Contents

[1. Overview 2](#_Toc114216154)

[1.2 Why we configure email configuration 2](#_Toc114216155)

[2. Steps to Configure Email Notification on Failure 3](#_Toc114216156)

[Step 1. 3](#_Toc114216157)

[Step 2. 4](#_Toc114216158)

[Step 3. 4](#_Toc114216159)

[Step 4. 5](#_Toc114216160)

[Step 5. 6](#_Toc114216161)

[Step 6. 6](#_Toc114216162)

[Step 7. Write click on the configured SQL job and go to properties and select the email profile from the drop-down and click ok 7](#_Toc114216163)

[Step 8. 7](#_Toc114216164)

[We can do this also by automatically executing the below script. 8](#_Toc114216165)

**Email Notification on SQL job Succeeds or Failure**

# 1. Overview

Here we are enabling a SQL agent service that will send the email about the status of the job. And we can easily get the information about the job. The purpose to configure is to intimate the user to take care of failed jobs. We will cover how to configure Database Mail, Email profile, SMTP Configuration, Operator to send email, notification, and SQL Server Agent configuration.

We know to perform a series of actions automatically on the SQL server side, we need to configure jobs. A task that is configured will be run one or more times and monitored for success or failure. Some examples of SQL jobs are daily, weekly, monthly and quarterly reports, sending auto emails, newsletters, writing log files, and audit logs to be sent automatically to local or remote locations.

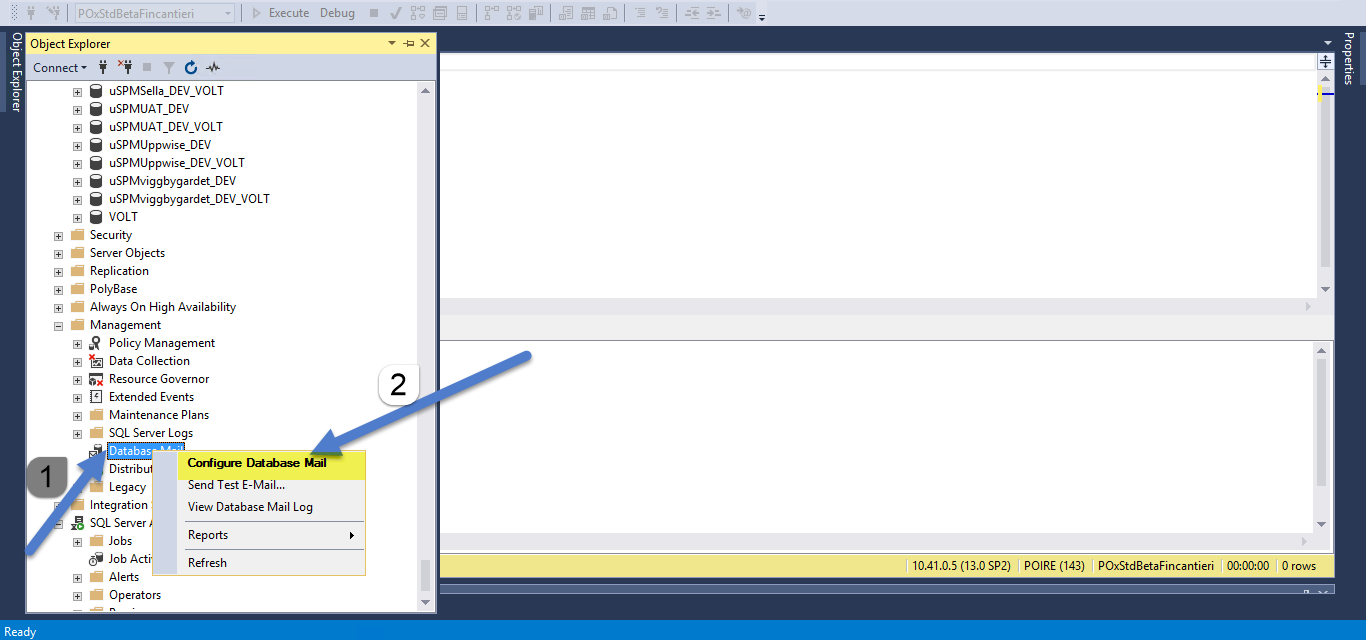
# 1.2 Why we configure email configuration

