Database Maintenance Guide

Medtech Evolution

Document Version 5

Last Modified on: February 26th 2015

(February 2015)

This documentation contains important information for all Medtech Evolution users and IT Support Personnel. We suggest filing it safely for future reference.
Table of Contents

Chapter 1 - Introduction ............................................................................................................. 4

Chapter 2 - Backup Database .................................................................................................... 4
  Pre-requisite ............................................................................................................................. 4
  Space in the hard drive ............................................................................................................. 4
  Active Transaction .................................................................................................................... 4
  Types of Backup ....................................................................................................................... 4
    Full Backup ............................................................................................................................ 5
    Differential Backup ................................................................................................................. 5
    Transactional Backup ............................................................................................................. 5
  Backup using SSMS .................................................................................................................. 5
  Backup using SQL Script ......................................................................................................... 8
    Full Backup ............................................................................................................................ 8
    Differential Backup .............................................................................................................. 8
    Transaction Log Backup ....................................................................................................... 8

Chapter 3 - Restore Database .................................................................................................. 9
  Pre-requisite ............................................................................................................................. 9
  Space in the hard drive ............................................................................................................. 9
  Administrative Access ............................................................................................................ 9
  Restore using SSMS ................................................................................................................ 9
  Restore using SQL Script ....................................................................................................... 10
  Restoration Process for Log Backups ..................................................................................... 11
  Backup and Restore SQL Server Database to a network shared drive .................................. 11
  Backup and Restore from Network shared drive ................................................................... 13
  Medtech Recommendations for DB Backups ....................................................................... 14

Chapter 4: Attach Databases .................................................................................................. 15
  Attach Database using SSMS ................................................................................................. 15
  Attach Database using SQL Script ......................................................................................... 16

Chapter 5: Database Maintenance .......................................................................................... 17
  Maintenance Plan Configuration for Standard/Enterprise Edition ........................................ 17
  Database Mail Configuration .................................................................................................. 17
  Operator Configuration .......................................................................................................... 18
  Maintenance Tasks .................................................................................................................. 18
    Backup DB (Full) .................................................................................................................... 20
    Backup DB (Log) .................................................................................................................... 21
    Execute SQL Server Agent Job ............................................................................................ 21
  Database Integrity ................................................................................................................... 22
  Clean-up History .................................................................................................................... 22
  Backup DB (Differential) ....................................................................................................... 23
  Maintenance Clean-up Task .................................................................................................... 23
  Report Generation .................................................................................................................. 24
  Verification and Complete ..................................................................................................... 24
  Maintenance Plan Schedule ................................................................................................. 25
  Maintenance Plan Configuration for Express Edition ......................................................... 25
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintenance Task Sequence</td>
<td>26</td>
</tr>
<tr>
<td>Execute Maintenance Task Scripts</td>
<td>26</td>
</tr>
<tr>
<td>TSQL Scripts</td>
<td>26</td>
</tr>
<tr>
<td>Batch File</td>
<td>27</td>
</tr>
<tr>
<td>Windows Task Scheduler</td>
<td>27</td>
</tr>
</tbody>
</table>
Chapter 1 - Introduction

This document covers the below sections which must be handled in the SQL server environment for Evolution database:

- Database Backup
- Database Restore
- Database Attach
- Database Maintenance Plan - Standard and Enterprise
- Database Maintenance Plan - Express

This document is applicable for all editions of SQL server (Express/Enterprise/Standard).

Refer the Evolution Software Requirement Specification document for the supported Versions.

Note: This guide does not include the Database Maintenance steps for the Evolution Database configured on High Availability using the SQL Server 2012 (SQL Always on Availability Groups).

Chapter 2 - Backup Database

Evolution database backup is essential for protecting critical information and responding effectively to a disaster. The following sections outlines topics that helps you achieve this purpose.

Pre-requisite

Space in the hard drive

Before you implement a backup and restore strategy, you should estimate how much disk space a full database backup will use.

