# Bulk Update Database Mail Password

Database mail is a critical part of getting notifications from your SQL Server when things go wrong. The problem comes when you have a database mail SMTP account where every 3 months the password expires and needs to be reset. Given you have many SQL Servers that are all using that same SMTP account to send email you may have to go to each SQL Server and step through the process to change the SMTP password test the email and confirm the email. For one environment where I needed to do this on just over 20 SQL Servers that took just over an hour to update the password.

In a perfect world you could get the SMTP account set to never expire, but that is not always an option depending on your environment, team and skillset.

I am someone who hates to waste time, but the way to do this through the SSMS user interface is the following:

- Expand the object explorer and expand the Management item.
- Right click on the Database Mail item and choose Configure Database Mail.
- Then choose Manage Database Mail accounts and profiles, and click Next.
- Next select "view, change or delete an existing account", and click next.
- Enter the new SMTP password twice, then click next.
- Then click finish.
- Next right click on the database mail again to test the email to confirm the change worked.

## But Wait... Is there an easier way?

First if you are using Registered Servers and you have the ability to run the same query against multiple SQL Servers, this can get really quick, if not you just have to copy and paste to run the same query on each SQL Server via SQL Server Management Studio, that looks something like this.

```
DECLARE @updateSQL AS NVARCHAR(MAX);
DECLARE @emailAccount AS NVARCHAR(1000);
DECLARE @password AS NVARCHAR(1000);
DECLARE @testEmailDestination AS NVARCHAR(1000);
DECLARE @mailProfileName as NVARCHAR(1000);



SET @password = 'putYourPasswordHere';
SET @emailAccount = 'NotificationEmailAccount@someDomain.com';
SET @testEmailDestination = 'WhoToSendTestEmaiTo@someDomain.com';
SET @mailProfileName = 'SQL Alerts';
SET @updateSQL = ''; -- initialize in the case that nothing is found in the select statement



SELECT @updateSQL = 'EXECUTE msdb.dbo.sysmail_update_account_sp
@account_name = ''' + a.[name] + '''
,@description = ''' + a.description + '''
,@mailserver_name = ''' + s.[servername] + '''
,@mailserver_type = ''' + s.[servertype] + '''
,@password = ''' + @password + '''
,@username = ''' + @emailAccount + '''
,@port = ' + cast(s.[Port] as varchar(100)) + ';'
FROM [msdb].[dbo].[sysmail_server] as s
INNER JOIN [msdb].[dbo].[sysmail_account] as a ON s.[account_id] = a.[account_id]
WHERE email_address = @emailAccount;



EXECUTE sp_executesql @stmt = @updateSQL;



EXEC msdb.dbo.sp_send_dbmail
@profile_name = @mailProfileName ,
@recipients = @testEmailDestination,
@body = 'Test Email',
@subject = @@SERVERNAME ;



SELECT s.[account_id]
,a.[name] AS [Account Name]
,s.[servertype]
```

```
,s.[servername] AS [SMTP Server Address]
,s.[Port]
,a.description
, *
FROM [msdb].[dbo].[sysmail_server] as s
INNER JOIN [msdb].[dbo].[sysmail_account] as a ON s.[account_id] = a.[account_id]
and username = @emailAccount;
```

After running that script, the email account will not only be updated, but you will receive an email from each SQL Server confirming it worked.

This saves me about an hour compared to 2 minutes every 3 months.

I hope you enjoy this query, and if you need any help, please reach out to me at Stedman Solutions, we can help with your SQL Server needs.

-Steve