
Send Email from SQL Server Express Using a CLR Stored Procedure

Written By: Edgewood Solutions Engineers -- 7/16/2009

Problem

One of the nice things about SQL Server is the ability to send email using T-SQL. The downside is that this functionality does not exist in SQL Server Express. In this tip I will show you how to build a basic CLR stored procedure to send email messages from SQL Server Express, although this same technique could be used for any version of SQL Server.

Solution

If you have not yet built a CLR stored procedure, please refer to this tip for what needs to be done for the initial setup.

[CLR Functions - Getting started with a string sort function](#)

In this CLR stored procedure we are going to pass in a few parameters such as:

- recipients - list of people to receive the email
- subject - subject line of the email
- from - who the email is from
- body - the body of the email

This is just the basic information to send an email. You could add as many parameters as you want to extend the functionality, but I kept it simple to illustrate how easy this is to do.

The code in this tip was taken from [Sending HTML formatted email using SSIS Script Task](#).

Step 1 - CLR code

The first thing we need to do is to write the CLR code. This could be written in either C#.NET or VB.NET. In this example we are using VB.NET.

The following code has a Class (StoredProcedure) and a Stored Procedure (spSendMail). The stored procedure takes four parameters:

- recipients - list of people to receive the email
- subject - subject line of the email
- from - who the email is from
- body - the body of the email

Before you save the code you will need to change these two lines to include the data for your SMTP server and your account you will use to send emails:

- `mySmtpClient = New SmtpClient("mail.yourservername.com")` -- this should be your SMTP server
- `mySmtpClient.Credentials = New NetworkCredential("email@domain", "yourPassword")` -- this should be the email address and password to authenticate in to your email server

After you have made these two adjustments save the code below in a file called: `C:\SendEmail.vb`

```
Imports System.Net
Imports System.Net.Mail

Public Class StoredProcedure
    <Microsoft.SqlServer.Server.SqlProcedure()> _
    Public Shared Sub spSendMail(ByVal recipients As String, ByVal subject As String, ByVal from As String, ByVal body As String)
        Dim mySmtpClient As SmtpClient

        Using myMessage As New MailMessage(from, recipients)

            myMessage.Subject = subject
            myMessage.Body = body

            myMessage.IsBodyHtml = True

            mySmtpClient = New SmtpClient("mail.yourservername.com")
            mySmtpClient.Credentials = New NetworkCredential("email@domain", "yourPassword")
            mySmtpClient.Send(myMessage)

        End Using
    End Sub
End Class
```

```
End Sub
End Class
```

Step 2 - Compile CLR Code

In order to use this code, the code has to be compiled.

The following command is run from a command line to compile the CLR code using the vbc.exe application. This is found in the .NET 2.0 framework directory. This may be different on your server or desktop. Also, this code should be compiled on the machine where the code will run.

So from a command line run a command such as the following:

```
C:\WINDOWS\Microsoft.NET\Framework\v2.0.50727\vbc /target:library C:\SendEmail.vb
```

The code should now be compiled in a file called: C:\SendEmail.dll

Step 3 - Create Assembly and Stored Procedure

After the code has been compiled you need to create the assembly and the stored procedure within SQL Server. To do this, run these commands in the database where you want to use the function. For this example I am using the msdb database.

The assembly ties an internal object to the external DLL that was created and the stored procedure is similar to a normal SQL Server stored procedure.

For the stored procedure you will see the components that are referenced [SendEmail].[SendEmail.StoredProcedure].[spSendMail]

```
USE msdb
GO

CREATE ASSEMBLY SendEmail FROM 'C:\SendEmail.dll'
WITH PERMISSION_SET = UNSAFE
GO

CREATE PROCEDURE [dbo].[spSendMail]
    @recipients [nvarchar](4000),
    @subject [nvarchar](4000),
    @from [nvarchar](4000),
    @body [nvarchar](4000)
WITH EXECUTE AS CALLER
AS
EXTERNAL NAME [SendEmail].[SendEmail.StoredProcedure].[spSendMail]
```

If you get error messages when trying to compile the code you may need to alter the database using the following command and then try again to create the assembly and the stored procedure.

```
ALTER DATABASE msdb SET trustworthy ON
```

Step 4 - Test It

To test the stored procedure, run the following statement based on the values you want to pass.

In this example we are sending an email to "tips@mssqltips.com".

```
EXEC spSendMail @recipients = 'tips@mssqltips.com', @subject = 'Email from SQL Express',
@from = 'tips@mssqltips.com', @body = 'This is a test email from SQL Server'
```

In this example we are sending an email to "tips@mssqltips.com" and "joe@edgewoodsolutions.com".

```
EXEC spSendMail @recipients = 'tips@mssqltips.com, joe@edgewoodsolutions.com',
@subject = 'Email two from SQL Express',
@from = 'tips@mssqltips.com', @body = 'This is a test email from SQL Server'
```

Step 5 - Cleanup

If you want to remove these objects follow these steps.

- Delete the DLL that was created
- Delete the VB file that was created

16/07/2009

Send Email from SQL Server Express...

- Run these T-SQL statements

```
USE msdb
GO
DROP PROCEDURE dbo.spSendMail
GO
DROP ASSEMBLY SendEmail
GO
```

Summary

That's all there is to creating a CLR stored procedure to send email. This was a basic example and only offered a few options, but you can easily extend this to add other features.

Next Steps

- If you do not have a way to send email from SQL Server give this CLR process a try
- If you are not running SQL Express take a look at these tips for setting up Database Mail which may be an easier way to go
 - [SQL Server 2005 Database Mail Install Hangs](#)
 - [SQL Server 2005 Database Mail Setup and Configuration Scripts](#)
 - [Database Mail and SQL Agent Mail setup by using a script](#)

Copyright (c) 2006-2009 [Edgewood Solutions, LLC](#) All rights reserved
[privacy statement](#) | [disclaimer](#) | [copyright](#)
Some names and products listed are the registered trademarks of their respective owners.