Note: The backup operation copies the data in the database to the backup file. The backup contains only the actual data in the database and not any unused space. Therefore, the backup is usually smaller than the database itself. You can estimate the size of a full database backup by using the sp_spaceused system stored procedure.

Active Transaction

Need to check if there are any active transactions happening in the database by using the statement sp_who2 which will give you the list of active users in the SQL server. You will still be able to take backup with those active users in the SQL server, however it is good to check with the active users and take the backup.

Types of Backup

There are 3 different types of backup strategy available, and the best can be implemented based on the business usage.
Full Backup

A full database backup backs up the whole database. These backups create a complete backup of your database as well as part of the transaction log, so the database can be recovered. This allows for the simplest form of database restoration, since all of the contents are contained in one backup.

Recommendation: This is recommended if the Evolution database size is minimal.

Differential Backup

A differential backup is a backup of any extent that has changed since the last FULL backup was created. A differential backup captures only the data that has changed since that full backup.

Recommendation: This is recommended if the Evolution database size is huge.

Transactional Backup

A transaction log backup allows you to back-up the active part of the transaction log. So after you issue a “Full” or “Differential” backup the transaction log backup will have any transactions that were created after those other backups are completed.

Recommendation: This is recommended only if there are numerous transactions in the Evolution database.

Backup using SSMS

1. Go to All Programs → Microsoft SQL Server xxxx → SQL Server Management Studio.
2. Connect using the login credentials (Admin user) and go to the database which you want to backup from the SQL server.
3. Right click the Database → Tasks → Backup and you will get the backup database screen.
4. Check the required options in the backup database screen:
   a. **Database**: Select the database that you are going to take backup.
   b. **Recovery model**: The full recovery model is the most complete recovery model and allows you to recover all of your data to any point in time as long as all backup files are useable.
   c. **Backup Type**: Select **Full/Differential/Transaction** in the drop down list based on the business requirement.
   d. **Copy-only Backup**: This option should be left unchecked.
   e. **Backup component**: The option **Database** should be selected.
   f. **Backup Set**: This section can be left unaltered with whatever values populated by default.
   g. **Destination**: Select the option **Disk** and modify the backup location.
h. **Options**: Let the **Option** tab with the default pre-populated value as below:

5. Click **OK** to start backup.
**Backup using SQL Script**

Below SQL script will perform full backup of the database to the specified location:

**Full Backup:**

```sql
USE DBName;
GO
BACKUP DATABASE DBName
TO DISK = 'G:\Backup\Evolution_25Aug2014.bak' -- Backup file
location with the backup file name
WITH FORMAT,
NAME = 'DB Name'; -- This name will be displayed while
restoring the DB in same/different server
GO
```

**Differential Backup:**

```sql
USE DBName;
GO
BACKUP DATABASE DBName TO DISK = 'G:\Backup\Evolution_25Aug2014.DIF' WITH DIFFERENTIAL
GO
```

**Transaction Log Backup:**

```sql
USE DBName;
GO
BACKUP LOG DBName TO DISK = 'G:\LogBackup\Evolution_25Aug2014.TRN'
GO
```
Chapter 3 - Restore Database

Pre-requisite

Space in the hard drive

Before starting the restore, check whether the hard drive has adequate space (where the log files are to be stored).

Administrative Access

The SQL user should have db_owner access to perform restore operation.

Restore using SSMS

1. Go to All Programs → Microsoft SQL Server xxxx → SQL Server Management Studio.
2. Connect using the login credentials and expand the server.
3. Right click the database and click Restore Database option.

4. Provide the below information in the Restore Database screen:
   a. To database : Database name to be restored
   b. To a point time: Most recent possible
   c. Source for restore : Select the option From device and provide the backup file location.
   d. Click the Options menu in the left panel, provide separate hard drive location for the transaction log files, i.e., if the file type is ‘Log’ then provide separate hard drive for that file and it would be preferable to have SSD hard drive for the transactional log file.
   e. Rename\Append the SECONDARY and SECONDARYBLOB MDF files with '_SECONDARY' and '_SECONDARYBLOB' so that there is no duplication of MDF files.
5. Click **OK** to Restore.

