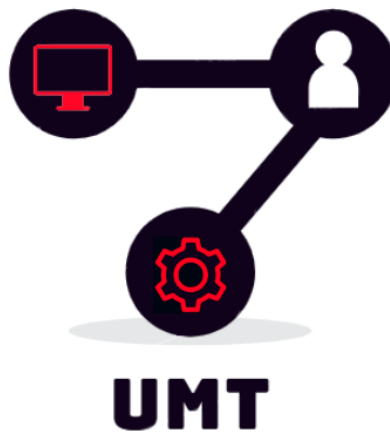


ANV User Management Toolkit

by ANV s.r.o.



User Manual



For more of our products, see: <https://www.anv-tech.com/en/software-development/>

To contact our online support, please send an email to: support@anv-tech.com

Table of Contents

About This Manual.....	3
Abbreviations	3
Requirements.....	3
Installation	3
Licensing and activation	4
Toolkit Description	6
Introduction	6
Configuration	6
Configure Data Storage.....	6
JSON, Binary storage types.....	7
MS SQL storage type	8
SQLite storage type.....	10
Default user	11
Manage Users.....	11
Configuration UI for managing users.....	11
Configuration UI for managing user roles.....	13
Configuration UI for managing privileges.....	14
User Management Basics.....	15
Toolkit API.....	16
Main palette	16
Login subpalette	17
Data subpalette	18
Sample Project	19
Configuration Files	19
Configuration Scope	19
Configuration File Types	19
Configuration Files Location	20
Configuration Files Content	20

About This Manual

This manual serves as thorough introduction for usage of ANV User Management Toolkit. Users of the toolkit are recommended to carefully inspect this manual before using the toolkit.

Information in this manual is accurate for the current version of toolkit to the best knowledge of the authors. We reserve rights for any errors which might appear in this manual.

This document is property of ANV s.r.o. All rights for the document are reserved.

Abbreviations

API – Application programming interface

JSON – JavaScript Object Notation

UMT – User Management Toolkit

VIPM – VI Package Manager

Requirements

These are the minimum software requirements for proper functionality of toolkit:

Windows 7, 8, 10 32 or 64 bit.

LabVIEW 2015 SP 1 or higher.

Following 3rd party VI packages available through JKI VI package manager are used by the toolkit:

1. JKI JSON \geq 1.1.10.37
2. OpenG MD5 Digest Library \geq 4.1.1.10
3. SQLite Library \geq 1.10.0.85
4. MGI Library \geq 1.2.0.4

All packages mentioned above are included in toolkit installer by default.

