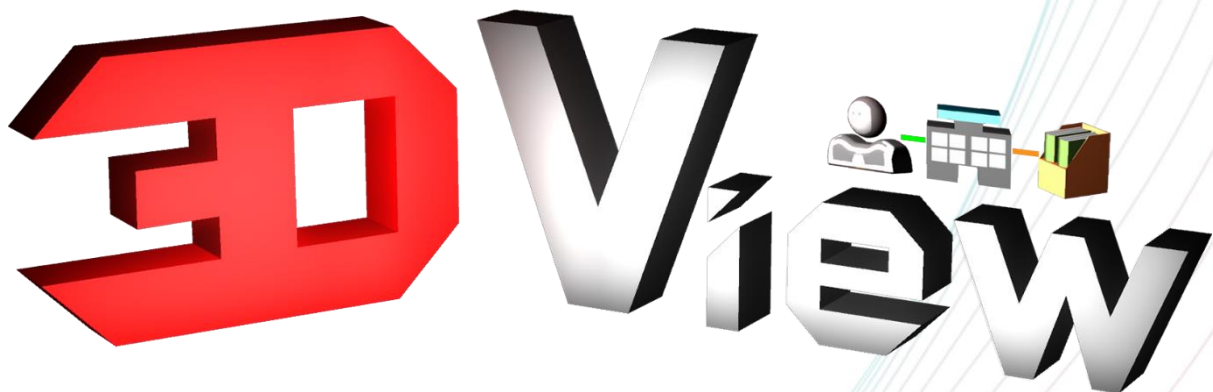


3D visualization of object structures in M-Files system



Student:

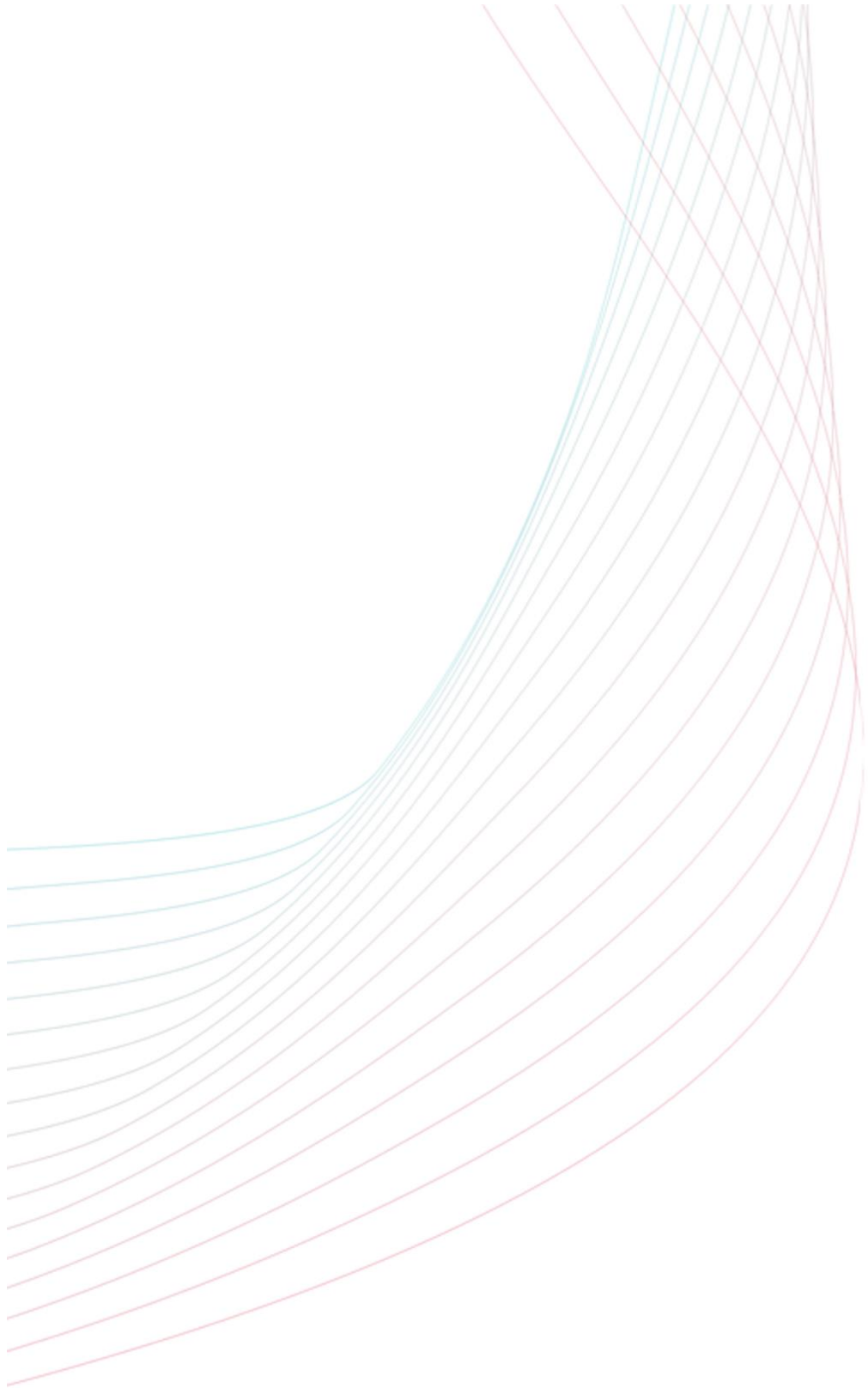
Marián Bañas | 421657
marian.banas@cadvision.sk

cadvision s.r.o. supervisor:

Ľuboš Rodanič
lubos.rodanic@cadvision.sk

Saxion supervisor:

Yvens Rebouças Serpa
y.reboucasserpa@saxion.nl



Glossary

metadata – a basic summarization of information in data

document – a generic term for files containing text or images

object – a generic term for custom objects (a person, a document, a company, ...)

attribute – metadata container on an object's metadata card

type – a unique object type identifier

root – currently centralized object

relationship-from – an object positioned higher in the objects structure than the root

relationship-to – an object positioned lower in the objects structure than the root

relationship chain – a portion of the objects structure consisting of a relationship-from, root and a relationship-to

client – a workstation capable of communicating and obtaining information from a server

server – a computer which manages access and communication with a repository

repository / vault – a place owned by a server where data is stored

view – a window within M-Files explorer containing filtered repository objects

UI – stands for User Interface, a graphical layout of an application

3D – stands for three dimensional, something that has width, height and length

model – a 3D representation of an object in virtual space

texture – an image, used to give models distinct appearance

rendering – a process of generating an image from data

API – Application Programming Interface

framework / engine – a platform for developing software applications

extension / plug-in – software which adds functionality to an already existing software

REST – Representational State Transfer is a software architectural style allowing for interoperability between computer systems on the internet

Contents

1. Introduction	1
1.1. Graduation company	2
1.2. Assignment description	3
1.3. Problem analysis	5
2. Preliminary problem definition	8
3. Problem definition	9
4. Main and sub-questions	10
5. Theory	11
5.1. M-Files' UI extending approaches	11
5.2. Laws of UI design	12
6. Methodology	13
7. Results	14
7.1. Which M-Files API is suitable for client – server communication?	14
7.2. How can the M-Files Desktop client's functionality be extended?	15
7.3. Which low-level 3D API is suitable for rendering 3D geometry?	17
7.4. What other popular ECM software does exist?	19
7.5. How do popular ECM products represent object structures?	20
7.6. How to make a user interface simple and straightforward?	22
7.7. Does the 3D based user interface improve the workflow?	25
8. Conclusion	27
9. Recommendations	28
10. Appendix	29
10.1. Reflection	29
10.2. Sources	33
10.3. Questionnaire	34
10.4. Installation manual	41
10.5. User manual	49
10.6. Visual Studio project's UML diagram	57

1. Introduction

Slovak company cadvision s.r.o. is an exclusive reseller of products made by Hexagon, Bricsys, Honeywell, Simtronics, Fluidflow and M-Files on the Slovak and Czech market, and was established in the year 2012 based on years of experience in the fields of consulting, analysis, setup, implementation, technical support and specialized training for advanced users of CAD and ECM software.

M-Files is an electronic document administration tool with focus on relationships between objects in object structures. This document focuses on the analysis and resolution of the M-Files software's user interface problem. According to its users, in a process of viewing object to object relationships, it displays currently irrelevant content together with the content of interest. Further, the object relationships are depicted in a flat, table-based way where often long text serves as the only means of object type recognition. M-Files does use icons to represent the object types, but these are very small and cannot be resized. These factors often result in confusion and time loss when working with the object to object relationship chain.

The aim of cadvision s.r.o. is to improve the workflow of their clients as well as their own when working with project, commercial and technical documentation. This thesis focuses on making use of 3D space as opposed to the flat, table-based object structures representation, which allows for better spatial distribution of objects within the structures, where the user would also have control over the currently displayed content thanks to filters. Furthermore, a richer use of intuitive imagery was made, to represent object types and reduce the need of text.

Since the product development is being executed by a technician, student of a bachelor's degree Creative Media & Game Technologies at Saxion University of Applied Sciences and not an artist, the priority is to provide essential M-Files functionality over visually appealing interface within the final product. The models and textures used in the final version of the product are expected to be only placeholders. However, it will be possible to customize the software without the need to adjust the product's code.

A group of experienced M-Files users will be asked to test and compare the M-Files Desktop Client with the product of this research, 3DView. Their rating and feedback will be collected and considered during the future iterations of development to ensure 3DView meets the users' demand as precisely as possible. Apart from improving the workflow, cadvision s.r.o. will get to explore the powers of 3D graphics as well as the APIs of M-Files in more depth and this way broaden their knowledge base.

1.1. Graduation company

Slovak company cadvision s.r.o. is an exclusive reseller of Hexagon, Bricsys, Honeywell, Fluidflow, Simtronics and M-Files companies' products on the Slovak and Czech market.

The company was established based on years of experience in the fields of consulting, implementation, setup, analysis, technical support and specialized training for advanced users of Autodesk, Hexagon and Bricsys companies' software. cadvision s.r.o. also develops and sells plug-ins for this software.