### Restore using SQL Script

SQL database can be restored using the below SQL scripts:

1. Execute the below SQL script which will return a result set containing a list of the database and log files contained in the backup set in SQL Server.
   
   ```sql
   RESTORE FILELISTONLY
   FROM DISK = 'G:\Backup\Evolution_25Aug2014.bak' -- Backup file location
   ```

   Change the backup file location to the corresponding path where the backup file is located in your hard drive. You will get a list of logical names as below based on the number of files it has:

<table>
<thead>
<tr>
<th>Logical Name</th>
<th>Physical Name</th>
<th>Type</th>
<th>FileName</th>
<th>Size</th>
<th>MaxSize</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEST</td>
<td>B:\Program Files\Microsoft SQL Server\MSSQL10_50\MSSQL\DATA\Evolution_25Aug2014.mdf</td>
<td>PRIMARY</td>
<td>NULL</td>
<td>164976</td>
<td>291000</td>
</tr>
<tr>
<td>TEST_log</td>
<td>B:\Program Files\Microsoft SQL Server\MSSQL10_50\MSSQL\DATA\Evolution_25Aug2014_log.ldf</td>
<td>L</td>
<td>NULL</td>
<td>198576</td>
<td>265000</td>
</tr>
</tbody>
</table>

2. The logical name in the above result set should be replaced in the below SQL script:

   ```sql
   RESTORE DATABASE TEST
   FROM DISK = N'G:\Backup\Evolution_25Aug2014.bak' -- Backup file location
   WITH
   MOVE 'Test' to 'D:\Database\Evolution_25Aug2014.mdf',
   MOVE 'TEST_log' to
   'L:\DatabaseLog\Evolution_25Aug2014_log.ldf',
   NORECOVERY, FILE =1
   ```

   In the above SQL script the values `TEST` and `TEST_log` are derived from the logical name in the previous SQL script. The number of log files may vary based on the type of the database and additional line needs to be added based on the result from the previous script.

   The location in the below line should be the backup file location:

   ```sql
   FROM DISK = 'G:\Backup\Evolution_25Aug2014.bak' -- Backup file location
   ```

   The location in the below lines can be any of your hard drive where you wish to save the log files, refer **Evolution Database Server Configuration Guide** for drive guidelines:
MOVE 'Test' to 'D:\Database\Evolution_25Aug2014.mdf',
MOVE 'TEST_log' to 'L:\DatabaseLog\Evolution_25Aug2014_log.ldf'

**Restoration Process for Log Backups**

The restoration of differential and transactional log files needs to be performed in the below order:

1. If the differential backup file needs to be restored, then a most recent full database backup need to be restored upon which the most recent differential backup needs to restored.
2. If the transactional backup file needs to be restored, then a most recent full database backup needs to be restored, upon which the most recent differential backup needs to be restored, upon which all the transactional backup file needs to be restored in sequential time order from the time differential backup is taken till the most recent transactional backup file.

**Backup and Restore SQL Server Database to a network shared drive**

*Note:* It is recommended to backup and restore SQL Server database on a local drive as it may be slow on a shared network drive.

The following steps will enable or disable XP_CMDSHELL using the SP_CONFIGURE system stored procedure. In order to use the XP_CMDSHELL you need to be a system administrator. Using the XP_CMDSHELL command we will do the Backup and Restore SQL server database to a network shared drive.

**Enable XP_CMDSHELL using SP_CONFIGURE**

A database administrator can enable XP_CMDSHELL using SP_CONFIGURE system stored procedure. To enable XP_CMDSHELL you need to execute the below mentioned script:

```
USE Master
GO
EXEC master.dbo.sp_configure 'show advanced options', 1
RECONFIGURE
GO
EXEC master.dbo.sp_configure 'xp_cmdshell', 1
RECONFIGURE
GO
```

**Output:**

- Configuration option 'Show Advanced Options' is changed from 0 to 1.
- Run the RECONFIGURE statement to install.
- Configuration option 'xp_cmdshell' is changed from 0 to 1.
- Run the RECONFIGURE statement to install.