Installation

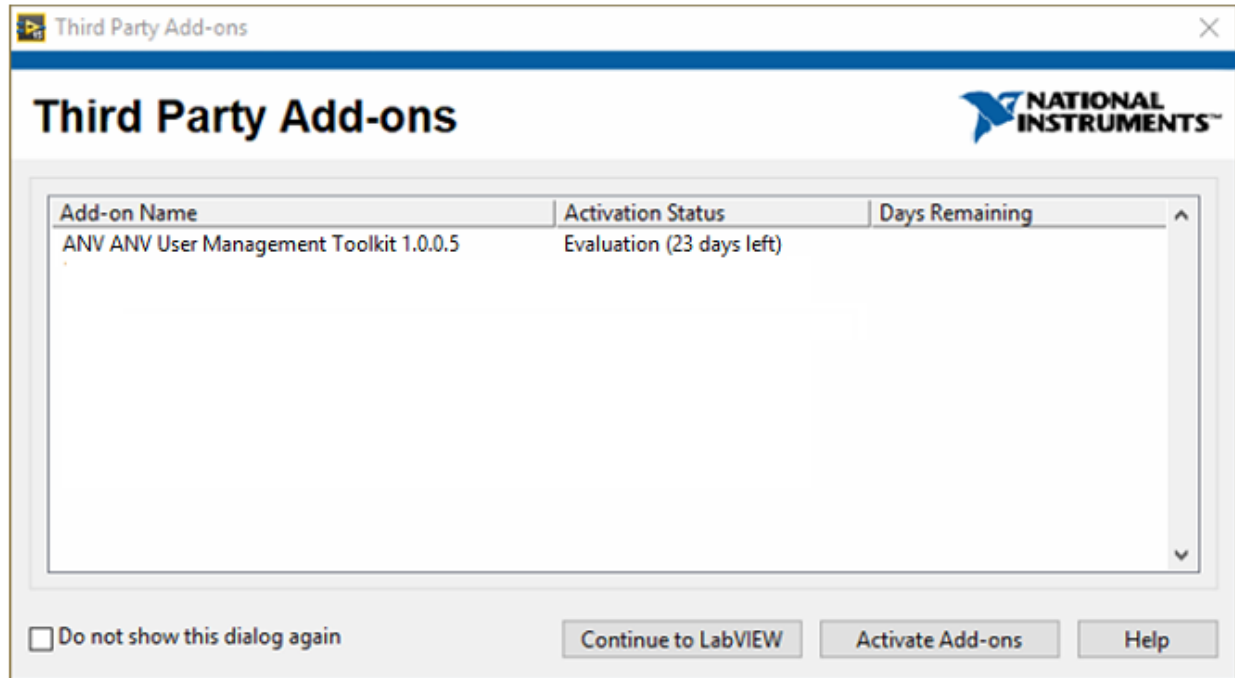
Toolkit is installed as package via JKI VI package manager. You can download it either through VIPM or from our website. After installation, 30-day evaluation period begins. Please see our website for download links.

During the installation/un-installation process, both LabVIEW and VIPM should be launched with administrator rights, otherwise installation may not work.

Licensing and activation

IMPORTANT: During activation of the toolkit, it is necessary that LabVIEW is run with administrator rights. Otherwise, activation process uses your license without activating the toolkit.

ANV UMT is available with single, development licensing option. After the toolkit is installed through VIPM, it comes with 30-day evaluation period. Note that with evaluation version of LST, user cannot build LabVIEW executables – this requires activation of the toolkit. You will be prompted about license status when you launch LabVIEW, unless your license is already activated.