"Do you have a hard time choosing from a wide range of solutions for projection, construction and document management? We are going to prepare a presentation of products from Autodesk, Hexagon, based on an analysis of processes in your company." (cadvision s.r.o., 2015)

Website: www.cadvision.sk
Technical support: support@cadvision.sk
Visiting address: Československej armády 6226/3, 03601 Martin, Slovakia



**Official BricsCAD
Solution Partner**



HEXAGON
PPM
Authorized Reseller

Hexagon, Bricsys, Honeywell, Fluidflow and Simtronics develop computer-aided design (CAD) software for infrastructure engineers and designers to improve their workflow. CAD is the use of computers to aid in the creation, modification, analysis or optimization of a design.

M-Files develops enterprise content management (ECM) software for companies to deal with project, commercial and technical documentation. ECM is a system for capturing, preserving, managing and delivering unstructured content and documents.

1.2. Assignment description

The client is interested to find out whether the use of 3D space, offering the freedom of camera movement and richer spatial distribution of objects in object structures, together with the use of textured 3D models, can eliminate the confusion and speed up the workflow when working with object structures in M-Files system.

M-Files is an electronic content management tool with focus on relationships between objects in object structures. M-Files works with structures created from metadata. Object structure can be created for example from combination of

Project – Project phase – Activity
Project – Product – Supplier – Customer

where the dash represents a relationship between two objects, as depicted in the Figure 1.

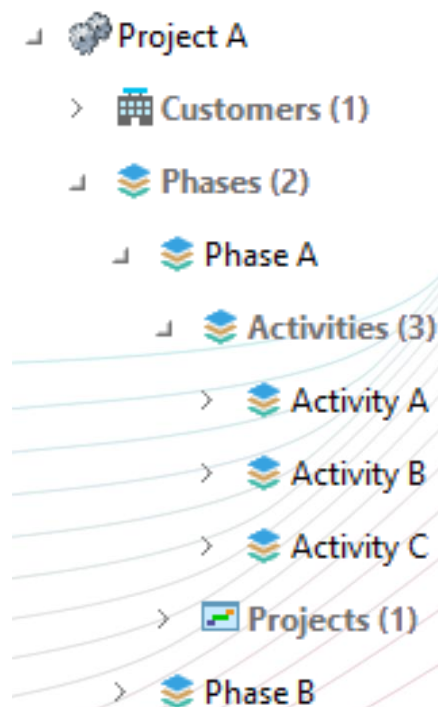


Figure 1: Illustration of Project – Project phase – Activity relationship.

3D visualization is meant to be a replacement of an existing flat way of visualizing relationships by spatial visualization. 3D environment allows for freedom of camera movement and better spatial distribution of objects in object structures.

Spatial visualization consists of relationships, documents, objects, object's attributes, state in life cycle and offers actions such as filtering, searching, navigation, reporting, as well as new object creation. The aim of this product is to improve the efficiency of working with project, commercial and technical documentation.

The client themselves as well as their customers have expressed their requests as the following workflow improvement:

- Realization of advantages of using 3D frameworks / engines in practice (*on desktop, laptop, mobile or virtual reality devices*)
- Simplification of content viewing and content and relationship creation
- Adequate / adaptive document, object and content filtering options
- Keyword based content lookup
- Navigation along relationship chains
- Document and object content summarization and reporting

The 3D visualizer should preferably be controlled from the already existing M-Files explorer as depicted in the Figure 2.

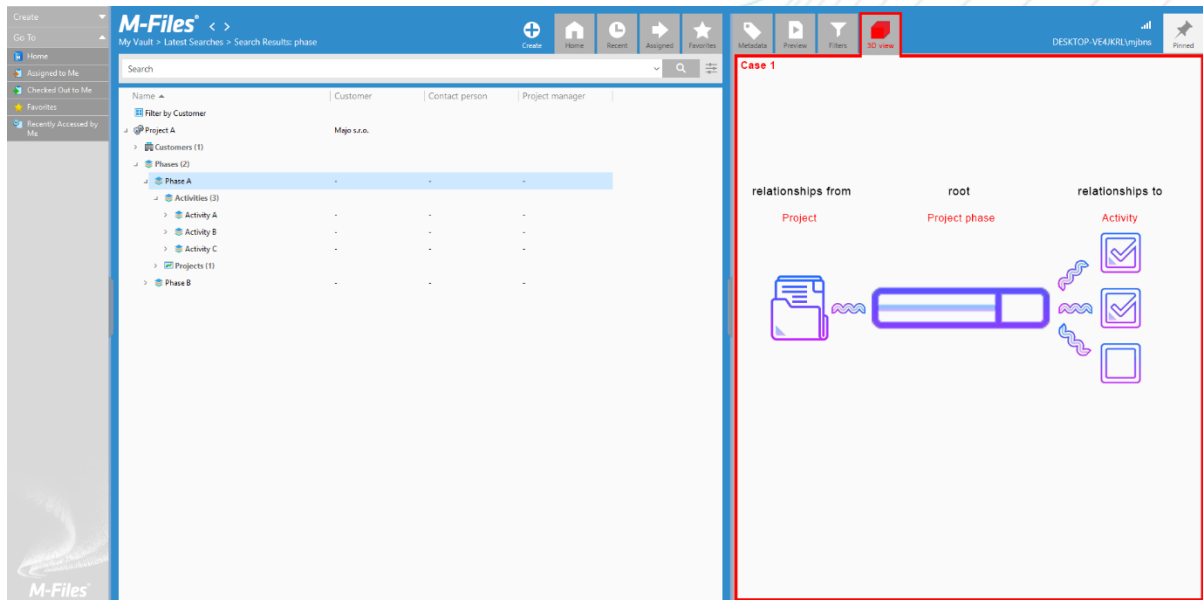


Figure 2: Illustration of the requested UI extension in M-Files explorer.

1.3. Problem analysis

In order to realize the true problem that needs to be resolved, a common case among many M-Files users will be discussed below.

Case:

A company holds a meeting and the status of a project phase is being discussed.

The person leading the presentation has a hard time keeping track of the information on the screen, while switching between navigating along the relationship chain and addressing a certain object during a discussion.

The flat, table-based way of relationship chain visualization of currently selected object “Phase A”, can be seen in the Figure 3.

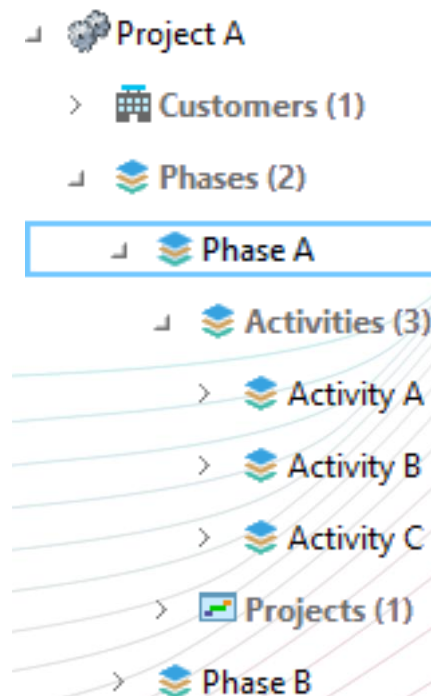


Figure 3: Currently selected object Phase A and it's related objects.

The objects of interest on the meeting are “relationships-to” (Activities), but it is not apparent which objects are “relationships-to”, because also “relationships-from” (Projects) are listed, causing a need to pause for a while, re-orientate and resulting in thought and time loss.

Separate list of “relationships-to” and “relationships-from” can only be viewed with a few extra clicks in a new pop-up window as depicted in the Figure 4 and Figure 5.

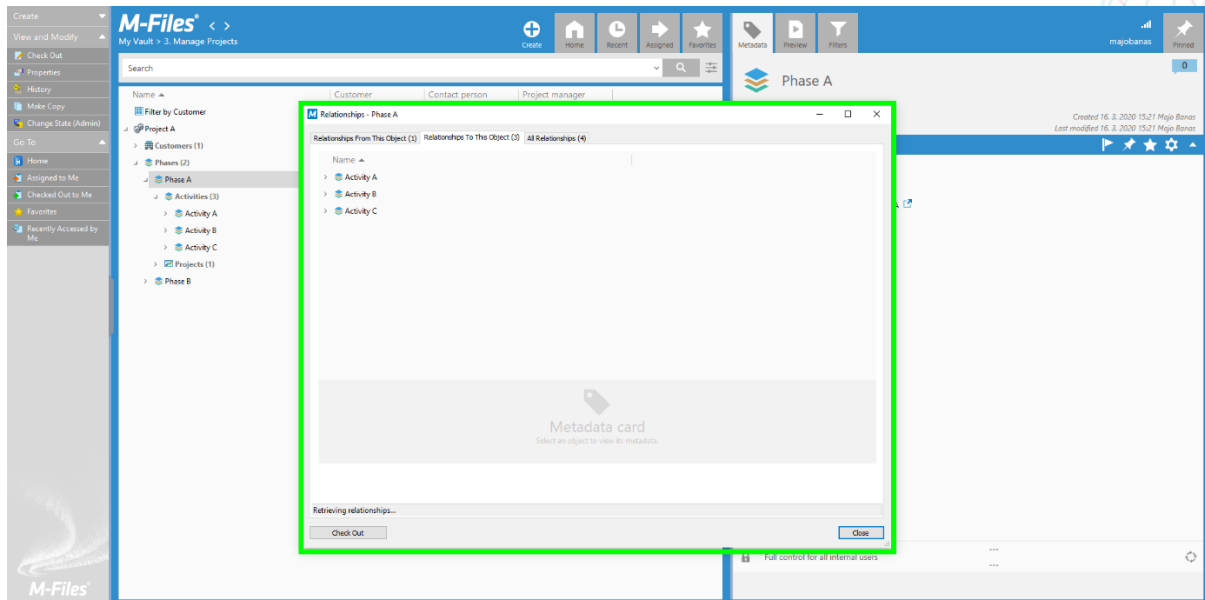


Figure 4: Pop-up window with “relationships-to” a selected object.