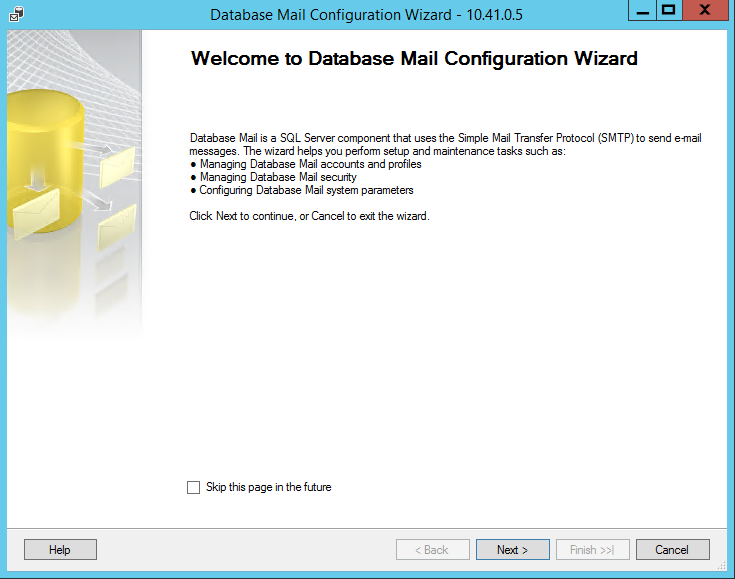
When our job runs, we can face any issue while executing the script. Such issues occurred when working with files, simultaneously data insertion, working with SSDT (SQL Server Data Tools), and many more scenarios. SQL job get stuck and takes more time than expected and finally fails. On job failure, we need to know on an immediate basis to get rid of the problem.

# 2. Steps to Configure Email Notification on Failure

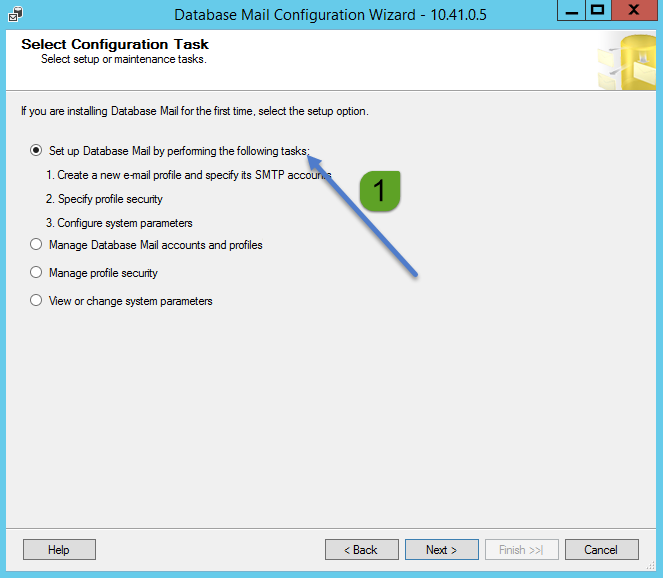
## Step 1.

Under Management, Configure Database Main. Right-click on Database Mail and select the first option (Configure Database Mail)







