# SMR

Sql Metadata Repository

## User Manual

# SMR – User Manual

## Inhoud

# SMR – User Manual

## 1   Introduction

### 1.1   Terms and concepts

| Term / concept | Explanation |
|---|---|
| Source platform | The database platform (brand) for source databases. |
| Catalog | A specific database on a database server. |
| Doelplatform | The database platform for the installation of SMR. |
| Datasource | A SQL Server database server that contains 1 or more Catalogs. |
| DWH | Datawarehouse. |
| Instance | A SQL Server database server installation with its own name and usually several user databases. |
| Metadata | Technical details regarding the form and structure of data. |
| Parameter | Adjustable stored variable to configure the behavior of the system. |
| Repository | Warehouse or storage place for (meta) data, in this case the SMR databases. |
| SP of Stored Procedure | Separately stored piece of T-SQL code that can perform a (part of a) process or operation. |
| T-SQL | Transact SQL, the SQL variant that is used to program for SQL Server. |

N.B.:   The terms data source / catalog instead of server / instance / database were chosen because these are generic terms across multiple DB platforms, the predecessor of this system also included a connector for Oracle sources and this may also be added as an option in a future version.

### 1.2   SMR

SMR stands for SQL Metadata Repository; the metadata of selected databases is kept. Differences are detected and, if desired, signaled.

### 1.3   Source platform / target platform

The only source platform that is supported at the moment is Microsoft SQL Server.
There are, however, plans to expand to other source platforms.

The target platform (where SMR is installed) is also Microsoft SQL Server.

### 1.4   Target group

All companies / organizations that use SQL Server for their core business in general and those who have a DWH and / or ETL connections in particular.

## 2   Case

How SMR originated…

### 2.1   Problem

When running a DWH, changes to source databases almost always have an impact. The degree of impact of course depends on the nature and extent of the changes, but also on the "intelligence" of the DWH, or the extent to which the DWH can handle such changes. This intelligence is often lacking, in which case it is up to the administrators / users to take timely action and adapt the DWH to the changed source database.

In practice, however, one is not always timely aware of any changes and if so, which changes. In that case, the consequence is often that the DWH processes simply "breaks down".

### 2.2   Cause

Sometimes the supplier (external or internal) of the source database informs the administrators about changes, but that is certainly not always the case. The danger here is that the database change has already passed the acceptance test and has been in production for some time without any problems, after which the DWH suddenly "breaks down".

Example: a column definition changes from 50 to 100 characters, only when more than 50 characters are actually used an error occurs in the processes that maintain the data in the DWH because 'something' no longer fits, in many cases the error situation will first occur in production.

To prevent production downtime, it is important that every change is detected and signaled as soon as possible - that is, during or even before testing - so that this can be taken into account and problems can be prevented.

### 2.3   Solution

The solution to the above problem is relatively simple; By keeping a repository of metadata about the relevant source databases, it is possible to periodically (daily?) compare this with the source and signal any changes to it (detect and report).

### 2.4   Functionality

The metadata is stored with a datetime stamp and periodically compared with the current metadata:

1. differences are detected;
2. differences are registered;
3. differences are reported;
4. history of changes is stored for a certain period.

## 3  Parameters

The parameters below can be found in **SMR_Data.dbo.Parameter**, the <mark>yellow marked ones</mark> are maintained by means of an SP and / or described in the installation manual.

The others must be edited manually in the table:

| Parameter | Purpose |
|---|---|
| ChangesHistDays | Number of days that history of changes is kept.<br>Default = 180. |
| CsvDelimiter | Marks the beginning and end of strings, the specified value is used for both the start and end delimiter. Intelligence for open and close quotes is not built in: eg {string} is not possible, this then becomes {string{.<br>Default is **"** (double quote, Excel CSV format). |
| CsvSeparator | Separator between different fields.<br>Default is **;** (semicolon, Excel CSV format). |
| EncryptLogins | 0 = Dont store Logins Encrypted, 1 = Do store Logins Encrypted for increased security.<br>Default = 0. |
| ExportLimit | Max. number of lines that is created per export file, 0 = nr limit.<br>Default is 0. |
| ExportType | CSV of JSON.<br>Default is CSV. |
| IgnoreColId | 0 = do not ignore if Column Id changes, 1 = do ignore (= do not signal, do register) and only signal changes of data type.<br>Default = 0. |
| Language | Language setting for *datetime* presentation and mail texts; NL or EN.<br>Default is NL. |
| LogHistDays | Number of days that Log lines are kept.<br>Default = 90. |
| LogHistRows | Number of Log rows that are kept. If this threshold is exceeded, the oldest 10% will be cleaned.<br>Default is 10.000. |
| ReportAlways | Switch for reporting, even if there is nothing to report.<br>0 = only with details, 1 = always.<br>Default is 0. |
| ReportCur | Switch for whether or not to supply the complete current population.<br>0 = do not supply, 1 = only supply if there are changes, 2 = always.<br>Default is 0. |
| ReportFileSize | Max size in bytes of an attachment, applies **per attachment** so the total of the attachments can be higher. SQL Server default is 1,000,000 bytes but that is pretty small.<br>SMR Default is 5MB (5,242,880 bytes). |
| ReportFolder | See installation manual |
| ReportHst | Switch for whether or not to supply the complete history. 0 = do not supply, 1 = only supply if there are changes, 2 = always.<br>Default is 0. |
| ReportMailAddress | See installation manual |

| Parameter | Purpose |
|---|---|
| ReportMailType | Only applicable if ReportMailAddress is stored.<br>0 = attachments, 1 = a link to the folder holding the files, 2 = just a bare informational mail.<br>Default is 0. |
| SuppressOutput | 1 = suppress output of command shell activities in the background, 0 = do not suppress (for example for debugging).<br>Default is 1. |
| UplLogin<br>UplPassword<br>UplProgram<br>UplURL | Only applicable if uploading of results is desired, these parameters are exclusively set via the procedure described in the installation manual. |

N.B.:  if EncryptLogins is changed when the system is already in use, any logins are unencrypted while it is specified that they are encrypted or vice versa. In both cases they will be unusable. So, if this setting is changed then the logins must be reregistered for both the catalogs and the upload credentials.

## 3.1  Catalog

Add a catalog as follows; open a query window on the SQL Server and execute:

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

The parameters are separated by a comma, only the first 2 are mandatory:

| Parameter | Type | Purpose |
|---|---|---|
| @DtsName | string | datasource (instance name) |
| @CatName | string | catalog (database name) |
| @Description | string | optional: logical description of this catalog |
| @Login | string | optional: Login with authorization to the catalog (db_datareader) |
| @Password | string | optional: Password of the Login, this is always stored encrypted |
| @IsActive | bit | optional: switch to check this catalog (0) or not (1), for example, this can be switched off temporarily.<br>Default is 1. |
| @DaysHist | int | optional: the number of days of history that are saved.<br>Default is 90 days. |
| @Remarks | string | optional: possibly comments on this catalog. |

N.B.:

1. If no Login and Password is specified, Integrated Security is assumed;
2. Do not forget to authorize the relevant Login on the relevant database (db_datareader role);

3. This procedure can also be used to implement changes, in which case only the changed values are specified in addition to the Datasource and Catalog, the rest of the parameters must then be NULL. If a string parameter is specified empty (""), it will be made NULL.

## 3.2 Examples

```
-- New registration:
exec   [dbo].[up_Register_Catalog]
       'SERVER\Instance1'
,      'Database1'
,      'First catalog'
,      'SMR_Reader'
,      'Password'
;
-- Changed password:
exec   [dbo].[up_Register_Catalog]
       'SERVER\Instance1'
,      'Database1'
,      NULL
,      NULL
,      '$trongerPa$$word'
;
```

## 4   Upload tool

For the purpose of a link with, for example, cloud applications, an upload tool is included: **SMRUploader.exe**.

This tool takes care of uploading the results in the background and is dedicated to SMR. A connection is made to the database via an internal account.

Installation and configuration is described in the installation manual.

### 4.1   Details

This section is purely informative, this is not mandatory reading material for normal use.