**Disable XP_CMDSHELL using SP_CONFIGURE**

It is always considered as a best practice to disable XP_CMDSHELL when not in use. A database administrator can disable XP_CMDSHELL using SP_CONFIGURE system stored procedure. To disable XP_CMDSHELL you need to execute the below mentioned script:

```
USE Master
GO
```
EXEC master.dbo.sp_configure 'xp_cmdshell', 0
RECONFIGURE
GO
EXEC master.dbo.sp_configure 'show advanced options', 0
RECONFIGURE
GO

Output:
- Configuration option 'xp_cmdshell' is changed from 1 to 0.
- Run the RECONFIGURE statement to install.
- Configuration option 'show advanced options' changed from 1 to 0.
- Run the RECONFIGURE statement to install.

Enabling Network drive:
For a network share to be visible on the SQL Server, the share drive must be mapped as a network drive in the session in which SQL Server is running.

Example:
EXEC xp_cmdshell 'NET USE z: \inlogicdbs4\EVOQADBTempBkup'
Once the above command is executed then we need to provide permission to access the network mapped drive for the SQL Server service account (example: Domain\Username). You will now be able to view the mapping of the drive across SQL Server.

**Backup and Restore from Network shared drive**

You need to execute the below script to enable that the mapping drive is visible in the backup and restore wizard.

```sql
-- allow changes to advanced options
EXEC sp_configure 'show advanced options', 1
GO
-- Update currently configured values for advanced options.
RECONFIGURE
GO
-- To enable xp_cmdshell
EXEC sp_configure 'xp_cmdshell', 1
GO
-- Update currently configured values for advanced options.
RECONFIGURE
GO
EXEC xp_cmdshell 'NET USE z: \inlogicdbs4\EVOQADBTempBkup'
```

This happens before execution:
Medtech Recommendations for DB Backups

The purpose of creating SQL Server backups is to enable you to recover a damaged database. Below are the Medtech recommendations for backup and restore process.

It is recommended:

- To schedule full database backup based on the business requirement. Medtech recommends taking FULL backup at least once in a week.
- To restore the transaction log files on the SSD hard drive than normal hard drive for the effective performance.
- To have adequate space in the hard drive where the transaction log files are stored so that there is no hassle during the transaction in the database.

Backup must be taken during non-working hours.
Chapter 4: Attach Databases

Attach database can be used in the following scenario:

a. If you need to move a database or database file to another server or disk, you can perform attach operation with the available log files.

b. If the backup strategy is FILE/SAN based, you can copy the database files to another server, and then re-create the database from the backup using attach operation.

Medtech would recommend performing SQL restore than attach operation in any situation. When there is a situation where the database is corrupted or database is not available due to disaster, it is ideal to perform attach process with the available log files.

The user performing this operation should have Windows SQL server authentication, and the user should have full access to the server (sysadmin).

Attach Database using SSMS

1. Go to All Programs → Microsoft SQL Server xxxx → SQL Server Management Studio.
2. Connect using the login credentials and expand the server.
3. Right click the database and click Attach.
4. Click the Add button in the Database to Attach screen and provide the mdf file location which needs to be attached to the SQL server.
   a. Databases to attach: Displays information about the selected databases.
   b. MDF File Location: Displays the path and file name of the selected MDF file.
   c. Database Name: Displays the name of the database.
   d. Attach As: Optionally, specifies a different name for the database
   e. Owner: Provides a drop-down list of possible database owners from which you can optionally select a different owner.
   f. Status: Displays the status of the database
   g. Message: Displays either a blank message or a “File not found” hyperlink.
   h. Original File Name: Displays the name of the attached file belonging to the database.
i. **File Type:** Indicates the type of file, Data or Log.

j. **Current File Path:** Displays the path to the selected database file. The path can be edited manually.

k. **Message:** Displays either a blank message or a "File not found" hyperlink.

l. When you are ready to attach the database, click **OK**.

