

REPORTING WITH MFSQL CONNECTOR

WHITE PAPER USING MFSQL CONNECTOR FOR REPORTING



Published by Laminin Solutions : 1 Jan 2019

Updated 21/3/2019

Table of Contents

Synopsys	2
Background	2
MFSQL Connector Benefit	2
Why choose MFSQL Connector.....	3
Options for reporting in M-Files.....	3
Key considerations	3
Data sources.....	4
Provisioning and Updating.....	6
Data preparation	7
Security considerations.....	8
APPENDIX A – M-FILES REPORTING.....	9

REPORTING WITH MFSQL CONNECTOR

APPENDIX B - MFSQL CONNECTOR (http://tinyurl.com/mfsqlconnector).....	10
Business cases	10
Certification.....	11
Components.....	11

Synopsys

MFSQL Connector is a powerful alternative to M-Files Reporting Tool and other methods to extract metadata from M-Files to generate extracts or pre-formatted reports from M-Files.

The Connector provides the data access layer of the reports. Any Report Designer tool allowing access to data in a database such as excel, Power BI, SQL Report Builder can be used to prepare the presentation layer for the user interface.

Reports can be accessed from within M-Files or through an external interface.

Background

M-Files provides standard functionality and a broad and powerful set of developer-orientated APIs, development tools, frameworks and methods to access the M-Files metadata. Access to the metadata allows the report designer to prepare the data for specific reports.

MFSQL Connector use SQL to extract the data into custom tables for each class table. It also included the ability to extract other types of M-Files data such as object history and the M-Files event log. A range of different methods can be used to trigger the updating of the data into SQL.

This whitepaper elaborates on the considerations for using the MFSQL Connector framework and tools for reporting projects.

Refer to appendix A and B for more information on

MFSQL Connector Benefit

The overarching benefit of using MFSQL Connector is to perform data integration tasks with M-Files without the need to use the M-Files Development Framework (MFDF) and the MFSQL Connector Framework (MFSQL). and program in .NET. The appropriate MFDF functions and methods is wrapped inside

REPORTING WITH MFSQL CONNECTOR

MFSQL to allow a SQL knowledgeable individual to interact with M-Files using T-SQL procedures. The M-Files Development Framework (MFDF) and the MFSQL Connector Framework (MFSQL) were used to develop the framework and enables these procedures.

Why choose MFSQL Connector

- a) MFSQL Connector should be a reporting option if your company have a requirement for reporting using the metadata in the vault.
- b) The Connector provides all the functionality of the M-Files Reporting module and more.
- c) Near realtime reporting. The data can be refreshed on demand and is not subject to the extended processes of refreshing that is often experienced with Reporting Tool
- d) Only changed data is updated. The entire dataset is not refreshed but only the changed data
- e) The SQL tables is easier to work with. The reporting tool requires extensive joins before any data can be extracted. The MFSQL tables include both id's and labels and therefore makes data analysis easier. (Joins could be used where necessary)
- f) Analysis of the vault that is not possible with M-Files Reporting Add-in such as event log and object stats
- g) Different methods can be used to trigger updating the data from fully automated to on demand.
- h) Any standard Report designer can be used to access the data in MS SQL.
- i) Customer have used PowerBI, Visual Studio Report Designer, Interject, excel. and more to generate reports from the data.
- j) It is built with the SQL developer/user in mind
- k) No .NET programming or M-Files API knowledge required

Options for reporting in M-Files

There are different options available to export data from M-Files for reports

- a) Export a view to a CSV file. This method is restricted by the number of records and also can only export one class at a time.
- b) M-Files Reporting module. This method exports a rich set of data to SQL based on a schedule. It also allows for reports to be shown in M-Files.
- c) Data extracts using API's. This method relies on custom development for the specific use case.
- d) MFSQL Connector – data exchange module. This method push and pull data from M-Files on demand, triggered by a variety of methods. It includes exporting of event log not currently available in the reporting module.

Key considerations

The choice of method or framework for reporting from M-Files involves many considerations. The considerations below highlight different aspects to consider. The considerations and weight of each consideration, will be unique for every reporting project.