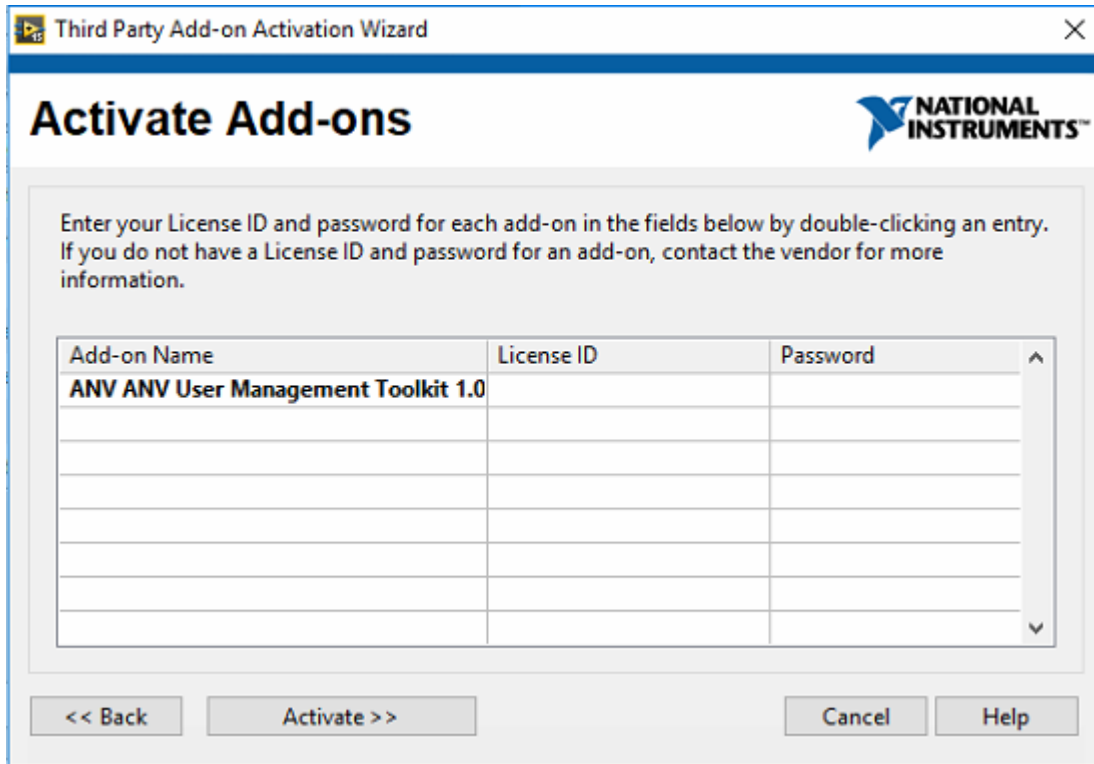
LabVIEW third party Add-on window

To activate license, either click Activate Add-ons button, or use LabVIEW menu. Select Help => Activate Add-ons from LabVIEW Toolbar. Dialog box with the list of available toolkits appears. Select ANV User Management Toolkit from the list and follow the activation wizard in order to activate toolkit.

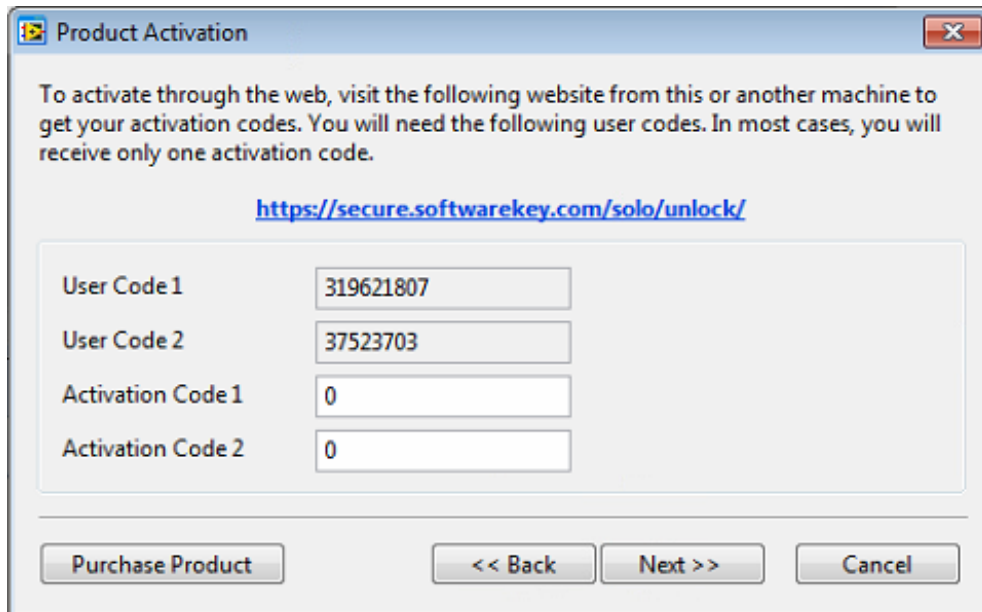
You have two options to activate your license – either automatically or manually. Automatic activation is done online. You enter License ID and Password you received when you purchased license into appropriate cells and click activate.

Manual activation is used, when your machine cannot connect to the internet. When you choose manual activation, user codes appear in dialog. On another machine, open browser and access <https://secure.softwarekey.com/solo/unlock>. Enter License ID and password that you received when you purchased your license and submit them. Then, enter user codes from the dialog of the machine, where you want to activate the license, into appropriate columns. You will then receive activation code, which you may enter below user code fields inside “Activation code 1” cell. Click next. Your license should be activated now.