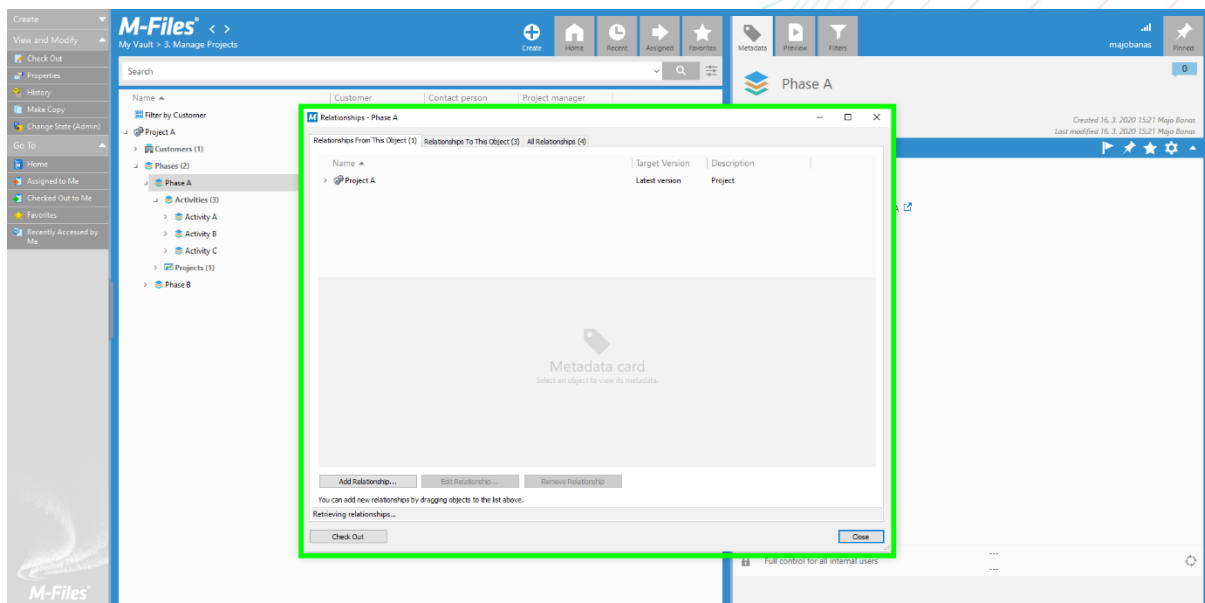


Figure 5: Pop-up window with “relationships-from” a selected object.

Problem:

“Relationships-to” and “relationships-from” of the currently selected object are not properly distinguished from each other in the M-Files’ user interface. Object type icons are too small and text must be read repeatedly to keep track. Re-orientation is time consuming and may lead to though loss.

Cause:

The current M-Files user interface lacks a clearer object to object relationship categorization. Object type icons cannot be resized, because there is no such setting.

Solution:

Visualize only one relationship chain with the selected project phase as the root of the view and one layer of relationships-to and relationships-from as depicted in the Figure 1 in the Case 1 highlighted by the red color.

Make better use of popular, well-known images and symbols to represent viewed object types, in future iterations on the product.

Conclusion:

To solve the problem in the above-mentioned case, the end users have a need for a better object structures representation tool. This tool needs to clearly separate “relationships-from” from “relationships-to”.

2. Preliminary problem definition

To solve the problem of the intricate way the object structures are represented in the current version of M-Files, a clearer, more organized design must be conceived and implemented.

Before the improved way of representing object structures can be implemented, the programmable components of M-Files must be studied.

Since it is already apparent that M-Files does not directly support 3D object rendering the way the client asks for it, it is necessary to construct a custom plug-in for M-Files, which will be capable of delivering the required features.

The plug-in, as it is detailed in the **1.2 Assignment Description** section, must be capable of rendering 3D objects in 3D space as well as filtering of displayed objects, creating new objects, assigning relations to existing objects and looking up objects based on a keyword.

To construct such a plug-in, a suitable graphics library must be chosen, which also runs on casual office computers, as well as a suitable programming language, which allows for communication between the programmable interface of M-Files and the graphics library.

The construction of the plug-in may begin after a correct choice of the above-mentioned tools.

Once the plug-in, capable of rendering 3D objects in 3D space and communicating with M-Files' object structures repositories, is ready, the process of designing, implementing and testing a more comprehensive object structure representation can begin.

Development of such tool requires research in the following topics:

- M-Files APIs
- M-Files extension support
- Low-level 3D API
- Plug-in creation in Visual Studio

Knowledge about the above-mentioned topics will help realize the correct approach for executing the following tasks:

- Design individual layers, user options and spatial distribution of the content, while taking all possible document occurrences and desires to perform an adequate action with them into account.
- Design a scalable code structure allowing for easy maintenance and extension to implement new functionality occur.
- Development of a prototype based on the requirements detailed in the assignment description.
- Connecting the developed prototype with M-Files API to allow for functionality assessment with real and existing data.
- Implementation of 3D visualization prototype into the M-Files software as an extension / plug-in.

3. Problem definition

The problem will be solved by choosing an appropriate integration approach and tools to develop the user interface extension framework.

The most up to date version of this framework can be found on GitHub under the link:

<https://github.com/majobanas/3DView>

Thanks to a functional framework, the resolution of whether the 3D based user interface, as opposed to already existing 2D, flat, table-based user interface, yields more effective workflow, when working with project, technical and commercial documentation within M-Files systems, may begin.

In order to realize this, a clear and meaningful user interface must be designed, implemented and a test must be conducted.

Main problem description

“Relationships-to” and “relationships-from” of the currently selected object are not properly distinguished from each other in the M-Files’ user interface. Object type icons are too small and text must be read repeatedly to keep track. Re-orientation is time consuming and may lead to though loss.

This problem occurs because the M-Files Desktop client’s user interface lacks a clearer object to object relationship categorization and the object type icons cannot be resized, because there is no such setting.

To solve this problem a better object structures representation tool must be available to work with. This tool needs to clearly separate “relationships-from” from “relationships-to”, as well as offer functionality allowing the users to perform routine tasks.

4. Main and sub-questions

Main questions

Does a 3D based M-Files user interface extension, that categorizes objects based on their relationships and uses less text, improve administrative workers' workflow when working with object structures in M-Files systems?

Sub questions

1. Which M-Files API is suitable for client – server communication?
2. How can be M-Files Desktop client's functionality extended?
3. Which low-level 3D API is suitable for rendering 3D geometry?
4. What other popular ECM software does exist?
5. How do popular ECM products represent object structures?
6. How to make a user interface simple and straightforward?
7. Does the 3D based user interface improve the workflow?

5. Theory

5.1. M-Files' UI extending approaches

Below are a few briefly introduced approaches to extending M-Files Desktop client's user interface as well as functionality.

Built-in command events

This approach offers the possibility to invoke external as well as internal events using JavaScript as a scripting language. Once working with Built-in command events, M-Files' internal API implies the code structure. Built-in command events allow asking an external application extension to perform a task and return a value, which can be later used in M-Files internally.

Using managed assemblies

This approach allows for embedding a custom application extension into the local M-Files Desktop client application. Once such application, offering extended functionality, is plugged in, using the **Built-in command events** offered by native M-Files API, results can be delivered directly into the M-Files explorer.

Show webpage in frame

This approach allows creation of a tab / control, where an HTML based context can be displayed. While this approach offers the desired solution, we will not be choosing to follow it, because then the application extension would have to be website based. The client has explicitly stated they do not wish to develop a web-based application extension.

Opening external applications

This approach allows for running an external file using an ActiveXObject as well as passing arguments for its execution, in case it is an executable. Such file has to be installed into a user's M-Files directory so that it can be found when needed.

5.2. Laws of UI design

Well known and frequently utilized laws of UI design should be studied so that the 3DView's user interface can become clear and straightforward. Below are definitions of a few laws of UI design used nowadays.

Doherty threshold

Provide system feedback within 400ms in order to keep users' attention and increase productivity. (*W. J. Doherty & A. J. Thadani (1982) IBM Systems Journal*)

Fitts's law

The bigger the interactive field and the closer to the user's point of attention, the faster the action can be executed. (*P. Fitts (1954) Examining the human motor system*)

Hick's law

The time it takes to make a decision increases with the complexity and amount of options. (*W. E. Hick & R. Hyman (1952) Hick-Hyman law*)

Law of proximity

Objects that are near, or proximate to each other, tend to be grouped together. (*Gestalt laws of grouping*)

Law of uniform connectedness

Elements that are visually connected are perceived as more related than elements with no connection. (*Gestalt laws of grouping*)

Tesler's law

For any system there is a certain amount of complexity that cannot be reduced. (*L. Tesler (1980s) Working for Xerox PARC*)

Jakob's law

If the users spend more time using other solutions, that means they expect your solution to work the same way the ones they already know do. (*J. Nielsen (1995) Consistency*)

6. Methodology

To answer the main question, the topics indicated by the sub-questions will be researched and the first version of the 3D based UI prototype will be tested by a group of M-Files users.

1. Which M-Files API is suitable for client – server communication?

To answer this question, existing M-Files APIs will be studied and compared. The APIs' documentations are available on the official developer's portal.

2. How can be M-Files Desktop client's functionality extended?

To answer this question, the official developer's portal has to be studied and searched for an appropriate approach to extending the M-Files Desktop client functionality. Once an approach is found, it will be tested and adopted.

3. Which low-level 3D API is suitable for rendering 3D geometry?

To answer this question, popular low-level 3D rendering APIs will be compared and the one fulfilling the requirements will be chosen to work with.

4. What other popular ECM software does exist?

To answer this question, a list of popular ECM products with high ratings will be gathered from the sources available on the internet. Since these products score high among the users, they will further serve as a pool of solutions to gather inspiration from on successful UI design practices.

5. How do popular ECM products represent object structures?

To answer this question, the previously gathered ECM solutions will be observed. Since these products score high on their popularity, it is safe to rely on their practices and gather inspiration from their implementation.

6. How to make a user interface simple and straightforward?

To answer this question, various publicly available articles concerning UI design development and creation will be studied and the practices will be adopted to improve the 3DView's UI design prototype.

7. Does the 3D based user interface improve the workflow?

To answer this question, a group of experienced M-Files users will be asked to perform a few tasks within the original M-Files Desktop client as well as the 3DView add-on. Afterwards, they will be asked to rate each process with a grade ranging from 1 to 5. They will also be given the opportunity to write an open feedback.

7. Results

7.1. Which M-Files API is suitable for client – server communication?

Before an object structure is drawn on the screen, it's layout must be retrieved from a server. This is usually done using a dedicated Application Programming Interface (API).