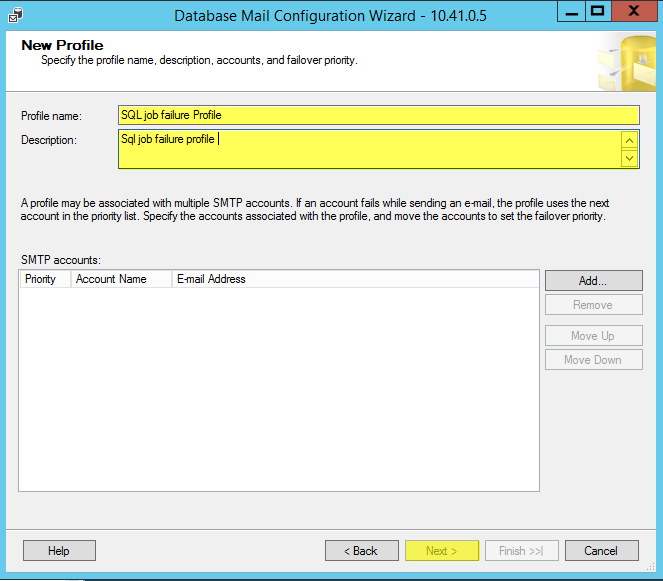


**2**

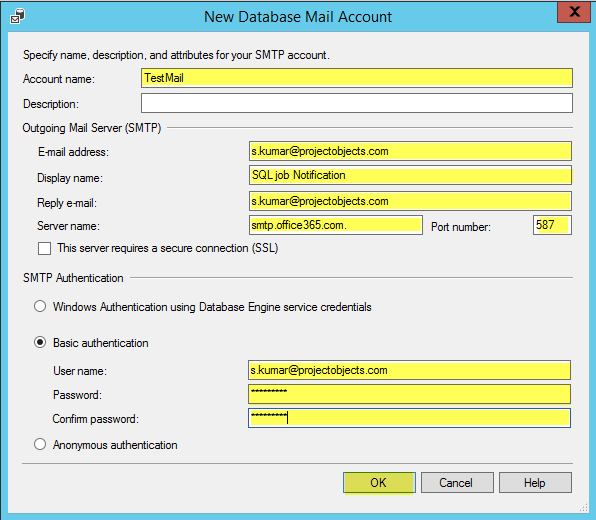


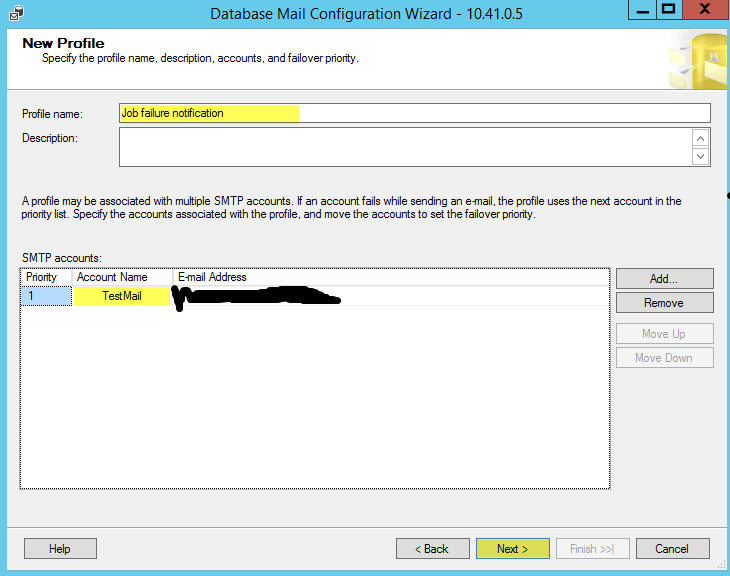
## Step 2.

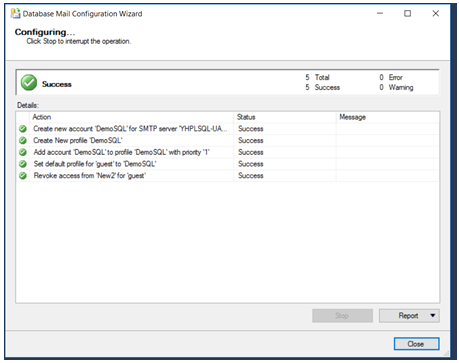
We are creating a new profile, so select the first option and click on Next.



Step 3. Write the profile name and description then click on Add button to add SMTP settings.







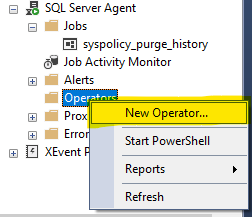
Step 4.

Configure your system level parameters like retry mechanism, attached file size, and allowed/prohibited extensions. Click on Next. The next screen shows all the details of your account simply select Finish. Your all configuration is successfully built.

## Step 5.

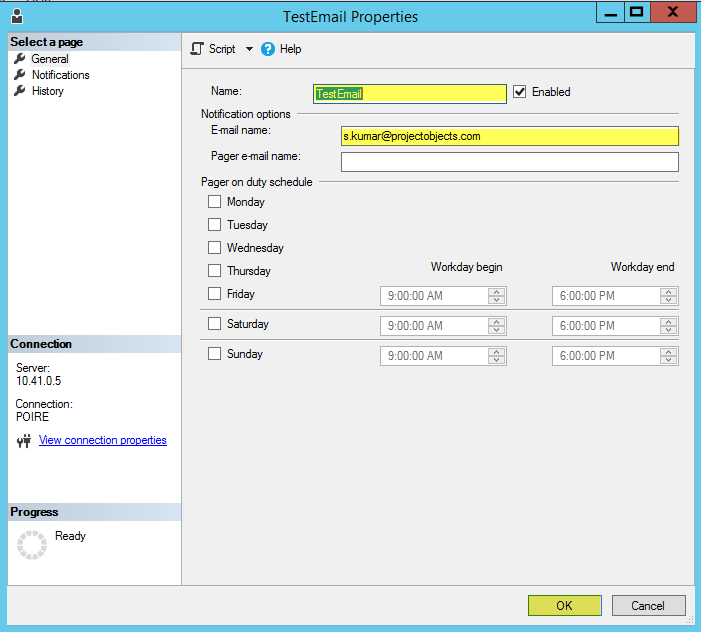
Create an Operator, now we have to make an operator to whom you want to send an email when the SQL job is failed, Expand the SQL Server agent

