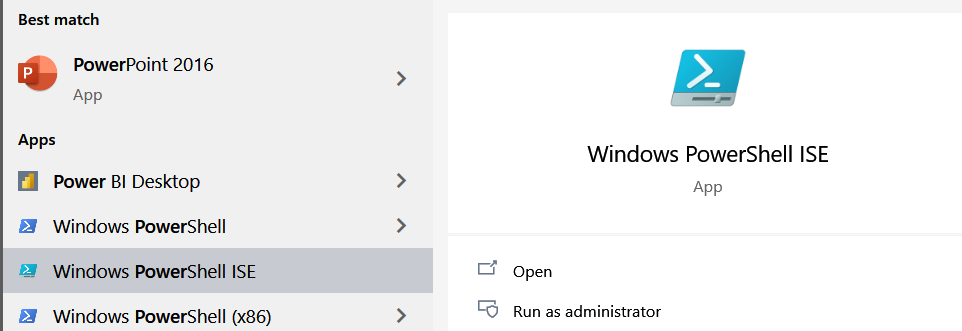
You can automate running SSIS packages from SQL Server Agent.

## **Demo 7:** SQL Server Integration Services Package Task

Let's learnhow to create and automate an SSIS package.

## PowerShell Tasks

"Windows PowerShell is a Windows **command-line shell** **designed especially for system administrators**. Windows PowerShell includes an interactive prompt and a scripting environment that can be used independently or in combination.



Unlike most shells, which accept and return text, Windows PowerShell is **built on top of the .NET Framework** common language runtime (CLR) and the .NET Framework, and accepts and returns .NET Framework objects. This fundamental change in the environment brings entirely new tools and methods to the management and configuration of Windows." (<https://docs.microsoft.com/en-us/powershell/scripting/getting-started/getting-started-with-windows-powershell?view=powershell-6>, 2017)

## Demo 8: PowerShell Task

Now, your instructor will **demonstrate** how to create a PowerShell script using the following script.

# Step 1) Create a new SQL Script

# Use the following Code to create a test script

If(Test-Path C:\BackupFiles) {

Write-Host "Folder Ready"

}

Else {

MD C:\BackupFiles

}

Backup-SqlDatabase -ServerInstance localhost `

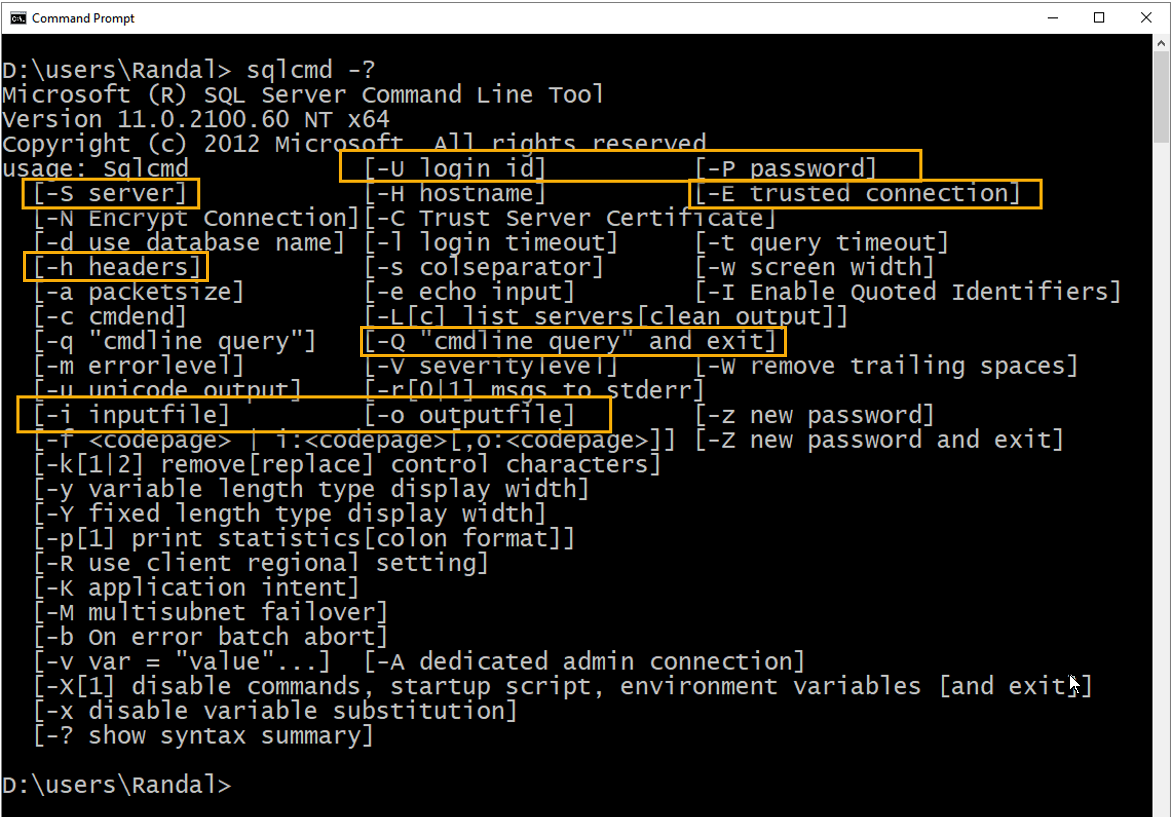
-Database Northwind -BackupFile C:\BackupFiles\NW.bak -Initialize

## Operating System (CmdExec) Tasks

The most powerful task allows you to **run any program** as you would from a command shell. This means you can run something like **SQL Server applications like Bulk Copy Program or SQLCmd**, as well **as custom applications** built with C#, Python, Java, and more.

