2018

Configuring DB MAIL

CREATE DB MAIL ACCOUNT & DB MAIL PROFILE

KAUSHIK AWALA, RAKSHIT VALLABHANENI

UNIVERSITY OF HOUSTON | UNDERGRADUATE STUDENT SUCCESS CENTER

Create a Database Mail Profile

Use either the Database Mail Configuration Wizard or Transact-SQL to create Database Mail public and private profiles. For more information about mail profiles see Database Mail Profile

Before You Begin

Prerequisites

Create one or more Database Mail accounts for the profile. For more information about creating Database Mail accounts, see Create a Database Mail Account.

Security

A public profile allows any user with access to the **msdb** database to send e-mail using that profile. A private profile can be used by a user or by a role. Granting roles access to profiles creates a more easily maintained architecture. To send mail you must be a member of the **DatabaseMailUserRole** in the **msdb** database, and have access to at least one Database Mail profile.

Permissions

The user creating the profiles accounts and executing stored procedures should be a member of the sysadmin fixed server role.

Using Database Mail Configuration Wizard

To Create a Database Mail profile

- In Object Explorer, connect to the SQL Server instance you want to configure Database Mail on, and expand the server tree.
- Expand the **Management** node
- Double click Database Mail to open the Database Mail Configuration Wizard.
- On the **Select Configuration Task** page, select **Manage Database Mail accounts and profiles** option and click **Next**.
- On the **Manage Profiles and Accounts** page, select **Create a new profile** option, and click **Next**.
- On the **New Profile** page, specify the Profile name, Description and add accounts to be included in the profile, and click **Next**.
- On the **Complete the Wizard** page, review the actions to be performed and click **Finish** to complete creating the new profile.

To configure a Database Mail private profile:

- Open the Database Mail Configuration Wizard.
- On the **Select Configuration Task** page, select **Manage Database Mail accounts and profiles** option, and click **Next**.
- On the **Manage Profiles and Accounts** page, select **Manage profile security** option and click **Next**.

- In the **Private Profiles** tab, select the check box for the profile you would like to configure and click **Next**.
- On the **Complete the Wizard** page, review the actions to be performed and click **Finish** to complete configuring the profile.

To configure a Database Mail public profile:

- Open the Database Mail Configuration Wizard.
- On the **Select Configuration Task** page, select **Manage Database Mail accounts and profiles** option, and click **Next**.
- On the **Manage Profiles and Accounts** page, select **Manage profile security** option and click **Next**.
- In the **Public Profiles** tab, select the check box for the profile you would like to configure and click **Next**.
- On the **Complete the Wizard** page, review the actions to be performed and click **Finish** to complete configuring the profile.

Call a stored procedure with parameters in C#

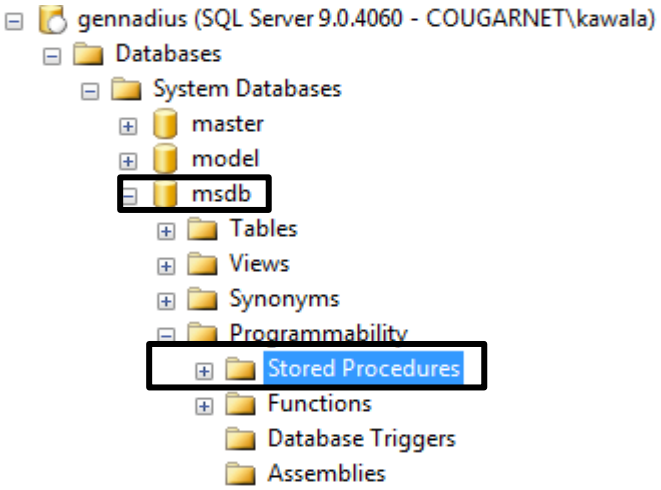
1. Calling a stored procedure is pretty much the same as running a query. Also, use using for all disposable objects, so that you are sure that they are disposed properly.
2. Note:
3. cmd. Parameters. Add (String parameterName, Object value) is deprecated now. Instead use cmd. **Parameters. AddWithValue (String parameterName, Object value)**
4. You may use the below code snippet as a reference to call a stored procedure from within a C# program file.

```
using (SqlConnection conMail = new
SqlConnection(ConfigurationManager.ConnectionStrings["sendMail"].ConnectionString))
{
  using(SqlCommand cmdMail = new SqlCommand("sp_send_dbmail", conMail))
  {
    cmdMail.CommandType = CommandType.StoredProcedure;

    cmdMail.Parameters.AddWithValue("@profile_name", SqlDbType.VarChar).Value =
"ussc";
    cmdMail.Parameters.AddWithValue("@recipients", SqlDbType.VarChar).Value =
emailString;
    cmdMail.Parameters.AddWithValue("@body", SqlDbType.VarChar).Value = body;
    cmdMail.Parameters.AddWithValue("@body_format", SqlDbType.VarChar).Value = "HTML";
    cmdMail.Parameters.AddWithValue("@Subject",
    SqlDbType.VarChar).Value = getAppConfigValue("subjectOfEmail");
    conMail.Open();
    cmdMail.ExecuteNonQuery();
  }
}
```

Important Notes:

1. The e-mail needs to be sent from a stored procedure **sp_send_dbmail** which is a part of the system database **msdb.** Hence, you need to specify the fully qualified name of the stored procedure. In the above case, please specify "**msdb.dbo.sp_send_dbmail**".
2. Use the connection string for the **msdb database** which is present under **system databases**.
3. Add the connectionString details in the web config file.



```
<add name="sendMail" connectionString=" User ID=DBUserID; Password=DBPassword;
Data Source=DB-IP-Address; Initial Catalog=msdb; Enlist=false;"
providerName="System.Data.SqlClient"/>
```

4. To be able to use the sp_send_dbmail stored procedure, you need to have a DB Mail Account and a DB Mail Profile.
5. Firstly, create a DB Mail Account.
6. Secondly, create a DB Mail Profile. Refer to Create a Database Mail Profile.docx documentation to create a DB Mail account and a profile.
7. Finally, configure the profile security for the profiles created to be able to send emails.
   Gennadius → Management → Database Mail
   - 7.1.1. Right-Click on Database Mail to configure database mail.
   - 7.1.2. Click Next
   - 7.1.3. Select Manage profile security
   - 7.1.4. Click Next
   - 7.1.5. Under public profiles tab, select the profile that you have created. To note, a public profile can be accessed by all users of any mail-host database.
     - 7.1.5.1. In case you would like to configure the profile security to private, click on private profiles tab, select the username from the dropdown menu and select the profiles that username should have access to. To note, a private profile can only be accessed by a specific user of a mail-host database.