M-Files provides two APIs for exchanging data between a client and the server:

- COM/.NET API
- M-Files Web Service (MFWS)

COM/.NET API

- provides interfaces for “user” and “administrative” functions
- can only be run on Windows
- supports the same connection protocols as the desktop client
- supports the same authentication schemes as the desktop client
- can be run in “client” or “server” mode
- requires the same version of the API on the client and the server machine

Client mode requires a vault connection to be already set up within the M-Files Desktop Settings, and can show M-Files dialogs such as the metadata card for object creation

Server mode does not require a vault connection to be set up on the host machine, but cannot show M-Files dialogs

M-Files Web Service (MFWS)

- a REST-like web service, available from within M-Files Web Access
- can be called from any environment that can make HTTP requests
- is not limited to Windows operating system
- supports most “user” operations, but cannot undertake “administrative” functions
- is not directly tied to the M-Files Server version

The comparison between available APIs can be seen in the Figure 6.

	User / Admin level	Platform	Versions
COM/.NET API	Both	Windows	Server must match Client
MFWS	User only	Cross-platform	Any

Figure 6: Comparison of available M-Files APIs.

The simplicity of REST API based approach and the thorough documentation provided on: <https://developer.m-files.com/APIs/REST-API/Reference/> allows M-Files Web Services to be chosen to enable communication between 3DView and the M-Files repository.

7.2. How can the M-Files Desktop client's functionality be extended?

M-Files hosts a GitHub repository where various ways of extending the M-Files Desktop client application can be found and studied.

The M-Files' GitHub repository can be found on:

<https://github.com/M-Files/MFilesSamplesAndLibraries/tree/master/Samples/UIX%20Applications>

The approaches mentioned below were chosen and together make it possible to interact with M-Files' system.

Built-in command events

The M-Files' user interface extension, called 3DView, can be accessed and launched thanks to the **Built-in command events** functionality. This functionality allows to add a button to the original M-Files user interface and assign a custom callback function to it. The location of the button added using Built-in command events can be seen in the Figure 7.

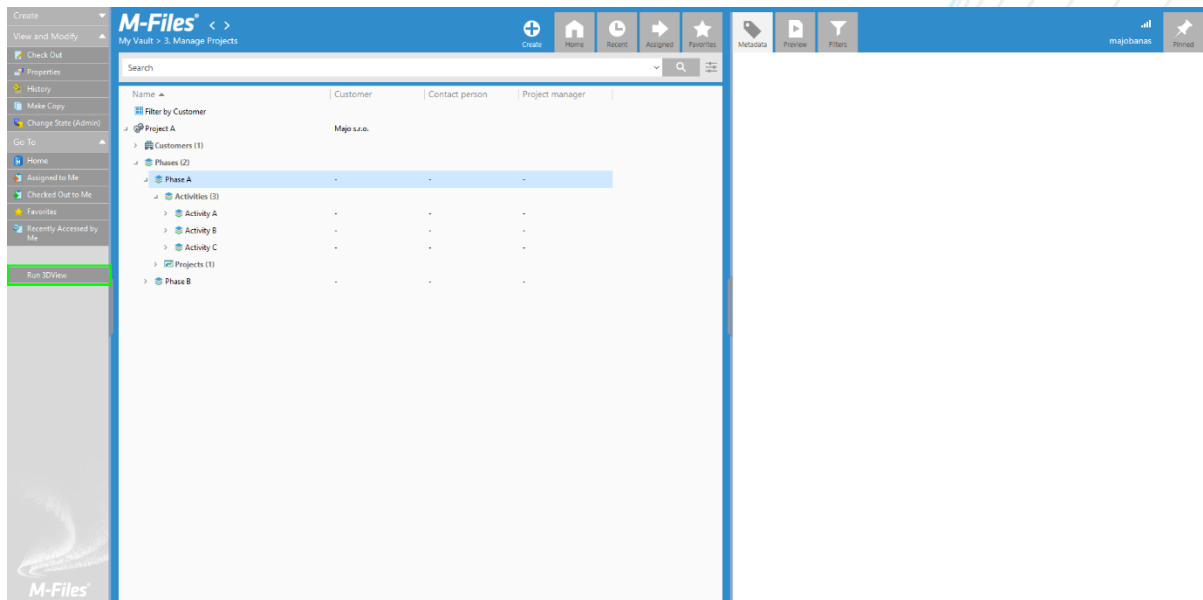


Figure 7: Location of the 3DView launching button.

Opening external applications

Thanks to the ability of M-Files' extension framework to run external files as well as executables, 3DView was built under the hood of Microsoft's Windows Forms as an executable.

This project consists of three layers:

- 3D renderer called 3DView
- renderer wrapper called 3DViewCLR
- Windows Forms Application called 3DViewCS

as shown in the Figure 8.

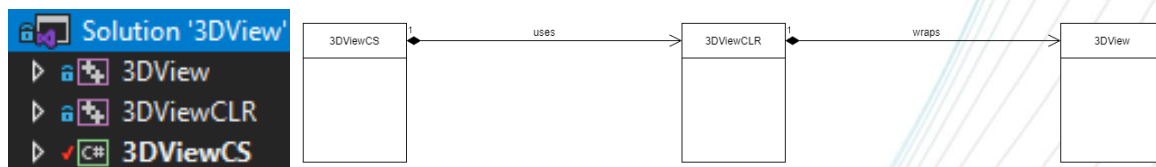


Figure 8: Visual studio project's composition.

The project's structure can be viewed in more detail in the Appendix under the headline **10.6. Visual Studio project's UML diagram.**

3DView

The 3DView controls the graphics' state machine and returns graphics related information to the 3DViewCS on demand. Since OpenGL is used as a low-level 3D graphics API, this sub-project is written in C++.

3DViewCLR

The 3DViewCLR wraps 3DView's functionality and serves as a middleman for exchanging data between 3DViewCS and 3DView. 3DView consists of "unmanaged" code, which cannot be plugged into a "managed" code without a Common Language Runtime wrapper, such as 3DViewCLR.

3DViewCS

The 3DViewCS is written in C#, responsible for communicating with the M-Files server, registering user's input and instructing the 3DView with view content changes.

7.3. Which low-level 3D API is suitable for rendering 3D geometry?

3D graphics have become so popular, that specialized APIs have been created to ease the processes in all stages of computer graphics generation. These APIs provide a way for programmers to access graphics hardware in an abstract way, so that multiple graphics hardware can be targeted, while still having the ability to specify required rendering commands.

There are numerous popular graphics APIs nowadays. Few of them are:

- Three.js (WebGL)
- Direct3D
- OpenGL and the OpenGL Shading Language

Three.js

Three.js is built on top of WebGL. WebGL is a JavaScript API for rendering 2D or 3D graphics within a compatible browser. Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser.

Direct3D

Direct3D is a graphics API for Windows. Part of DirectX, Direct3D is used to render three-dimensional graphics in applications where performance is important, such as games. Direct3D uses hardware acceleration if it is available on the graphics card, allowing for hardware acceleration of the entire 3D rendering pipeline or even only partial acceleration. Direct3D exposes the advanced graphics capabilities of 3D graphics hardware, including Z-buffering, W-buffering, stencil buffering, spatial anti-aliasing, alpha blending, color blending, mipmapping, texture blending, clipping, culling, atmospheric effects, perspective-correct texture mapping, programmable HLSL shaders, and effects.

OpenGL and OpenGL Shading Language

Open Graphics Library (OpenGL) is a cross-language, cross-platform API for rendering 2D and 3D vector graphics. The API is typically used to interact with a graphics processing unit (GPU), to achieve hardware-accelerated rendering. Applications use it extensively in the fields of computer-aided design (CAD), virtual reality, scientific visualization, information visualization, flight simulation, video games and many other cases.

Comparison of the listed low-level 3D APIs' rendering and platform targeting capabilities can be seen in the Figure 9.

	Recommended Language	3D Rendering	Platform
Three.js	JavaScript / TypeScript	✓	Web-browsers
Direct3D	C / C++	✓	Windows
OpenGL	C / C++	✓	Cross-platform

Figure 9: Comparison of low-level 3D APIs' rendering and platform targeting capabilities.

Since the client does not wish to develop an application dependent on web-browser based components, Three.js is not an option. Direct3D is an API rich on capabilities, but targets only Windows platform, which is a potential bottleneck for future product scaling. OpenGL, being a simple, cross-platform API, capable of performing tasks on GPU or CPU based graphics cards will be therefore chosen to work with during the development of M-Files user interface extension.

7.4. What other popular ECM software does exist?

The five most popular ECM products according to G2 and TrustRadius (2020, May 26) can be seen in the Figure 10 and Figure 11.






ECM	Stars (5)	Reviews	Logo
Laserfiche	4.5	207	
eFileCabinet	4.4	268	
DocuWare	4.4	137	
Zoho Docs	4.4	99	
Xerox DocuShare	3.9	115	

Figure 10: g2.com's Best Enterprise Content Management (ECM) Software.

ECM	Stars (5)	Reviews	Logo
Laserfiche	4.5	64	
M-Files	4.5	104	
PowerDMS	4.5	128	
Box	4.5	447	
MS SharePoint	4	1546	

Figure 11: trustradius.com's Enterprise Content Management Products.

Since these products score high in popularity among the users, they will further serve as a pool of solutions to gather inspiration from on successful UI design practices. The extraction of common UI design practices is done in the next section 7.5. **How do popular ECM products represent object structures?**

7.5. How do popular ECM products represent object structures?

3DView is meant to make use of the 3rd dimension while depicting object structures and the relationships between its objects. It is evident that all of these ECM products represent the repository content in a flat, table-based way.

The user interface of each of the collected ECM products can be viewed under the “Videos” or “Screenshots” section on:

Laserfiche - <https://www.g2.com/products/laserfiche/reviews>

Xerox DocuShare - <https://www.g2.com/products/docushare/reviews>

DocuWare - <https://www.g2.com/products/docuware/reviews>

Zoho Docs - <https://www.g2.com/products/zoho-docs/reviews>

eFileCabinet - <https://www.g2.com/products/efilecabinet-efilecabinet/reviews>

M-Files - <https://www.g2.com/products/m-files/reviews>

PowerDMS - <https://www.g2.com/products/powerdms/reviews#reviews>

MS SharePoint - <https://www.g2.com/products/microsoft-microsoft-sharepoint/reviews>

Box - <https://www.g2.com/products/box/reviews>

Unfortunately for the purpose of this research, 3rd party software does not return much tips and inspiration on how to cope with laying out the content of a repository in three-dimensional space, but on the other hand, it does provide inspiration on other important aspects each ECM software must not forget about.