### Using SQL Server SqlCmd

"The sqlcmd utility is a **command-line utility** for ad hoc, interactive execution of Transact-SQL statements and scripts and **for automating Transact-SQL scripting tasks**. To use sqlcmd interactively, or to build script files to be run using sqlcmd, users must understand Transact-SQL." (<https://docs.microsoft.com/en-us/sql/relational-databases/scripting/sqlcmd-use-the-utility>, 2017)



For a better description of the options use this webpage: <https://docs.microsoft.com/en-us/sql/tools/sqlcmd-utility?view=sql-server-ver15>

### Using Bulk Copy Program (BCP)

The BCP utility can both **EXPORT** and IMPORT data from data files and query results. Here is an example of how you would import data **from a table to a file**:

**NOTE:** These programs are installed separately from Azure Data Studio or SQL Server Management Studio**.**

**For Windows and Linux:** <https://docs.microsoft.com/en-us/sql/tools/bcp-utility?view=sql-server-2017>

**For Mac:** <https://cloudblogs.microsoft.com/sqlserver/2017/04/03/sql-server-command-line-tools-for-mac-preview-now-available/>

## Demo 9: Operating System Tasks

Let's learnhow to create an Operating System Task that uses SQLCmd.

SQLCmd **-S** localhost **-E -o** "C:\\_SQL330\Demo09-JobHistory.xml" **-Q** "Exec TempDB.dbo.pSelXMLJobHistories" **-h** -1



# Maintenance Plans

"Maintenance plans **create a workflow of the tasks** required to make sure that your database is optimized, regularly backed up, and free of inconsistencies. The Maintenance Plan Wizard also creates core maintenance plans, but creating plans manually gives you much more flexibility.

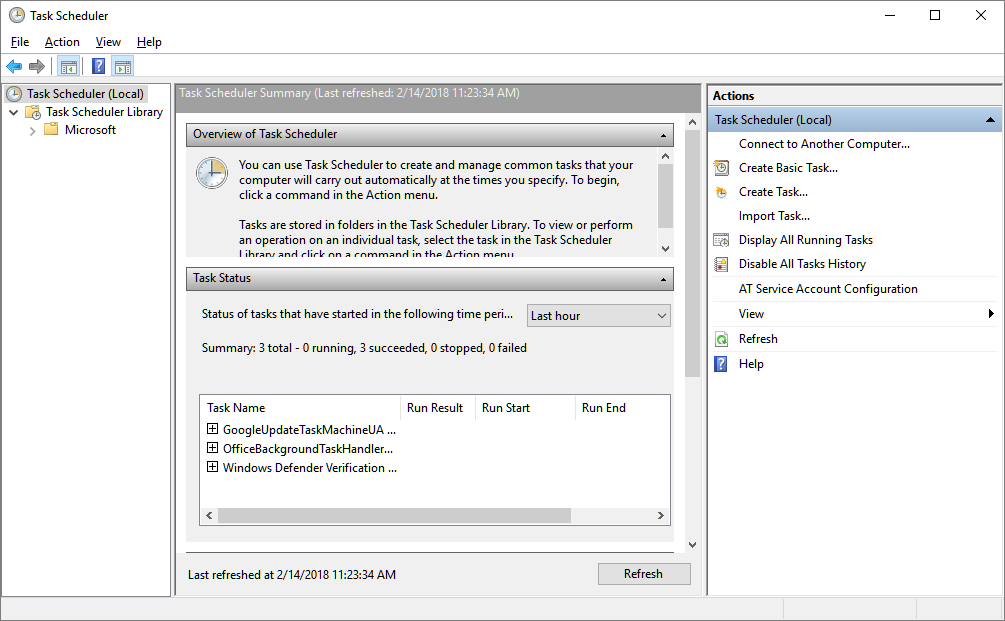
..., maintenance plans create **an Integration Services package**, which is **run by a SQL Server Agent job**. Maintenance plans can be run manually or automatically at scheduled intervals." (<https://docs.microsoft.com/en-us/sql/relational-databases/maintenance-plans/maintenance-plans>, 2017)

## Demo 10: SQL Server Maintenance Plan

Let's seehow to create a SQL Server Maintenance Plan.

# Windows Task Scheduler

"The Task Scheduler enables you to **automatically perform routine tasks** on a chosen computer. The Task Scheduler does this by monitoring whatever criteria you choose to initiate the tasks (referred to as triggers) and then executing the tasks when the criteria is met." (<https://msdn.microsoft.com/en-us/library/windows/desktop/aa383614(v=vs.85).aspx>, 2017)



## Demo 11: Using Windows Scheduler

Let's learn how to create a scheduled task in Task Scheduler.

# 