Only the name of the SMR instance and the name of the file to be uploaded must be specified as parameters, the other parameters are read from the **SMR_Data** database. Normally there is no output but if the value 1 is specified as the 3rd (optional) parameter, the program gives the output below for testing purposes.:

```
InstanceName . . :
FileName . . . : AAAA-1111-BBBB-2222.Col_Chg_20190726_151440.csv
SourceURI. . . : C:\TMP
TargetURI. . . :
Uploading. . . : Done
Verifying. . . : OK

Press a key. .
```

The credentials for the upload location are also stored as parameters in the database, the password is always encrypted but in this case the name also:

| UplLogin | zY4FBSB4PiGTjyzhnx5rkFzzPqR7LzYOVFFidp9Pksw= |
|----------|---------------------------------------------|
| UplPassword | NJNR7o/HG6WfKNel7c4rgNd9Zw9Uzbo4nY+Ce1xwoR23jes4ehQOmfBOgU7Aq0eO |

The program returns the following values:

| 1 | OK |
|----|----|
| 0 | Undefined |
| -1 | Target URI is notrecognized as URI |
| -2 | Upload failed, specific message is output on screen |
| -3 | Verification failed, specific message is output on screen |
| -4 | Too few parameters (2 required) |
| -5 | Other |

## 5   Output

### 5.1   Contents

There are 3 (sub) collections:

1. Changes; all differences between the source and the repository;
2. Current Population; content of the repository;
3. History; current history of changes from the cleaning date (see Parameters) to date.

These are exported as CSV (default) or JSON file, see Parameter **ExportType** at chapter **3 Parameters**.

#### 5.1.1   Changes

| | A | B | C | D | E | F | G | H | I | J | K | L | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | DataSource | CatalogName | ObjectType | SchemaName | ObjectName | ColumnName | ColumnId | TypeDefinition | Col_DtCreated | Col_DtUpdated | Col_DtDeleted | ChangeTypeDescription | Info |
| 2 | SomeInstance | SomeDatabase | View | dbo | TestView1 | Remarks | 3 | [nvarchar](1000) NULL | 29-07-2019 10:19 | 29-07-2019 10:51 | | Modification | Befor |
| 3 | SomeInstance | SomeDatabase | View | dbo | TestView1 | Species | 2 | [sysname] NULL | 29-07-2019 10:19 | 29-07-2019 10:51 | | Modification | Befor |

Zoomed in on Info:

| M |
|---|
| Info |
| Before: Column Id was 2 and Type Defitition was '[nvarchar](max) NULL' |
| Before: Column Id was 3 |

#### 5.1.2   Current Population

| | A | B | C | D | E | F | G | H | I | J |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | DataSourceName | CatalogName | ObjectType | SchemaName | ObjectName | ColumnName | ColumnId | TypeDefinition | Col_DtCreated | Col_DtUpdated |
| 2 | SomeInstance | SomeDatabase | View | dbo | TestView1 | Animal | 1 | [sysname] NULL | 29-07-2019 10:19 | 29-07-2019 10:19 |
| 3 | SomeInstance | SomeDatabase | View | dbo | TestView1 | Remarks | 3 | [nvarchar](1000) NULL | 29-07-2019 10:19 | 29-07-2019 10:51 |
| 4 | SomeInstance | SomeDatabase | View | dbo | TestView1 | Species | 2 | [sysname] NULL | 29-07-2019 10:19 | 29-07-2019 10:51 |

#### 5.1.3   History

History is identical to Changes but with the column **Col_DtDeleted** as an extra.

### 5.2   Signaling

The output is made in the form of one or more CSV files. There are various methods to offer that output to the user:

1. Mail with attachments;
2. Mail without attachments;
3. Upload via internet to a server;
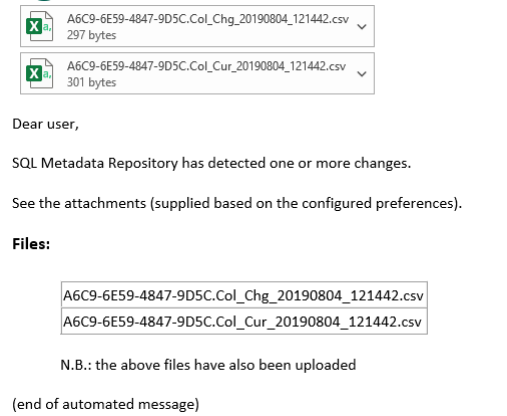4. Combinations of 1 or 2 with 3.