IMPORTANT: Make sure that user codes you entered online are correct and do not cancel activation dialog until you enter activation code. Your license is used when you enter it online, so if you entered incorrect user codes online, or you cancelled the dialog, you will not be able to activate workstation with the license.



Automatic license activation dialog



Manual license activation dialog

Toolkit Description

Introduction

ANV User Management Toolkit (ANV UMT) is a toolkit, which helps developers with tasks related to management of users in LabVIEW. It allows to create applications with multiple users, possessing different user rights and store created user configuration in different storage types such as JSON, binary files or different database types. Several user interfaces are provided, with aim to simplify creation and modification of users. Also, set of API VIs allows developers to easily handle common tasks related to users, such as storing/ loading user configuration programmatically, logging users, or checking user rights.

The toolkit consists of two main parts – configuration part accessible through LabVIEW tools menu and API, which is available from LabVIEW functions palette.

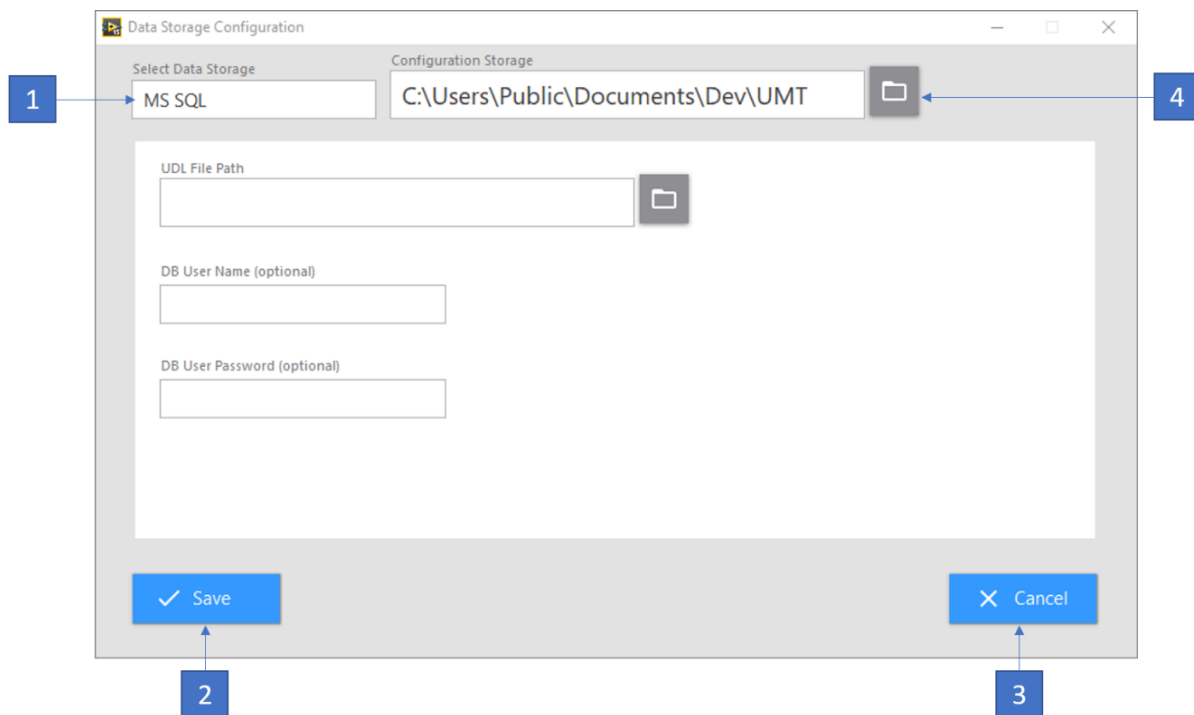
Configuration

Toolkit has the following menu options:

- 1) Configure Data Storage
- 2) Manage Users
- 3) Manual

Configure Data Storage

Configure Data Storage launches “Data Storage Configuration” window. Purpose of this UI is to setup target storage (file or database) of the users’ configuration.