REPORTING WITH MFSQL CONNECTOR

Any reporting project will involve the following

- a) Identify source of data to be included in the report.
- b) Provision the collection of data to be used.
- c) Updating the latest changes in M-Files into the datasets
- d) Preparing the data for each specific report element
- e) Presenting the data in a report including graphs, visualisation, tabular or other forms of presentation
- f) Distributing the reports to all the relevant users using ODBC, Online Portals, Direct access, Report Servers and other methods of distribution.

The following sections highlight the key aspects in MFSQL connector supporting these considerations. Point e) is not included in the scope of MFSQL Connector and partially supports point f).

Report designers

MFSQL Connector supports any report designer such as PowerBI, Chrystal Reports, Tableau, SQL Report Designer, Excel, Interject and many more that are able to access a datasource in and SQL Table.

The selection of the report designer has many considerations. The following considerations are specifically related the capabilities of MFSQL Connector in relation to M-Files. The following table highlight the considerations for a selection of report designers.

Capability or feature	Power BI	SSRS	Excel	Interject
Connect to MFSQL Table & View	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Use store procedure to produce report	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Update cell change from report to M-Files	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Refresh Data using scheduled refresh (SQL Standard edition only)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Refresh only changed data	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Refresh data using Context menu (MF on premise only)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Refresh data when report is opened	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Dynamic drill down	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Refresh data on demand	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

REPORTING WITH MFSQL CONNECTOR

Refresh data when record changes (only on premise)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Show report from within M-Files (using context menu; on premise only)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Data sources

MFSQL Connector allow access to a large range of meta and other data from M-Files. Special tables in database further enhance the usefulness of the data.

Object Type	Description	Related Tables
M-Files related tables & views		
Object Class	All data defined on the metadata card for a class	Classes included in app defined in Class
Object Version		AuditHistory, ObjectTypeToClassObject
Property history	The change history of any property including workflow states	ObjectChangeHistory
M-Files event log	All events in the log can be exported and included in reports	EventLog_OpenXML, MFilesEvents
Metadata Structure	Object structure and its relationships with other objects	UserAccount, loginAccount, class, Property, objectType, valuelist, Valuelistitems, workflow, workflowState
Connector related tables & views		
Transaction and update logs	Optional tracking of update history and associated process descriptions	Searchlog, UpdateHistory, UserMessages, DeploymentDetail, ProcessBatch, ProcessBatchDetail
Error log	Tracking of system errors	Log
Settings & configuration	Reference tables for some types of reports and analysis	AuthenticationType, ProtocolType, ClassProperty, ContextMenu, DataType, PublicLink, Settings, VaultSettings
File import and export log	Export file history has information about the files of an object.	ExportFileHistory, FileImport

REPORTING WITH MFSQL CONNECTOR

Exclusions:

The following data is not accessible using the Connector:

- Permissions, ACL or related security data
- Metadata related to IML documents

In addition to the data from M-Files, MFSQL Connector allow the incorporation of data from third party systems to be included in the reports.

Note that all MFSQL Standard tables are prefixed with 'MF' in MFSQL Connector database

Provisioning and Updating

Reporting with MFSQL Connector requires the 'Data Exchange module' of the connector.

The Connector have a range of helper procedures to support data exchange.

Update Type	Description	Related Procedures
Update to and from M-Files	Several procedures with a range of options and applications in different scenarios are available	spMFupdateTable spMFupdateTableInBulk spMFupdateTableWithLastModified spMFupdateMFilesToSQL spMFupdateAllTablesIncludedInApp
Setup of reporting	Procedures to setup and create the startup class tables	spMFSetup_Reporting
Get property change history	Procedure to extract the change history of properties	spMFGetHistory
Get Event log	Procedure to Extract M-Files event log	spMFGetMFilesLog
Get Object Version	Procedure to extract object version data only	spMFTableAudit spMFObjectTypeUpdateClassIndex
Update metadata structure	Procedures to update changes in the metadata structure	spMFSynchronizeMetadata spMFSynchronizeSpecificMetadata spMFDropAndUpdateMetadata

The key features and benefits of provisioning data with MFSQL Connector include:

REPORTING WITH MFSQL CONNECTOR

Feature	Description	Options
Update to and from M-Files	Rapid and targeted update of only the records required for reporting Processing large volumes of data take-on	Filter by last modified date in M-Files to only update records changed since last update Filter by object id to process specific records Batch Update to process large volumes of data
Setup of reporting	Rapidly setup SQL to synchronise with M-Files	Auto create class tables, update metadata and setup action menus for selected classes
Get property change history	Get change history for specific properties (such as workflow state)	Filters include: properties, Number of days, start date to only update data needed and optimise time to process
Get all comments	Get all the comments of an object	Use the change history procedure on the comments property
Get Event log	Get all the types of event from the log and allow analysis based on the type of event.	Include data such as user activity; file download history; performance analysis and many more
Get Object Version	Get listing of all the objects and their versions across all object types	Enables analysis and reporting for number of objects by class
Update metadata	Get metadata structure and related objects	Allows analysis of vault, for example the use of specific properties on class tables

Data preparation

A large part of any reporting project is focussed on accessing and categorising the data in a format that suites the analysis and report result.

Reports could be one off analysis or built into the daily operations of the business. MFSQL Connector helper procedures and structure of the tables aids the data discovery and allows for the preparation of custom tables, views and procedures to support reporting. The schema 'custom' is available for these tables, views and procedures to differentiate it from the standard product.

The following special features and aids are available:

Feature	Description	Related
Class table structure	The class table structure is automatically created with features to facilitate reporting, especially to report on class table without extensive joins to get the labels of lookups.	Lookup properties have both ID and label included in the table Multilookup properties show values as delimited list

REPORTING WITH MFSQL CONNECTOR

		Does object have files, and how many?
Special functions	Helper Functions included in the Connector support: Parsing a delimited string, especially to work with multi lookup values Produce hyperlink to the object	fnMFParseDelimitedString fnMFObjectHyperLink
Special views	Special view included in the Connector supports: Analysis of metadata structure View for reviewing log table stats Object stats in vault	MFvwMetadataStructure MFvwLogTableStats, spMFLogTableStats MFvwObjectTypeSummary with spMFObjectTypeUpdateClassIndex
Special procedures	Several helper procedures support the analysis and exploration of data and structure: The use of columns (properties) Show Update History	spMFClassTableColumns spMFUpdateHistoryShow

Security considerations

Securing the data and access to the data in MFSQL Connector is crucial and requires careful consideration depending on the requirements of the company and the types of data that is made available. The following should be considered when designing and implementing the security framework:

Topic	Consideration	Mitigation
M-Files security context	The restrictions to access objects in M-Files is not transferred into MFSQL Connector	Restrict access to the MFSQL Connector database to only users with authorisation to access the data.
Encryption of data in database	MFSQL Connector does not enable encryption of the database by default. Encryption of columns and tables are available from SQL Express 2016 SP 1	Implement encryption in the database if required
Default MFSQL Connector user	The Default MFSQL Connector user 'MFSQLConnect' has a standard password	Change the standard password and manually update the password settings in the vault.

REPORTING WITH MFSQL CONNECTOR

	This User should only be used for external applications including Context Menu.	Do not use or share this user password to everyone in the business.
M-Files user used by MFSQL Connector	The M-Files user assigned to MFSQL Connector has admin rights in the vault.	Restrict the use of this user to MFSQL Connector only The password for this user is saved in encrypted format in the database. Restrict access to the encrypt/decrypt procedures to prevent misuse.
Access to the connectionsettings in the vault	Change the user members of the MFSQLConnector user group in M-Files to be restricted to only admin and the M-Files – MFSQL connector user	
Database Roles in MFSQL Connector database	The MFSQLConnect_db role is created and assigned with appropriate permissions to function as the internal user for third party applications to access the database. From version 4.2.8.46 the installation will create additional roles for MFSQL_Admin and MFSQL_Reporting.	Only use the database role MFSQLConnect_db for the MFSQLConnect user. Assign users to the MFSQL_Admin and MFSQL_Report roles instead of giving access directly to the database.