---

**Attach Database using SQL Script**

Before proceeding to attach database, move the MDF and LDF files to be restored to a specific location. Once done, execute the below SQL script by changing the database name and location of the log files (highlighted):

```sql
USE [master]
GO
CREATE DATABASE [DatabaseName] ON
    -- MDF file location
    (FILENAME = N'D:\Database\Evolution_25Aug2014.mdf'),

    -- LDF file location
    (FILENAME = L:\Database\Evolution_25Aug2014_log.ldf)  
FOR ATTACH
GO
```
Chapter 5: Database Maintenance

Maintenance task setup is mainly to reduce the manual intervention in managing the SQL server by setting up certain configuration which will handle the clean-up activity in the SQL server. Setting this configuration is one time activity after which the maintenance task like backup, integrity, etc., will be taken care of by the SQL server based on the configuration.

There are several maintenance tasks discussed in the following sections out of which the intended area can be applied to the server based on the business requirement.

Maintenance Plan Configuration for Standard/Enterprise Edition

Maintenance plan will be configured using the SQL server management studio (SSMS). Connect to the SSMS and provide the login credentials:

![Connect to Server](image)

Database Mail Configuration

This mail configuration is for database and hardware and job failed email notification, configure one or more email ids with “;”.

- SMTP Access rights/setting required for the Server
- SMTP Details required for Configuration
  - Server Name or IP Address
  - Password for Server
  - Port Number
  - Username (email)
  - Email Password
- Test SMTP details using EmailTestApp.exe application

Database Mail Configuration Wizard

- From Object Explorer, connect to an instance of SQL Server.
- Expand Management, right-click Database Mail, and select Configure Database Mail.
- Choose the Set up Database Mail option to set up Database Mail for the first time.

Choose one of the other options for specific maintenance tasks:
- Manage Database Mail accounts and profiles.
- Manage profile security.
- View or change system parameters.
Operator Configuration

Operator configuration is to perform mail group configuration.

- Enable Mail System and Mail Profile & Enable Fail Safe operator with Email Notify
- Restart Sql server and Agent
- Create Alert & Configure Alert with Operator Name (execute Script)
- System Purge job fail notification configure


Maintenance Plan

Once login, expand the Server→Management→Maintenance Plans→Maintenance Plan Wizard:

This will take you to the SQL server maintenance plan wizard where you can set the maintenance task in a step by step process.
There are two ways of scheduling a task in SQL server:

- Create single schedule for the entire task which will be performed in the order defined in the ‘Select Maintenance Task’ order wizard (discussed below)
- Create separate schedules for each task

When you select **Single schedule for the entire plan** or **No schedule radio** button, the schedule box below the radio button will get activated and you can schedule the time for the task(s) to be executed in the SQL server. Click the **Change** button and configure the schedule task time and the occurrence proportion.

If the task needs to be performed separately at a different time frame, select **separate schedules for each task** radio button so that you can schedule task at different times than executing them at a single instance.

Select **Maintenance task** option and click **Next** to proceed with the configuration of maintenance plan settings. The subsequent steps in the maintenance plan setup are identical which will be explained in the following sections.
Maintenance Tasks

Medtech recommends the below tasks to be scheduled in the periodic interval for better performance of SQL server:

<table>
<thead>
<tr>
<th>Maintenance Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup Database (Full)</td>
</tr>
<tr>
<td>Re-organize Index (Custom)</td>
</tr>
<tr>
<td>Rebuilt Index (Custom)</td>
</tr>
<tr>
<td>Update Statistics (Custom)</td>
</tr>
<tr>
<td>Check Database Integrity</td>
</tr>
<tr>
<td>Backup Database (Differential)</td>
</tr>
<tr>
<td>Backup Database (Transaction Log)</td>
</tr>
<tr>
<td>Clean-up History</td>
</tr>
<tr>
<td>Maintenance Cleanup Task</td>
</tr>
</tbody>
</table>

Select the tasks in the **Select Maintenance Plan** Wizard.