Right-click on Operators and select new operator

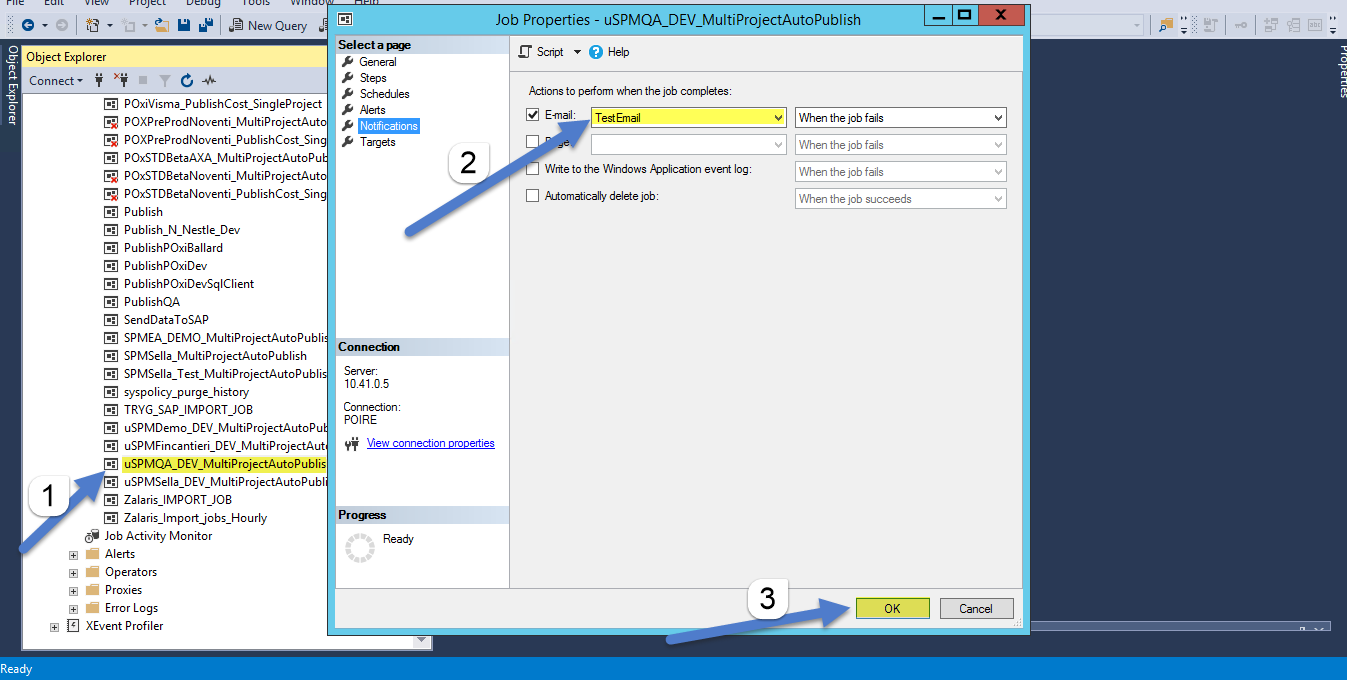


Step 6.