With option 2 only, the user must retrieve the files himself from the relevant folder, it is of course a requirement that this is a shared environment, such that the correct access is arranged for both the SMR server and the user.

See **ReportMailType** Parameter in Chapter **3 Parameters**.
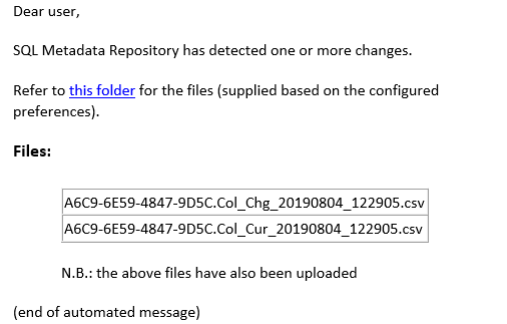
# SMR – User Manual

## 5.3   E-Mail notificaties

With attachments
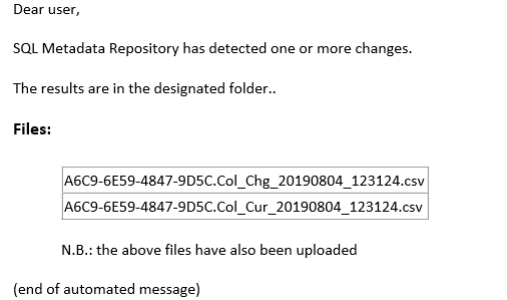Parameter ReportMailType = 0

A6C9-6E59-4847-9D5C.Col_Chg_20190804_121442.csv
297 bytes

A6C9-6E59-4847-9D5C.Col_Cur_20190804_121442.csv
301 bytes

Dear user,

SQL Metadata Repository has detected one or more changes.

See the attachments (supplied based on the configured preferences).

**Files:**

A6C9-6E59-4847-9D5C.Col_Chg_20190804_121442.csv
A6C9-6E59-4847-9D5C.Col_Cur_20190804_121442.csv

N.B.: the above files have also been uploaded

(end of automated message)

Without attachments, with just a link
Parameter ReportMailType = 1

Dear user,

SQL Metadata Repository has detected one or more changes.

Refer to this folder for the files (supplied based on the configured preferences).

**Files:**

A6C9-6E59-4847-9D5C.Col_Chg_20190804_122905.csv
A6C9-6E59-4847-9D5C.Col_Cur_20190804_122905.csv

N.B.: the above files have also been uploaded

(end of automated message)

Without attachments nor link
Parameter ReportMailType = 2

Dear user,

SQL Metadata Repository has detected one or more changes.

The results are in the designated folder..

**Files:**

A6C9-6E59-4847-9D5C.Col_Chg_20190804_123124.csv
A6C9-6E59-4847-9D5C.Col_Cur_20190804_123124.csv

N.B.: the above files have also been uploaded

(end of automated message)

## 5.4   System notifications

If there are notifications from the license check or other processes:

- The number of registered databases exceeds the license;
- The license has expired or will expire soon;
- Error messages or other notifications.

... then they are mentioned in the email. If no mail is configured but only an upload, there is no mechanism to pass these notifications on to the user directly. The important ones are however logged in the Windows event log.

An example of an email including an error message:

Dear user,

SQL Metadata Repository has detected one or more changes.

The results are in the designated folder.

**Files:**

| |
|---|
| A6C9-6E59-4847-9D5C.Col_Chg_20190804_153128.csv |
| A6C9-6E59-4847-9D5C.Col_Cur_20190804_153128.csv |
| A6C9-6E59-4847-9D5C.Col_Hst_20190804_153128.csv |

N.B.: the above files have also been uploaded

**Errors:**

| |
|---|
| Error - The number of registered Catalogs (6) exceeds the LicenseType (5), only the Licensed amount will be handled |

(end of automated message)