Each of the products:

- requires the user to authenticate before they can access a document repository
- depicts the documents in a structured manner inside a view of some kind
- enables for creation of new objects
- depicts an object's metadata in a metadata card
- allows for adjusting the metadata
- implements versioning and offers access to previous versions of an object
- allows for task-oriented workflow and notifies adequate users
- shows an object's relationship to the viewed structure
- distinguishes objects based on their type
- and the list goes on

Since M-Files also does provide these features, the 3DView add-on must also implement a solution for performing these tasks. Otherwise, if 3DView did not offer the ability to perform these routine tasks, 3DView would only be a structure viewer and not a practical tool / a complete alternative to the M-Files Desktop client.

Apart from solving the problems indicated in the assignment description, a complete alternative to the M-Files Desktop client is something cadvision s.r.o. would also appreciate.

7.6. How to make a user interface simple and straightforward?

Several laws of UI design earlier explained in the theory section 5.2. **Laws of UI design**, were taken into account when designing the user interface and the results can be seen below.

Doherty threshold

This law was taken into account when programming the UI extension framework, particularly instead of repeatedly requesting the same data from the server, the data is being cached. This reduces the time required for performing tasks such as opening an object's attached file, exporting an object's metadata into an Excel file or unfolding an object's relationships.

Fitts's law

The use of this law can be observed when a user desires to perform an action with a currently selected object. Upon clicking the right mouse button, an action bar shows up at the cursor's location, allowing for quick task execution. See the Figure 12.

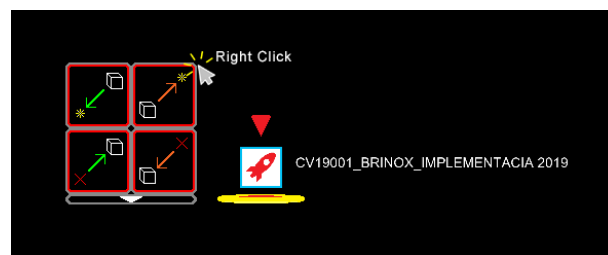


Figure 12: Fitts's law & the action bar.

Hick's law

The use of this law can be observed when a user desires to open an action bar. If the object has a file attached to it, an extra icon shows up. If the object does not have a file, only default icons show up. See the Figure 13.



Figure 13: Hick's law & the action bar.

Law of proximity

The use of this law can be observed when unfolding an object's relationships. The "relationships-from" unfold to the left-hand side from the object and the "relationships-to" unfold to the right-hand side from the object. See the Figure 14.

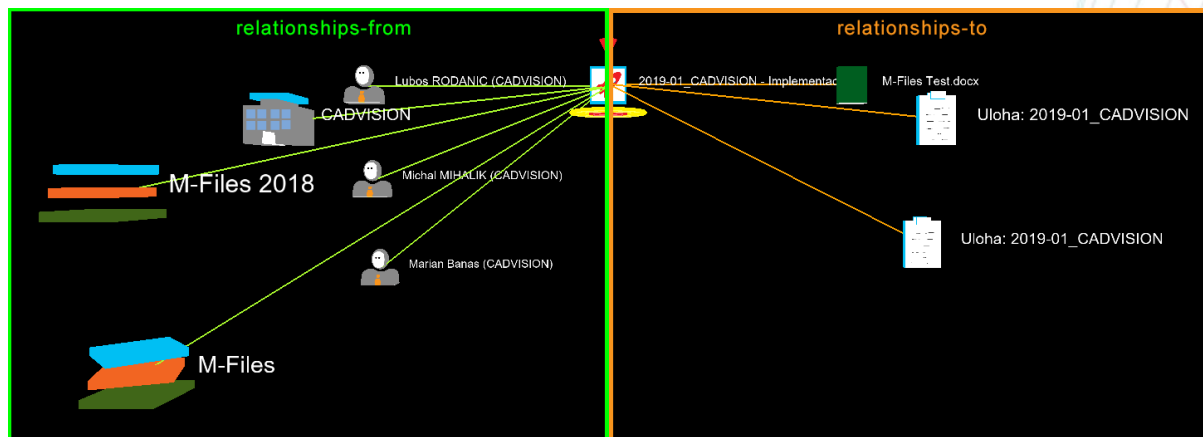


Figure 14: Law of proximity & an object's relationships.

Another example of proximity is when an object's metadata is displayed. It shows up next to the button which reveals the metadata card, as well as next to the action bar relevant for the selected object. See the Figure 15.

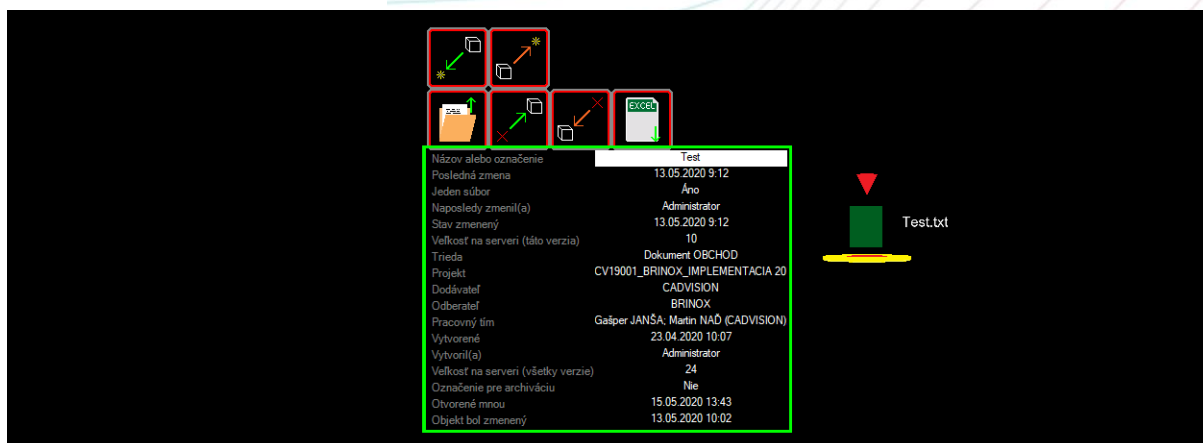


Figure 15: Law of proximity & an object's metadata card.

Law of uniform connectedness

The use of this law can be observed when viewing the relationship chains of objects. Multiple objects can be seen at the same time, but actual relation is depicted by a line connecting two objects. See the Figure 16.

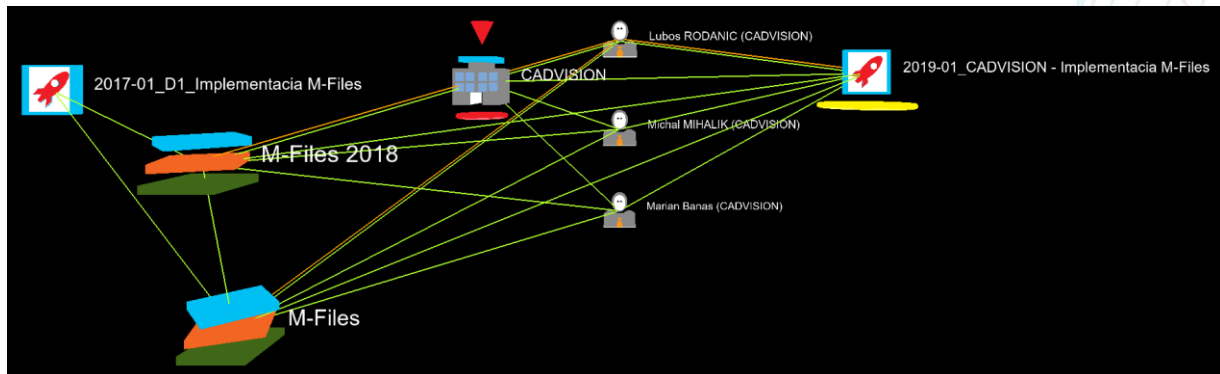


Figure 16: Law of uniform connectedness & the lines depicting a relation.

Tesler's law

There are cases when the viewed relationship chain becomes too chaotic, such as in the Figure 14, but it is entirely up to the user, whether such situation is necessary or not. The user has an option to erase the whole scene and continue from any object selected.

Jakob's law

The use of this law can be observed when moving or rotating the camera. The camera controls were made as similar to popular CAD software as possible. According to the client, the M-Files users within cadvision's reach are experienced CAD software users. This way they do not have to get used to different controls, but focus on their task.

7.7. Does the 3D based user interface improve the workflow?

To find out whether the 3D based user interface does improve the workflow, an online test, due to an ongoing pandemic, was conducted.

The testers were provided with a questionnaire and a user manual, which can be found in the **10.3. Questionnaire** and the **10.5. User manual** sections respectively.

The M-Files Desktop client and the 3D based user interface extension, 3DView, was configured for a vault each tester is already familiar with. This way the tester truly focuses on gathering feedback on 3DView's design and functionality, rather than getting confused by new vault's object-structure.

The tasks the testers were asked to perform are defined in the questionnaire's questions itself. The testers were asked to:

- read the question
- perform the related task in M-Files Desktop client as well as in the 3DView extension
- rate the process of performing each task within the 3DView extension in comparison to M-Files Desktop client.

Grade leaning towards 1 indicates that performing a task within the original M-Files' user interface is clearer and therefore the prototype of 3DView's user interface was found confusing or cumbersome to work with. On the other hand, a grade leaning towards 5 indicates that the 3DView's user interface felt clearer and more straightforward in comparison to the original M-Files' user interface.

The testers were also given an opportunity to write open feedback, which allowed the tester to bring across detailed improvement hints to be processed by the developers and considered in future development iterations.

Workflow improvements in comparison to M-Files Desktop Client

7 testers have been asked to test 3DView and the results can be seen in the Figure 17.



Figure 17: Workflow improvements results. (1 - no improvement, 5 - high improvement)

The results of the first test show the testers greatly appreciate the improvements in regard to performing the tasks 2, 3, 4, 5, 7, 8, 9, 10 and 11. The scores assigned to the test scenarios 1 and 6 indicate the improvements are not significantly noticeable.