Data Storage Configuration UI

1. Selector of target data storage. Available types: "JSON", "Binary", "SQLite", "MS SQL". Selector also has value "Not Defined", which is not possible to select manually – it is used to indicate error in case of any.
2. Button "Save" saves configured parameters to ini file.
3. Button "Cancel" closes window without changes saving.
4. Configuration Storage folder path selection – selects folder, where configuration file will be saved. By default, this path is set to "<current project path>\UMT" path.

Below is brief description of each storage type configuration.

JSON, Binary storage types

These storages – are JSON and binary files, respectively.

```

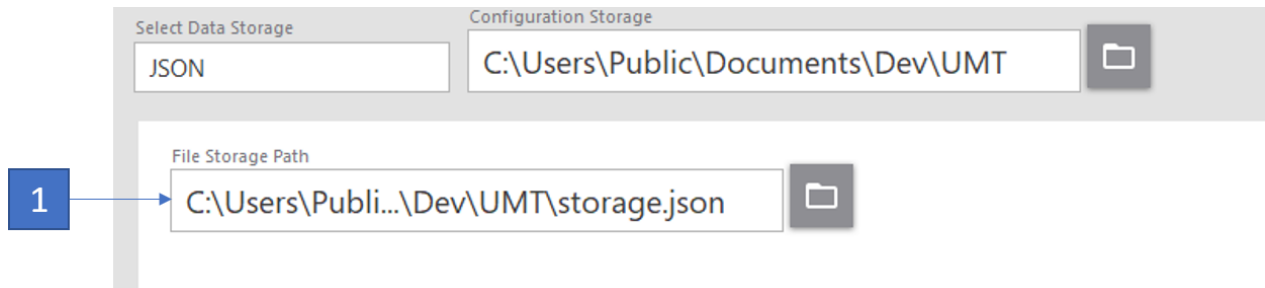
1  {
2  "Users": [
3  ],
4  "Privileges List": [
5  ],
6  "Roles List": [
7  ],
8  "Privileges by Roles": [
9  ]
0  }

```

JSON file structure

Note: it is not recommended to edit JSON file manually.

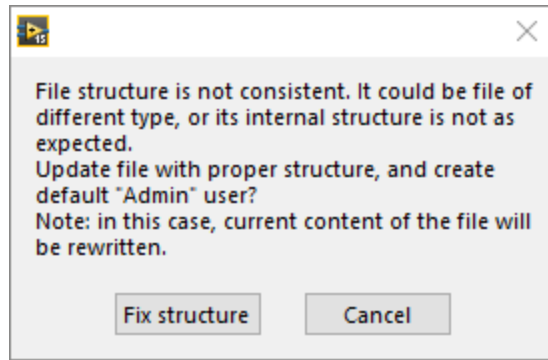
Configuration window for both type of files looks the same way



JSON Storage configuration

1. File Storage Path – this path selector allows to select new/existing files for configuration storage. In case of JSON storage, default file extension is *.json. In case of Binary storage, default file extension is *.txt.

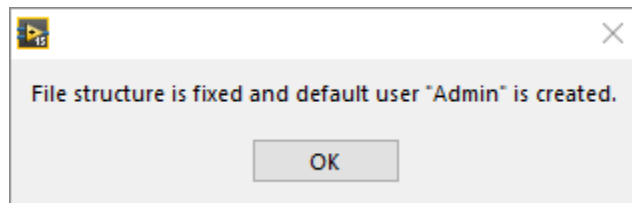
File structure is verified when configuration is saved. In case when structure is incorrect, the following dialog window appears:



Warning dialog window

On “Fix structure” button press, file structure will be fixed, and default user “Admin” will be created. On “Cancel” button press file structure will remain not fixed, and configuration will be not saved.