Click **Next** to select the order of execution of maintenance plan:
Use Move Up or Move Down button to give the order of execution of the maintenance plan.

Backup DB (Full)
This task lets you to automate the full backup of the database in regular basis:
- Choose the database
- Change the Backup file path

Backup DB (Log)
This task let you to automate the transaction log backup of the database in regular basis:
- Chose the database
- Change the Backup file path

Execute SQL Server Agent Job
SQL Server Agent is a Microsoft Windows service that runs jobs that have been defined in an instance of SQL Server. This wizard will show the list of jobs that are available in the SQL server which you can select to execute at the scheduled time.
Configure “IndexOptimize - USER DATABASES” job.

Database Integrity
Database integrity task checks the allocation and structural integrity of all the objects in the specified database for the selected list of databases.

Include Indexes option enables to check the integrity of all the index pages as well as the table data pages.

Clean-up History
By using the History Cleanup task, a package can delete historical data related to backup and restore activities, SQL Server Agent jobs, and database maintenance plans.
Backup DB (Differential)

This task enables you to automate the differential backup of the database in a regular basis:

Maintenance Clean-up Task

The Maintenance Clean Up task removes files related to maintenance plans, including database backup files and reports created by maintenance plans. Maintenance Clean Up task can be configured to delete files that have reached a specified age. For example, the task can be configured to delete files that are older than four weeks. If you do not specify the minimum age of files to delete, all files of the specified type are deleted.
Report Generation

This option will get you the report in a file saved in the specified location. You can select the check box to send e-mail when a task fails. To use this task you must have Database Mail enabled and correctly configured with MSDB as a Mail Host Database, and have a Microsoft SQL Server Agent operator with a valid e-mail address.

Verification and Complete

You can verify the choices made in the maintenance plan wizard like the plans you have selected, location of the report etc., in this wizard.
Once you verify the report, click Finish to complete. With this the setting up of maintenance plan is complete and it will be executed automatically on the scheduled time.

**Maintenance Plan Schedule**

Schedule the Maintenance plan configuration.
- Enable Schedule
- Choose Occurs frequency
- Choose Time Frequency

**Maintenance Plan Configuration for Express Edition**

SQL server Express Edition is a freeware which doesn’t have all the functionalities as in the Standard Edition. One such functionality that is missing in the Express Edition is the Maintenance Tasks. However, we have created some batch process which in turn invokes certain SQL scripts to perform maintenance task in the SQL server Express Edition.

To perform the maintenance task in the SQL server express edition, we need to create windows scheduler task to execute a batch file which in turn invokes some TSQL script
to perform the maintenance tasks. We have detailed the process and the method of execution in the following sections of this document.

- **Daily Task**
  - Backup Database (Full)
  - Backup Database (Transaction Log)
  - Index Tasks
  - Check Database Integrity

- **Daily (Hours) Task**
  - Backup Database (Diff)
  - Backup Database (Transaction Log)

Transaction log is applicable, if Database recovery model is FULL.

### Maintenance Task Sequence

There is some sequence of steps which should be followed to achieve the maintenance task in SQL server Express Edition.

- **TSQL Script** - There are some TSQL scripts for each process (Backup, Index, etc.,) which will perform the maintenance task against the database.
- **Batch Process File** - A Batch file is available which will invoke all the TSQL scripts and execute in order against the database.
- **Windows Task Scheduler** - A windows task scheduler is scheduled in the system which will automatically perform the maintenance task using the batch process and TSQL scripts.

The means of achieving the above steps is explained in the remaining sections of the document in detail, but please be noted the document has been created with dynamic value in the batch file which should be replaced with the corresponding database value against which the maintenance task is to be performed.