Open feedback

The testers have also submitted a free written feedback and below are the most common remarks, which will be logged and considered in future iterations of 3DView development.

“I would like to be able to specify more than one keyword separated by a semicolon, while using the objects filter, so that I can filter by multiple criteria.” (Luboš Rodanič, 2020)

“While summarizing an object’s metadata into an excel file, the list exceeds the screens boundaries, causing a need to scroll sideways. This could be reworked so that overflowing text shifts to a new paragraph underneath.” (Andrej Lukáč, 2020)

“The camera controls are intuitive and straightforward.” (Adam Miča, 2020)

“I would like to be able to skip the second log-in process, since I have already signed into M-Files Desktop client once.” (Martin Čarnogurský, 2020)

“The graphical aspects of the whole environment need polishing, so that the product becomes market ready and feels professional.” (Michal Mihálik, 2020)

8. Conclusion

In general, 3DView has shown to be an improvement to the original M-Files' way of categorizing an object's relationships.

However, in regard to all of the requirements stated by the client and the feedback on the user interface design provided by the testers, 3DView does require further development and testing iterations, so that it can become a truly autonomous and polished tool.

The research has shown 3DView is currently one of its kind on the market of electronic content management software, because it uses 3D environment to represent object-structures, while the currently popular products still use 2D to represent the structures and relationships.

Additionally, according to the client cadvision s.r.o., the research in the underlying technology provided by M-Files in a form of various APIs as well as the use of 3D during this project has opened a window of possibilities for future development of existing or new products.

One of the limitations discovered during the prototyping phase was the inability to render the scenery within an already existing M-Files Desktop client window. Fortunately, the client has expressed satisfaction with rendering to an external window as well as noted this approach brings advantages for future feature implementations.

9. Recommendations

It is recommended to go through the designing, prototyping and testing phase as often as possible, preferably on biweekly basis, to ensure the product meets the end users demand as precisely as possible.

It is recommended to invest time into implementing the remaining M-Files Desktop Client functionality into 3DView, so that it can become a fully equipped alternative while performing routine administrative work.

It is recommended to research the pricing of graphics engines such as Unity or Unreal engine and consider using it for future development, as it already contains the underlaying 3D graphics technology and many more useful components and this way prevent the time to be spent on developing already existing tools.

10. Appendix

10.1. Reflection

I. Technological competences

1. Technical research and analysis

On my arrival, the company's programmers did not have enough time to invest into development of M-Files due to other requests with higher priority, though, tasks related to M-Files have been on their list for a few months by then. I was asked to research ways of communicating with M-Files' databases from within an external application, ways to extend the M-Files Desktop Client's user interface as well as a suitable 3D graphics library for rendering the acquired content as 3D geometry. This required me to study the M-Files APIs' documentation, create threads on the M-Files Community forum, pick a suitable graphics. I managed to explore several ways of extending the M-Files' user interface and picked a graphics library with the guidance of online documentation, forums and developers familiar with related issues.

2. Designing, prototyping and realizing

Since the product I was working on, did not have any fundamentals to build on, I had to put the gained knowledge to a test and get familiar with the practices. I have firstly extended the M-Files user interface with a simple clickable button, then I moved onto wrapping a 3D graphics library into a library which could be embedded into a C# application and last but not least, I have extended the M-Files user interface with a Windows Forms based extension capable of rendering 3D graphics. During this stage I gained practical experience with extending M-Files' user interface as well as it's functionality and how to wrap a C++ based project into a Common Language Runtime project, which then can be used in a C# based project. Then I have prototyped a client – server communication model, which was responsible for exchanging real data from M-Files' database.

3. Testing and rolling out

Once the M-Files extension, called 3DView, was plugged into M-Files with it's capabilities, it was time to assess it's functionality with real data located on M-Files' vault. During this phase, I was able to resolve unforeseen situations as well as formatting errors.

II. Designing competences

4. Investigating and analyzing

Since the goal of 3DView is to eliminate confusion arising from the intricate M-Files Desktop Client's representation of the object-structures as well as other visual aspects the users get into contact with during their work, one of the goals of my work was to design a clearer and more comprehensive working environment. I have decided to explore the market of most popular electronic content management software, to get an overview of common practices and inspiration. I found out M-Files has a lot of functional similarities with those products, but all of them make use of 2D based approach to represent the structures as well as the user interface.

5. Conceptualizing

With the results from the user interface design research, I was able to conceptualize the layout of the heads-up display (HUD) components. The more difficult task was to define a way the objects in the object-structure would spread across the 3D space, because their amount can be arbitrary across different M-Files databases. I have had numerous conversations with my supervisors and together we were able to suggest a uniform layout.

6. Designing

Once the HUD and the object layout concept were available, I moved onto drawing and programming what was agreed on. The final versions of the prototypes were reached through a couple of iterations of consulting, producing and testing.

III. Organizational competences

7. Enterprising attitude

During all stages of 3DView development, I have kept in mind the need to represent everything in a simple, straightforward way, whether it is the ease of installation, the simplicity of launching the add-on or using the user interface. I found these things important because the user always wants a quick and simple solution. Long processes tend to discourage the users to continue paying attention and using a product.

8. Enterprising skills

Since everyone at the company has a load of tasks to do every day, such as my supervisors themselves, I have taken the initiative to try out new things and discuss their relevance with my supervisors once I was sure the approach is beneficial. I am not afraid to say I have been the initiator of discussions when I had a need to make certain things clear. It has always been a positive experience to talk to my supervisors and they have been very nice and found time to listen to me.

9. Working in a project-based way

Apart from reporting to the supervisors and consulting the right way to implement a certain feature, the whole product was produced by one student. I have always listened to the demands of my supervisors and the users. The product is meant to enhance the workflow of M-Files users, not mine only, so accepting an opinion is only vital.

10. Communication

Once the 3DView has been developed to a presentable state and the time to collect feedback from external users has come, I presented its capabilities, the problems it aims to solve and explained what is expected from the testers while a questionnaire was to everyone's disposal. These presentations were recorded so that I can look back at my presentational manners and improve where necessary.

IV. Professional competences

11. Learning ability and reflectivity

During the late stages of development of 3DView, while searching for a way to implement one small feature, I have noticed an updated version of M-Files API was released. The updated version seemed to have made a 3rd party programmer's life, such as mine, easier by encapsulating certain boilerplate code. I would recommend using this updated API version in future developments, as it is also the most recent official release, which will probably stay supported for longer.

12. Responsibility

Developing 3DView has taught me many new things. Looking back at the way the whole project was programmed, I see many places to improve. I do realize I am responsible for the way the project was made and I have also mentioned this fact to my supervisors, so that they are informed of potential adjustment needs. This may seem quite negative, but a person from the same field, may understand the impacts of working with new technology. On the other hand, I am proud of achieving the results that can be seen in my work and I am happy to be confronted about any of it's aspects.

10.2. Sources

Wikipedia (2020, April 17) Enterprise content management
from: https://en.wikipedia.org/wiki/Enterprise_content_management

Wikipedia (2020, April 15) Computer-aided design
from: https://en.wikipedia.org/wiki/Computer-aided_design

cadvision s.r.o (2020, January 1) Who is cadvision s.r.o.
from: <http://www.cadvision.sk/>

Hexagon (2020) Who is Hexagon
from: <https://hexagon.com/>

M-Files (2020) What is M-Files
from: <https://www.m-files.com/en/explore-m-files>

M-Files (2020) M-Files developer portal
from: <https://developer.m-files.com/>

M-Files (2020) The COM/.NET API
from: <https://developer.m-files.com/APIs/COM-API/>

M-Files (2020) The M-Files Web Service (REST API)
from: <https://developer.m-files.com/APIs/REST-API/>

GitHub (2020) M-Files Samples and libraries
from: <https://github.com/M-Files/MFilesSamplesAndLibraries/tree/master/Samples>

Wikipedia (2020, February 24) 3D of graphics libraries
from: https://en.wikipedia.org/wiki/List_of_3D_graphics_libraries

Wikipedia (2020, April 14) OpenGL
from: <https://en.wikipedia.org/wiki/OpenGL>

Wikipedia (2020, March 11) WebGL
from: <https://en.wikipedia.org/wiki/WebGL>

Wikipedia (2020, February 5) Direct3D
from: <https://en.wikipedia.org/wiki/Direct3D>

Wikipedia (2020, April 15) Three.js
from: <https://en.wikipedia.org/wiki/Three.js>

G2 (2020, May 26) Best Enterprise Content Management (ECM) Software
from: https://www.g2.com/categories/enterprise-content-management-ecm?utf8=%E2%9C%93&order=g2_score

TrustRadius (2020, May 26) Enterprise Content Management Products
from: <https://www.trustradius.com/enterprise-content-management-ecm>

LAWSOFOX (2020) Laws of UX
from: <https://lawsofux.com/>

10.3. Questionnaire



3D visualization of object structures in M-Files systems

This questionnaire aims to collect feedback on the 3D-based object-structures viewer prototype called 3DView.

The main goal of 3DView is to:

- display an object's relationships in a categorized format (relationships-from / relationships-to)
- use more intuitive imagery and less text
- eliminate confusion and time loss while navigating along relationship chains

Besides the visual improvements, 3DView also aims to offer functionality required to perform routine work, so that it can become a functional alternative to the already existing M-Files Desktop Client.

While filling out the questionnaire, use numbers 1 / 2 / 3 / 4 / 5 (5 being the highest score), to indicate whether 3DView was more comprehensive than M-Files Desktop Client while performing a certain task. Leave free written feedback below each question, if you wish to. This will help us greatly improve 3DView in the future versions, since the user's opinion counts the most! Feel free to respond in English, Slovak or Czech language.

An example answer to a question would be:

1. How would you rate the process of searching for a certain object?

Score: 3

Open feedback: I find both applications' interfaces regarding this task similarly clear.

or

2. How would you rate the process of accessing a certain object's relationships?

Score: 5

Open feedback: I like how 3DView clearly displays an object's relationships in two categories.

