SMR

Sql Metadata Repository

Installation Manual

SMR – Installation Manual

Inhoud

SMR – Installation Manual

1 Introduction

SMR stands for SQL Metadata Repository, see also the User Manual for the terminology used and a detailed description.

2 Preconditions

2.1 System Requirements

- SQL Server 2012 or higher, standard edition
- Non dedicated, can be added to existing installations
- CLR enabled
- xp_cmdshell enabled
- .NET Framework 4.5 installed
- 5GB diskspace (no experience data known but this is already on the large side)
- Agent XPs and SQL Server Database Mail enabled and set up (if opting for overviews by mail)
- Internet access from the database server (if you choose to upload overviews)
- Ad hoc queries must be enabled on the SMR instance

N.B.: system requirements for the SMR instance that are programmatically adjustable are actually set up during the installation, see ***0***

Configuring.

2.2 Authorization

Credentials can be specified per catalog, the relevant user needs the ***db_datareader*** role on the source database to be able to retrieve the metadata.

The credentials are stored with the catalog data in table ***SMR_Data.dbo.Cat***, of course possible Passwords are encrypted. If no credentials are given, then Integrated Security is assumed, which means that the person or service carrying out the process must have authorization.

When the process is executed as a Job - which will be the normal course of events - then with Integrated Security the SQL Agent account must be authorized for this on the source databases..

3 SQL Server

3.1 Installing the databases

Restore the supplied backups to the desired location, do not change the database names.

N.B.: this is a SQL Server 2012 backup, restoring on earlier versions is not possible.

3.2 Configuring

Open a query window on the SQL Server and execute:

```
SMR_Core.dbo.up_AutoConfig
```

This makes sure that:

- database owner of SMR_Core is 'sa';
- database owner of SMR_Data is 'sa';
- XP_CMDSHELL is enabled;
- CLR is enabled;
- Agent XP's are enabled;
- Database Mail XP's are enabled;
- Login SMR_Uploader is created and linked to the database User;
- License is initialized: the ProductCode is generated and shown shown for the purpose of applying for the License ***save this!***;
- The default System parameters are created.

4 Upload tool

The tool ***SMRUploader.exe*** and Dynamic Link Library ***DDCrypt2.dll*** have to be placed together 'somewhere' in one folder on the SMR database server.

The path to the Upload tool is then stored in the Parameter table, along with the credentials for the Web server so that SMR can execute it in the background.

4.1 Installeren

Decide where to put the files and place them there.

4.2 Configureren

Open a query window on the SQL Server and execute:

```
exec SMR_Core.dbo.up_Register_UploadParameters <Parameters>
```

The parameters (comma separated) are:

Parameter	Type	Purpose
<Program>	string	the full path to the tool including the tool itself, eg C:\SMR_Components\SMRUploader.exe
<URL>	string	The internet location for the Uploads
<Login>	string	The login for the internet location
<Password>	string	The password for the internet location

These values must be made available by the internet site administrator.

With this registration it is tested whether the program is executable, we expect -4 as a return value because no further parameters are specified, this is described in the User Manual with the Upload tool.

Output (example):

```
Uploader is found at the specified location: '<Locatie>\SMRUploader.exe'
Uploader can be executed; returned -4 as expected
Uploader Parameters are updated
```

5 License

5.1 Apply

With ***0***

Configuring a Product Code is generated, pass this on to the supplier and indicate for how many databases a license is required. The Registration Code is provided on this basis.

If this product code is not written down, it can be retrieved; open a query window on the SQL Server and execute:

```
select SMR_Core.dbo.exf_Get_ProdCode()
```

5.2 Register

The supplier provides the following information:

- LicenseType
- Validity Date
- RegistrationCode

These can be entered into the system; open a query window on the SQL Server and execute:

```
exec exp_License_Register '<Type>', '<Datum>', '<Code>'
```

6 Remaining parameters

In the steps above SP's have been used to configure certain components, other components can be configured by directly editing the ***SMR_Data.dbo.Parameter*** table.

Some have a default value but there are two that must be set specifically for the environment in which SMR is placed:

Parameter	Purpose
ReportFolder	A (temporary) folder for export files, default C: \ TMP, if it does not exist then SMR does not work. Create the folder or change the setting.
ReportMailAddress	The e-mail address where alerts and possibly. attachments are sent to. This is optional, but if receiving mail is desired it must be set correctly: 1 or more mail addresses separated by semicolons.

The complete set of parameters is described in the User Manual.

7 Catalogs

When everything above is done, the system is ready for use. However, the system does not yet 'know' which catalogs it should check.

Registering catalogs is described in the user manual.

8 SQL Job

It is common to run such processes as a SQL Server Job, advice is to do the same here; create a Job that executes the SP ***up_Check*** in database ***SMR_Core*** every day, this takes care of the entire process from checking up to signaling via mail and / or upload.