APPENDIX A – M-FILES REPORTING

M-Files Reporting can improve your management decisions by providing business intelligence on important metrics related to sales processes, completed and in-process projects, the size and state of proposals, order volumes, employee training and certifications, compliance audit findings and deviations, and so much more.

M-Files Reporting utilizes the popular and powerful Microsoft SQL Server Reporting Services report generation system, providing a rich and production-proven backend capable of supporting a wide variety of reporting and business intelligence needs.

With M-Files Reporting, you get:

- Graphic visualizations of M-Files object relationships.

REPORTING WITH MFSQL CONNECTOR

- Business reports highlighting facts and figures from stored data.
- Analytical calculations for decision support.

Ability to display external content in M-Files, such as data from CRM and ERP systems. For example, you could show the latest sales and production figures, in addition to analyzing your document approval processes all in the same interface.

[M-Files Website](#)

[M-Files Catalog](#)

More details on the M-Files reporting tool is available in the M-Files [Knowledgebase](#).

APPENDIX B - MFSQL CONNECTOR (<http://tinyurl.com/mfsqlconnector>)

MFSQL Connector (The Connector) is a developer framework designed to allow M-Files to interact with Microsoft SQL without having to first apply the M-Files development technologies to create the integration code. The Connector allows for using SQL procedures to perform the integration operations. In the background, the Connector use the standard M-Files API's, Vault Application Framework, and UIX framework as SQL CLR assemblies and M-Files Applications.

[Business cases](#)

The Connector targets a range of business cases including but not limited to:

- Integration of M-Files with other applications with two-way integration between M-Files and MS SQL
- Creating M-Files Add-on applications using SQL as the extended database for these applications
- Advanced M-Files metadata cleansing and metadata management
- M-Files metadata reports using any report writer or integrating the reports with in-house web applications

In order to expose M-Files metadata, MFSQL Connector imports data from M-Files using LSConnectWrapper as CLR (Common language Runtime) assemblies in SQL, wrapping the interop.MFilesAPI into SQL. Bi-directionally updates using a range of T-SQL procedures interacts with the assembly to perform the integration processes.

REPORTING WITH MFSQL CONNECTOR

MFSQL Connector is installed on the SQL server into a separate database and does not interact directly with the M-Files Vault database and requires the M-Files client to be installed on the SQL Server.

Certification

MFSQL Connector is a certified M-Files Add-in in terms of the M-Files Certified Application Developer (CAP) program.

Components

Assemblies: These are proprietary programs (.dll) to enable the interaction between M-Files and SQL. The main purpose of the assemblies is to wrap the standard M-Files API's into code classes that allows data to be extracted from M-Files into SQL tables and to update M-Files from the SQL Tables. There are four assemblies included.

SQL Connector Tables: These tables contain the metadata of the vault, settings, and transaction logging data. MFSQL Connector tables are all prefixed with "MF" to distinguish them from non-Connector tables in the database.

Metadata tables: The metadata about metadata table contain the information about the structure of M-Files (e.g. classes, properties, valuelists, etc.) Connector Table related to this type of metadata include Object Types; Classes; Properties; Class_Property; Valuelists; Valuelist items; Workflows; Workflow States; User Accounts; Login Accounts

Utility tables: These are supporting tables and include: Datatypes; Settings; Process; DeploymentDetail

Logging Tables: These tables reference processing logs and include: MFLog; UpdateHistory; BatchProcessLog

M-Files Class Tables: These table represent the data in the vault such as Customers, Contacts and any other class in the vault. Class tables are not created by default, but created when the data that is required for the specific application has been determined. The Class Tables therefore represent the metadata of the specific classes of objects that the application will interact with.

SQL Procedure & functions: These T-SQL procedures and functions perform operations on the tables and the assemblies. These procedures are all prefixed with "spMF".

Context Menu: Execute SQL Procedure or URL from within the M-Files desktop context and include:

- In M-Files
 - M-Files Vault Application
 - UIX components
 - Configuration Content Package

REPORTING WITH MFSQL CONNECTOR

- In SQL
 - Context Menu Tables
 - Context menu Procedures

- More information is available in the [Connector Guide](#)
- <https://Lamininsolutions.com>