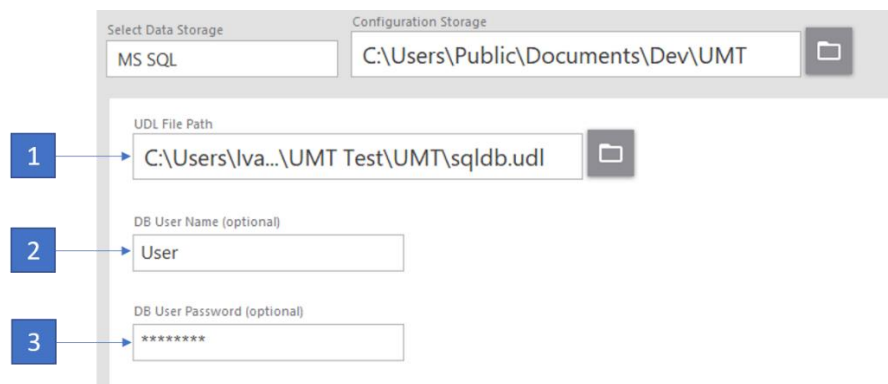
In case of successful operation, the following window appears:



Successful file update message

MS SQL storage type

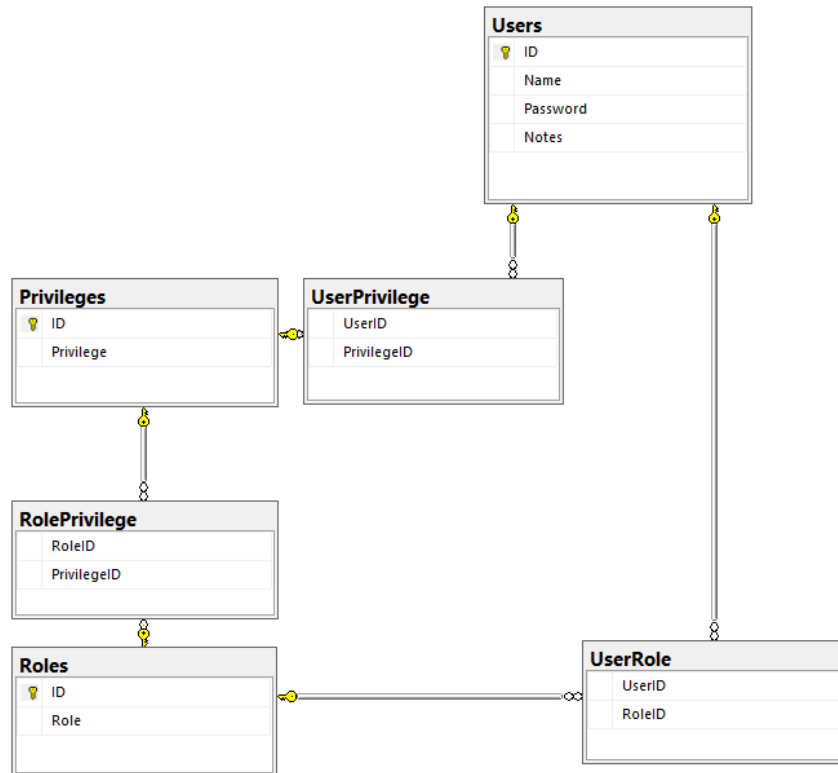
MS SQL storage type – stores users configuration in MS SQL database. In order to use this storage type, MS SQL server should be installed on the target machine (computer).



MS SQL Storage configuration

1. UDL File Path – this path selector is used to select udl or dsn files, which are used to connect to database. These files could be created also by native LabVIEW means, using Tools -> Create Data Link... option.
2. DB User Name (optional) – optional input of database user, if database requires authentication.
3. DB User Password (optional) – optional input of database password, if database requires authentication.