***Required**

1.a) How would you rate the process of searching for a certain object? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

1.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer

2.a) How would you rate the process of accessing a certain object's relationships? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

2.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer

3.a) How would you rate the process of filtering a certain object's relationships? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

3.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

4.a) How would you rate the process of navigating along a certain object's relationship chain? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

4.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

5.a) How would you rate the process of viewing a certain object's metadata? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

5.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

6.a) How would you rate the process of editing a certain object's metadata? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

6.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

7.a) How would you rate the process of viewing a certain object's file? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

7.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer

8.a) How would you rate the process of editing and uploading a certain object's file? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

8.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer

9.a) How would you rate the camera controls? *

1 2 3 4 5
Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

9.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

10.a) How would you rate the placement of action buttons on the screen? *

1 2 3 4 5
Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

10.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer _____

11.a) How would you rate the process of starting up the 3DView add-on? *

1 2 3 4 5

Confusing / Cumbersome ☐ ☐ ☐ ☐ ☐ Clear / Straightforward

11.b) Open feedback

Leave a free written feedback below each question, if you wish to.

Your answer

Do you have any other remarks?

Your answer

Google Forms

10.4. Installation manual



Description:

3DView installation manual



Company:

cadvision s. r. o.
46753788
Diaková 135
038 02 Dražkovce pri Martine
Slovenská republika

Represented by:

Ing. Michal Mihálik
Managing director
+421910447783
michal.mihalik@cadvision.sk



M-Files®



HEXAGON
PPM

Authorized Reseller



Content

1. Prerequisites	2
2. Configuration	3
3. Installation.....	6

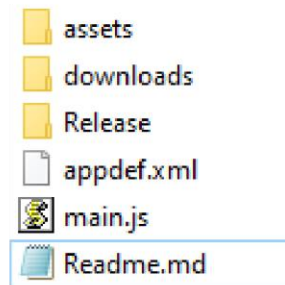
1. Prerequisites

Before you can install 3DView add-on to M-Files Desktop Client, you will need:

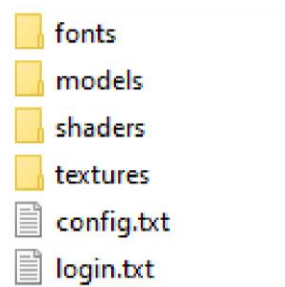
- an existing M-Files document vault
- M-Files Admin tool
- 3DView add-on files

2. Configuration

1. Extract the 3DView add-on files into a folder of your choice.



2. Navigate to a folder **<folder_of_your_choice>/assets**.



3. Find and open the **config.txt** file.



4. Locate the line containing **i default_entry_object_type**

```
//Necessary values to modify per document vault  
  
//Replace ? with object type ID of your choice  
i default_entry_object_type=?  
//Replace ? with 0 or 1  
i load_models=?  
//Leave at least one value on after the = sign  
i models=0,104,103,102,105,10,108,9,101
```

5. Assign a type ID that corresponds with a type ID available at the target vault
(This will be the default object type to load if a user does not select an object before launching 3DView)

6. Locate the line containing **i load_models**

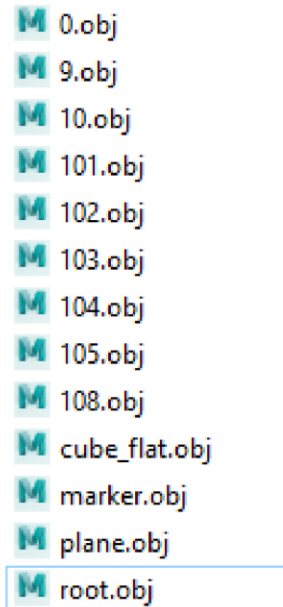
```
//Necessary values to modify per document vault  
  
//Replace ? with object type ID of your choice  
i default_entry_object_type=?  
//Replace ? with 0 or 1  
i load_models=?  
//Leave at least one value on after the = sign  
i models=0,104,103,102,105,10,108,9,101
```

7. If you do not have custom 3D models per object type assign **0**.
(Add-on is ready for installing and you can skip the remaining steps)
8. If you do have custom 3D models per object type, assign **1** and follow the remaining steps.
9. Locate the line containing **i models**

```
//Necessary values to modify per document vault  
  
//Replace ? with object type ID of your choice  
i default_entry_object_type=?  
//Replace ? with 0 or 1  
i load_models=?  
//Leave at least one value on after the = sign  
i models=0,104,103,102,105,10,108,9,101
```

10. Assign all numbers, separated by a comma, corresponding to the object type IDs available on the target vault.
(Example: *i models=1,0,2,6,8,4,5,9*)

11. Navigate to a folder **<folder_of_your_choice>/assets/models/**



12. Make sure this folder contains a model for each of the object types in **.obj** format.

Follow the following naming conventions:

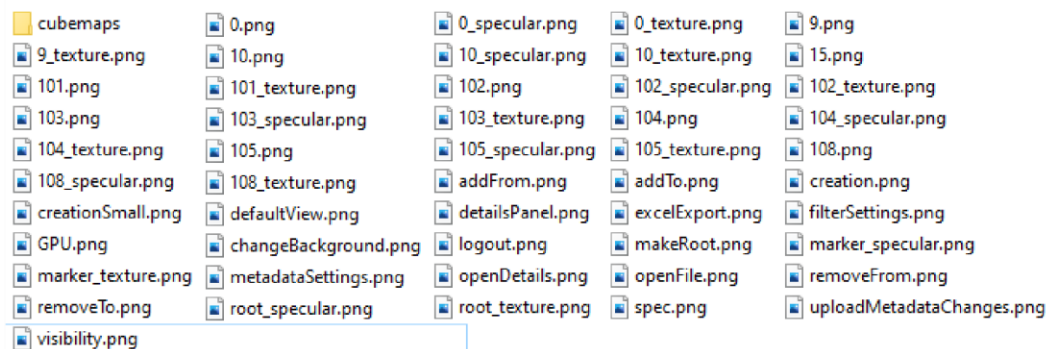
<object_type_id>.obj

(*<object_type_id> corresponds to the object type ID*)

(*Example: 1.obj*)

(*.obj files must contain vertex, normal and uv coordinates*)

13. Navigate to a folder **<folder_of_your_choice>/assets/textures/**



14. Make sure this folder contains a texture and a specular map for each of the object types.

Follow the following naming conventions:

for texture maps: **<object_type_id>_texture.png**

for specular maps: **<object_type_id>_specular.png**

(*<object_type_id> corresponds to the object type ID*)

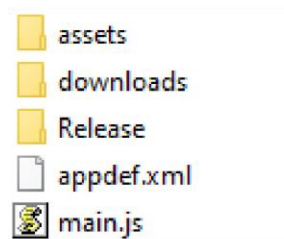
(*Example: 1_texture.png ; 1_specular.png*)

(*Add-on is ready for installing*)

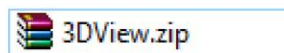
3. Installation

1. Navigate to a folder **<folder_of_your_choice>/**
2. Select the following folders and files:

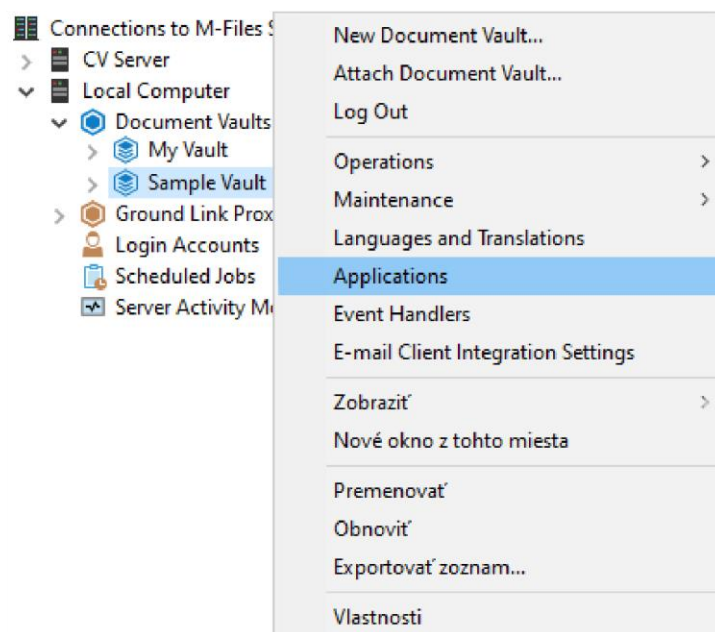
**assets/
downloads/
Release/
appdef.xml
main.js**



and place them into a **<name_of_your_choice>.zip** file.



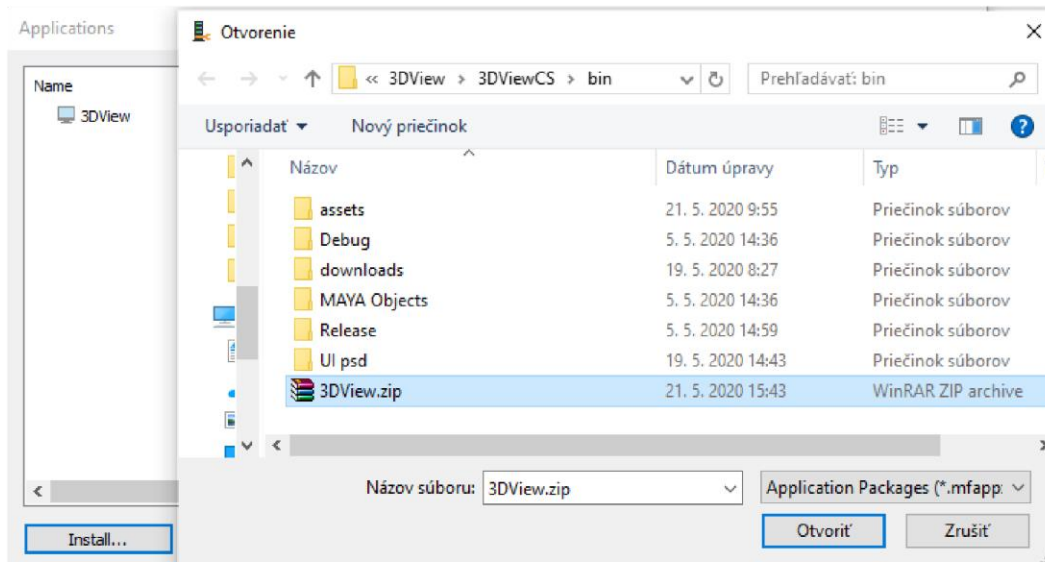
3. Open up the M-Files Admin tool
4. Locate the target vault
5. Right-click it and select **Applications** menu item



