**Configuring and setting up SQL database mail**

Database Mail is an enterprise solution for sending e-mail messages from the SQL Server Database Engine. Using Database Mail, your database applications can send e-mail messages to users. Database Mail is not active by default. To use Database Mail, you must explicitly enable Database Mail by using either the Database Mail Configuration Wizard or sp\_configure stored procedure. Once it has been enabled, you can configure SQL mail using either GUI (Wizard) or by script.

After the Account and the Profile are created successfully, we need to configure the Database Mail. To configure it, we need to enable the Database Mail XPs parameter through the sp\_configure stored procedure, as shown here:

sp\_CONFIGURE -- shows the options for server properties (about 17)  
GO

sp\_CONFIGURE 'Show Advanced', 1 -- shows the options for server properties (about 69)  
GO  
RECONFIGURE  
GO

sp\_CONFIGURE 'Database Mail XPs',1 -- configure the database mail  
GO  
RECONFIGURE  
GO

--------------------------------------------------  
sp\_CONFIGURE 'Show Advanced', 0 -- set the options off for server properties  
GO  
RECONFIGURE  
GO

**Setting up SQL database mail via wizard:**

Database Mail Configuration through scripts  
When you need to setup Database Mail on dozens of SQL Server instances, rather than perform this tedious task using the SSMS GUI, use the following script that saves me a lot of time. Below is the template. The sysmail\_add\_account\_sp @username and @password parameters might be required depending on your SMTP server authentication and you will of course need to customize the mail server name and addresses for your environment.

-- 1. Enable Database Mail for this instance  
EXECUTE sp\_configure 'show advanced', 1;  
RECONFIGURE;  
EXECUTE sp\_configure 'Database Mail XPs',1;  
RECONFIGURE;  
GO

-- 2. Create a Database Mail account  
EXECUTE msdb.dbo.sysmail\_add\_account\_sp  
@account\_name = 'Primary Account',  
@description = 'Account used by all mail profiles.',  
@email\_address = 'myaddress@mydomain.com', -- enter your email address here  
@replyto\_address = 'myaddress@mydomain.com', -- enter your email address here  
@display\_name = 'Database Mail',  
@mailserver\_name = 'mail.mydomain.com'; -- enter your server name here

--3. Create a Database Mail profile  
EXECUTE msdb.dbo.sysmail\_add\_profile\_sp  
@profile\_name = 'Default Public Profile',  
@description = 'Default public profile for all users';

-- 4.Add the account to the profile  
EXECUTE msdb.dbo.sysmail\_add\_profileaccount\_sp  
@profile\_name = 'Default Public Profile',  
@account\_name = 'Primary Account',  
@sequence\_number = 1;

-- 5.Grant access to the profile to all msdb database users  
EXECUTE msdb.dbo.sysmail\_add\_principalprofile\_sp  
@profile\_name = 'Default Public Profile',  
@principal\_name = 'public',  
@is\_default = 1;  
GO

--6.send a test email  
EXECUTE msdb.dbo.sp\_send\_dbmail  
@subject = 'Test Database Mail Message',  
@recipients = 'testaddress@mydomain.com', -- enter your email address here  
@query = 'SELECT @@SERVERNAME';

GO

Database Mail keeps copies of outgoing e-mail messages and other information about mail and displays them in msdb database using the following scripts:  
use msdb  
go  
SELECT \* FROM sysmail\_server  
SELECT \* FROM sysmail\_allitems  
SELECT \* FROM sysmail\_sentitems  
SELECT \* FROM sysmail\_unsentitems  
SELECT \* FROM sysmail\_faileditems  
SELECT \* FROM sysmail\_mailitems  
SELECT \* FROM sysmail\_log