### Execute Maintenance Task Scripts

- Execute maintenance task script in master database.
- Configure batch file
- Change database name and Directory path (backup file) in SQL file

### TSQL Scripts

This section has the list of TSQL scripts which are used for the database maintenance tasks.

#### DB Backup & Log Backup Script

The script is to take backup for database. In this SQL script, we have provided the default value for @CleanupTime variable as 24 hours which should be modified based on the business requirement. Based on this value, database backup will be deleted automatically from the backup folder.

#### Index Script

The script is to reorganize, rebuild, and update statistics index script is to optimize database performance.

#### Check Database Integrity

The script is to check the database health.

---

© Copyright Medtech Limited
**Batch File**

Batch file is used to execute the above set of SQL scripts in a sequential order. Batch file will be invoked by the windows task scheduler and this batch file will in turn invoke the SQL scripts in order to execute against the provided SQL server.

The below batch file can be opened in the notepad file. Open the attached file in notepad and replace the below with their corresponding value:

- **username** (U) - It should be replaced with the SQL login user name
- **password** (P) – It should be replaced with the SQL login password
- **servername** (S)– It should be replaced with the name of SQL server
- **dbname** (D)– It should be master
- Add `-o<space>"SpecifyLogfile Path"` in the end of the command i.e. `-o<space>"C:\file.txt"`

Modify all these information and save the batch file. Place all the SQL script and the batch file in the same folder without which the batch file can’t invoke the SQL scripts. If you wish not to execute any of the SQL script, you can remove that SQL script from the folder so that it won’t be executed against the database.

Please ensure that the SQL user configured in the batch file has `db_backupoperator` role membership. (To verify this, go to Microsoft SQL Management Studio → Login to the server → Security → Login → Rightclick the userlogin → Properties → UserMapping → check if the `db_backupoperator` role is checked.

**Windows Task Scheduler**

Windows task scheduler schedules the task in the windows system. To schedule a task in the system, go to **Control Panel → System and Security → Administrative Tools → Task Scheduler**.

In the Task Scheduler, open the **Basic task** in the menu list and provide the task information. Schedule whether the task needs to be performed daily/weekly/monthly or so and the time of execution in the consecutive screens. Once you reach the **Start the program** screen, browse the batch file we created and give the batch file path only for **Start in** (optional) text box and complete the schedule task.

Scheduled task can be viewed in the Task Scheduler Library in the windows task scheduler screen. The above content is ensuring that we can perform a db back up and log back up, shrink db, shrink log, reorganize index, rebuild index, update statistics, clean up through task scheduler using above sql files in SQL Server Express Edition.

**Note:** Tick, run where user is logged on or not and run with the highest privileges.

**Reference:** [http://windows.microsoft.com/en-nz/windows/schedule-task#1TC=windows-7](http://windows.microsoft.com/en-nz/windows/schedule-task#1TC=windows-7)
Extended Events for Capturing Performance Metrics

Extended events help you to capture all the supporting information on the Evolution database to troubleshoot performance issues. Follow the below steps to configure the Extended Events:

1. Change the File Storage Path to the respective local path to save the extended events data file. Change the below value to appropriate local path

   filename=N'G:\Medtech Support\XEData\query_performance.xel'

   metadatafile=N'G:\Medtech Support\XEData\query_performance.xem'.

2. Change the Max_Memory value to 1% of SQL Server Max Memory or 1GB. Whichever is lower we need to configure it. (For Example SQL Server Max Memory – 1GB configure Extended Events Max memory as ’1MB’.

   [Image: MedTech_EvolutionXE_Query_Performance]

For further information, or any other queries regarding the changes in this release, please contact Medtech Support as follows:

- Via the Evolution application [Help ► Contact Support]
- By email to support@medtechglobal.com
- By fax to 0800 MEDTECH (633 832)
- By phone on 0800 2 MEDTECH (633 832)