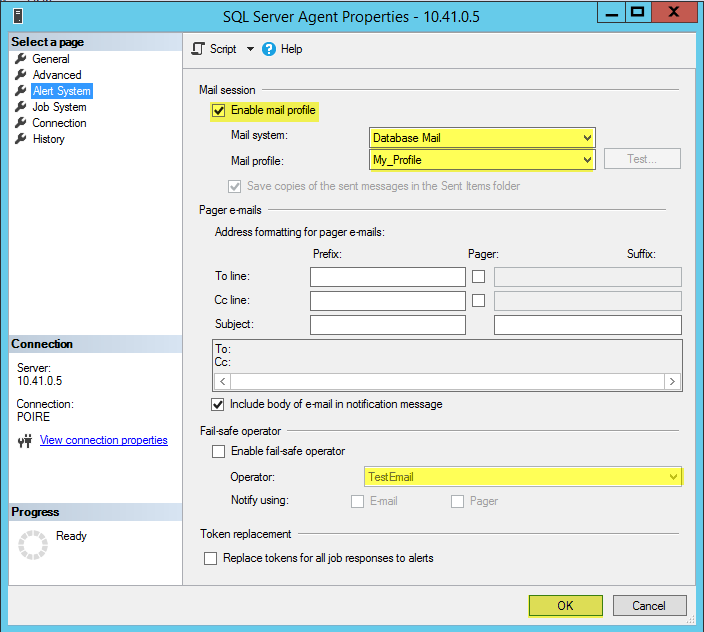
In the following screen enter your Operator name and E-mail name to whom you want to send emails. If you want to send an email to more than one person just separate the email address with ';'.



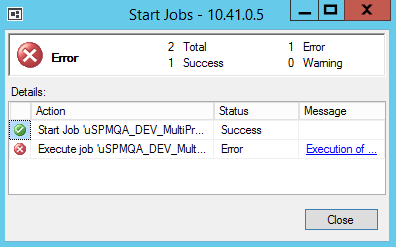
## Step 7. Write click on the configured SQL job and go to properties and select the email profile from the drop-down and click ok



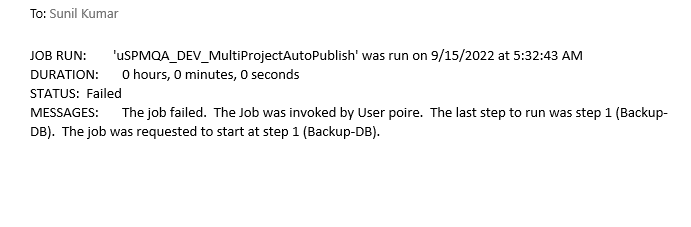
Step 8. You have to Enable the SQL agent alert system. Right-click on the SQL server agent and go to the properties then select which alert profile you want to enable it.



When the job is manually executed or on the scheduled time and failed then you as seen in the below snapshot. And you have got the email on that email id that you have entered in the operator.



After the SQL job fails, we have got this type of email



# We can do this also by automatically executing the below script.

DECLARE @YourEmail NVARCHAR(50) SET @YourEmail = 's.kumar@projectobjects.com' --Put here your E-mail

DECLARE @YourPassword NVARCHAR(50) SET @YourPassword = 'XYZ' --Put here your Password

DECLARE @YourSMTPSserver NVARCHAR(50) SET @YourSMTPSserver = 'smtp.office365.com' --Put here your SMTP Server

DECLARE @YourPort int SET @YourPort = '587' --Put here your SMTP port

-- Create a Database Mail account

EXECUTE msdb.dbo.sysmail\_add\_account\_sp

@account\_name = 'My\_Account',

@description = 'Mail account for you.',

@email\_address = @YourEmail,

@replyto\_address = @YourEmail,

@display\_name = 'Your Mailer Account',

@mailserver\_name = @YourSMTPSserver,

@port = @YourPort,

@use\_default\_credentials = 0,

@username = @YourEmail,

@password = @YourPassword;

-- Create a Database Mail profile

EXECUTE msdb.dbo.sysmail\_add\_profile\_sp

@profile\_name = 'My\_Profile',

@description = 'Your profile used for your e-mail' ;

-- Add the account to the profile

EXECUTE msdb.dbo.sysmail\_add\_profileaccount\_sp

@profile\_name = 'My\_Profile',

@account\_name = 'My\_Account',

@sequence\_number =1 ;

-- Grant access to the profile to the DBMailUsers role

EXECUTE msdb.dbo.sysmail\_add\_principalprofile\_sp

@profile\_name = 'My\_Profile',

@principal\_name = 'public',

@is\_default = 0;

EXEC msdb.dbo.sysmail\_help\_account\_sp;

-- show advanced options

EXEC sp\_configure 'show advanced options', 1

GO

RECONFIGURE

GO

-- enable Database Mail XPs

EXEC sp\_configure 'Database Mail XPs', 1

GO

RECONFIGURE

GO

-- check if it has been changed

EXEC sp\_configure 'Database Mail XPs'

GO

-- hide advanced options

EXEC sp\_configure 'show advanced options', 0

GO

RECONFIGURE

GO