3DView installation manual



6. Click **Install** on the pop-up window
7. Locate the **<name_of_your_choice>.zip** in the dialog window and press **Open**
(It will take a few seconds to install the add-on)
(Wait until M-Files Admin tool starts responding)
(Installation is done)



8. Log-out and log-in to the target vault using M-Files Desktop Client and wait for the add-on to download.
(This step is required for each M-Files user once per add-on installation)

Made by: Marián Bañas

Email: marian.banas@cadvision.sk

*Our references are long years of experience and team of specialists...**Cadvision - we execute your visions for higher effectiveness and development of company*www.cadpiping.sk | www.cadvision.sk | www.bricscad.sk | www.fluidflowinfo.com

www.cadvision.sk

Page 7 / 7

10.5. User manual

Description:

3DView user manual



Company:

cadvision s. r. o.
46753788
Diaková 135
038 02 Dražkovce pri Martine
Slovenská republika

Represented by:

Ing. Michal Mihálik
Managing director
+421910447783
michal.mihalik@cadvision.sk



M-Files®



HEXAGON

PPM

Authorized Reseller

INTERGRAPH®



Content

1. Prerequisites	2
2. Launching 3DView	3
3. Camera controls	4
4. Working with vault's object structure	5

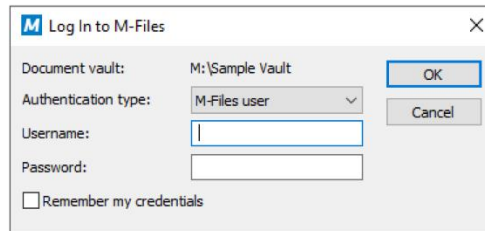
1. Prerequisites

Before you can run 3DView add-on with M-Files Desktop Client, you will need:

- an existing M-Files document vault
- 3DView add-on installed on the vault

2. Launching 3DView

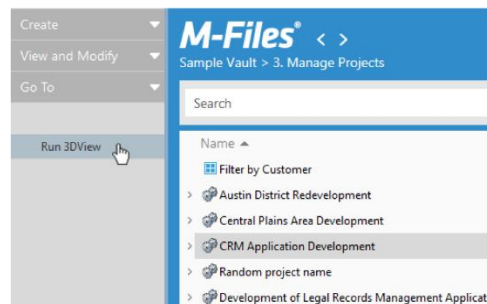
1. Open up M-Files and log-in to a vault as usually.



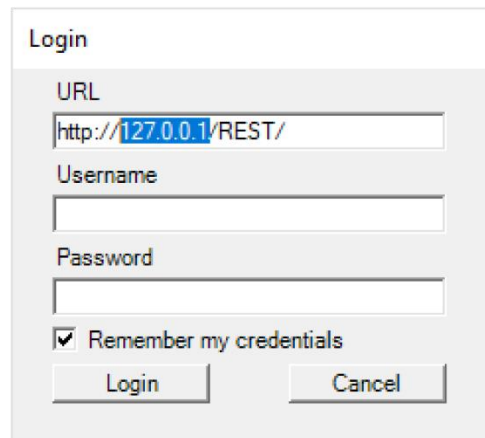
The 'Log In to M-Files' dialog box is shown. It contains the following fields and options:

- Document vault: M:\Sample Vault
- Authentication type: M-Files user (dropdown menu)
- Username: [text input field]
- Password: [password input field]
- ☐ Remember my credentials
- Buttons: OK, Cancel

2. Select an arbitrary object and press the **Run 3DView** button



3. Input the server's IP address without the port and log-in to 3DView with the same credentials



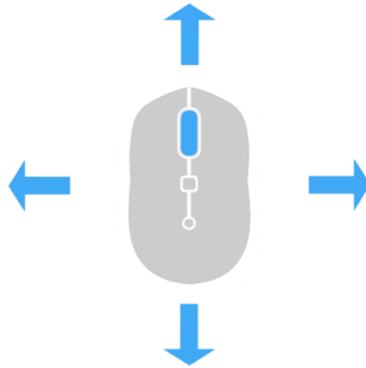
The 'Login' dialog box for 3DView is shown. It contains the following fields and options:

- URL: http://127.0.0.1/REST/
- Username: [text input field]
- Password: [password input field]
- ☒ Remember my credentials
- Buttons: Login, Cancel

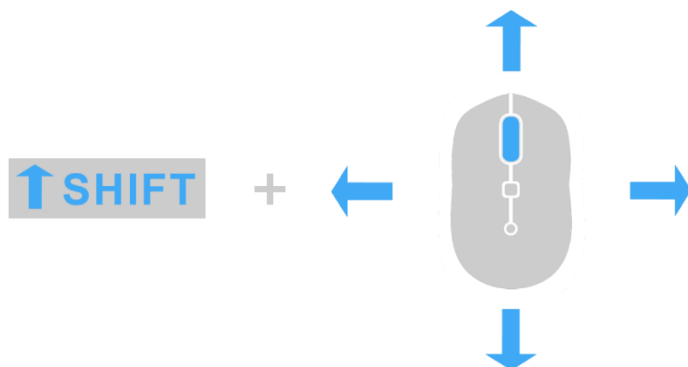
4. Check **Remember my credentials**, so that the next time you do not have to log in.
(You will be able to log out in case you would like to log-in under different identity)

3. Camera controls

1. Pan the camera's view: **hold middle mouse button & move the mouse**



2. Rotate the camera's view: **hold left shift + middle mouse button & move the mouse**

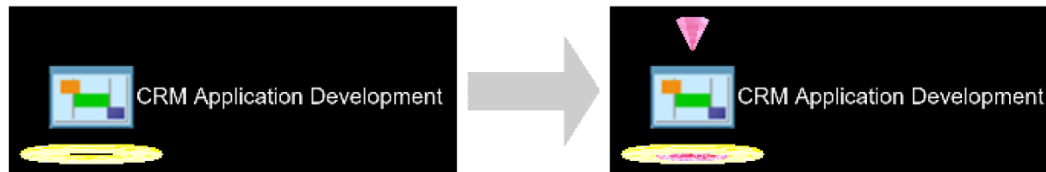


3. Zoom the camera's view: **scroll the mouse wheel**

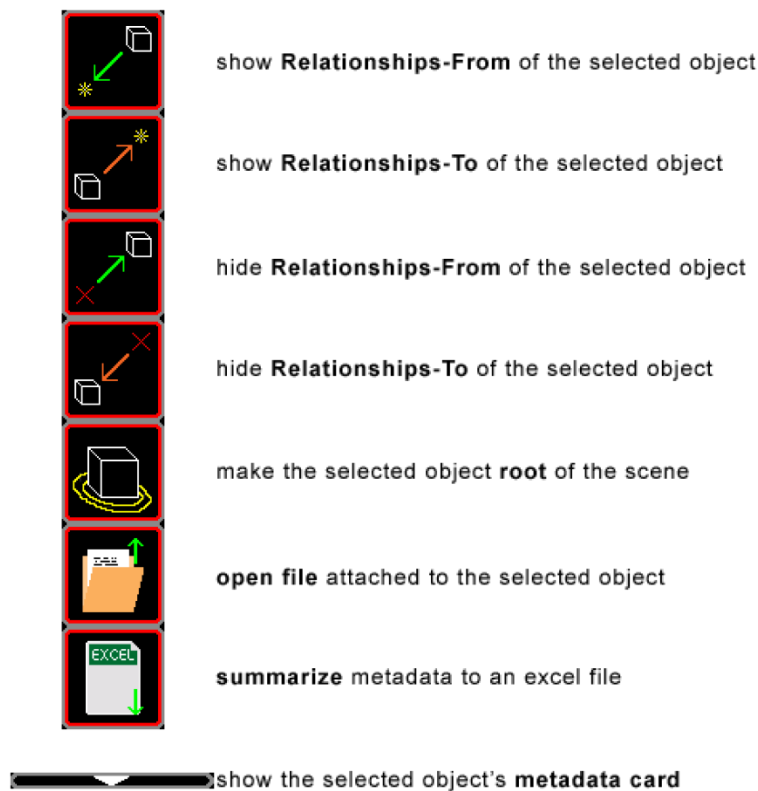


4. Working with vault's object structure

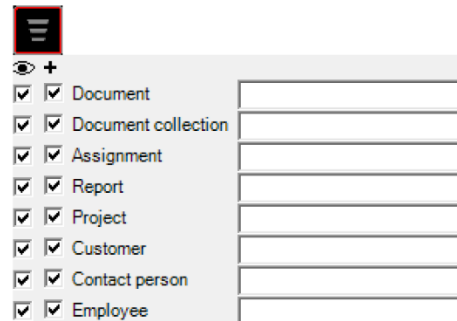
1. **Left-click** an object to select it



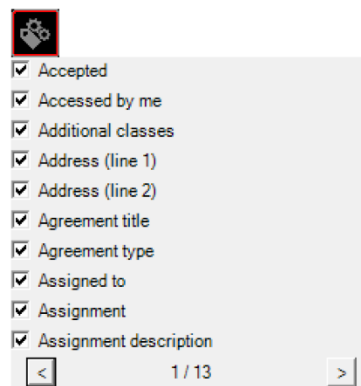
2. **Right-click** anywhere while an object is selected, to open an action bar



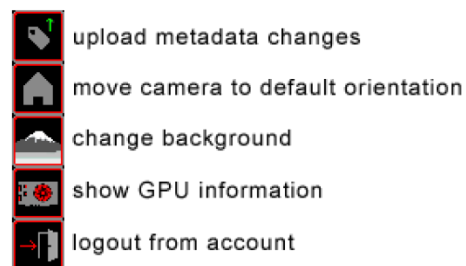
3. Filter objects by type or metadata



4. Filter metadata displayed on a metadata card



5. The rest of the functionality



Made by: Marián Bañas

Email: marian.banas@cadvision.sk

Our references are long years of experience and team of specialists...

Cadvision - we execute your visions for higher effectiveness and development of company

www.cadpiping.sk | www.cadvision.sk | www.bricscad.sk | www.fluidflowinfo.com

10.6. Visual Studio project's UML diagram