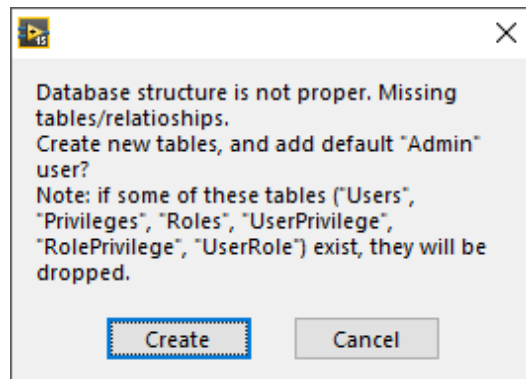
Structure of database is as the following:



Database relationships diagram

In case when target database is empty/does not contain expected tables/does not have expected relationships between tables – then all the tables and relationships between them are created. Note, that in such case existing tables with the same names as expected by toolkit will be dropped, and recreated.

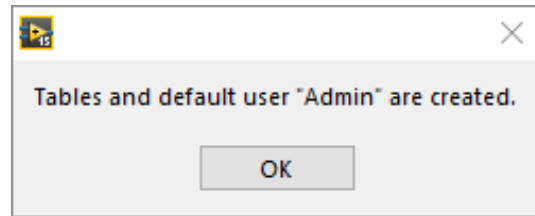
Database structure is verified when configuration is saved. In case when structure is incorrect, the following dialog window appears:



Warning dialog window

On "Create" button press, database structure will be fixed. On "Cancel" button press database structure will remain not fixed, and configuration will be not saved.

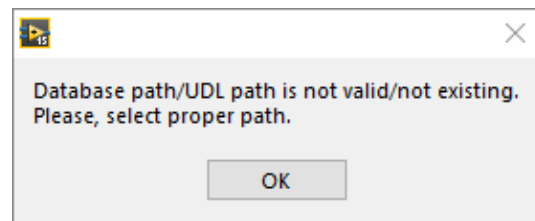
In case of successful operation, the following window appears:



Successful database update message

In case when target database already contains all the tables, and relationships between them are as expected, then this database could be used to store the configuration.

In case when UDL file path is empty/not valid, the following warning window appears:

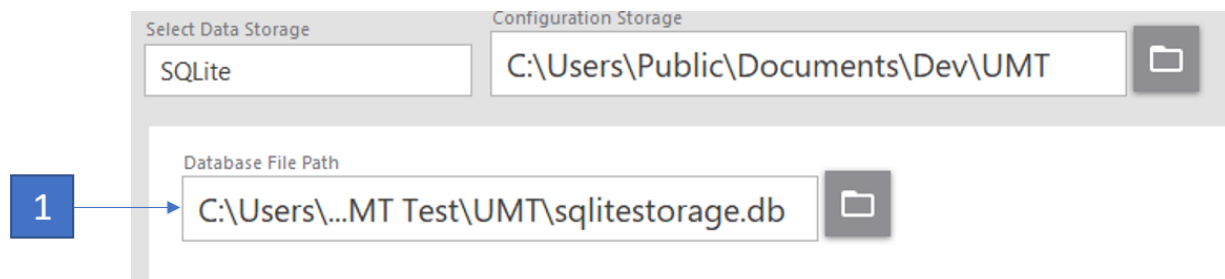


Warning dialog window

So it is possible to save configuration just in case selection of proper udl file, which points to database with expected structure.

SQLite storage type

SQLite storage type – stores configuration to SQLite database file. This type of storage does not require installation of any additional database servers/drivers.



SQLite storage configuration

1. Database File Path – this path selector allows to select new/existing SQLite database file. Default file extension is *.db.

Database structure is the same as for MS SQL database. Also, the behavior of database structure verification is the same – in case of not proper database structure, it is recreated as expected, in the same way as it is described for MS SQL storage type.

Default user

As mentioned above, in case of empty storage selection, its structure is fixed, and default user is created. User's name is "Admin", password is empty, and user has default "Edit configuration" privilege.

Manage Users

Manage Users option launches configuration editor window. Purpose of this UI is to allow developer creation of users, user roles and privileges during development. When launched, it reads configuration from data storage and displays it in several UIs.

This window consists of three parts:

1. Configuration UI for managing users.
2. Configuration UI for managing user roles.
3. Configuration UI for managing privileges.

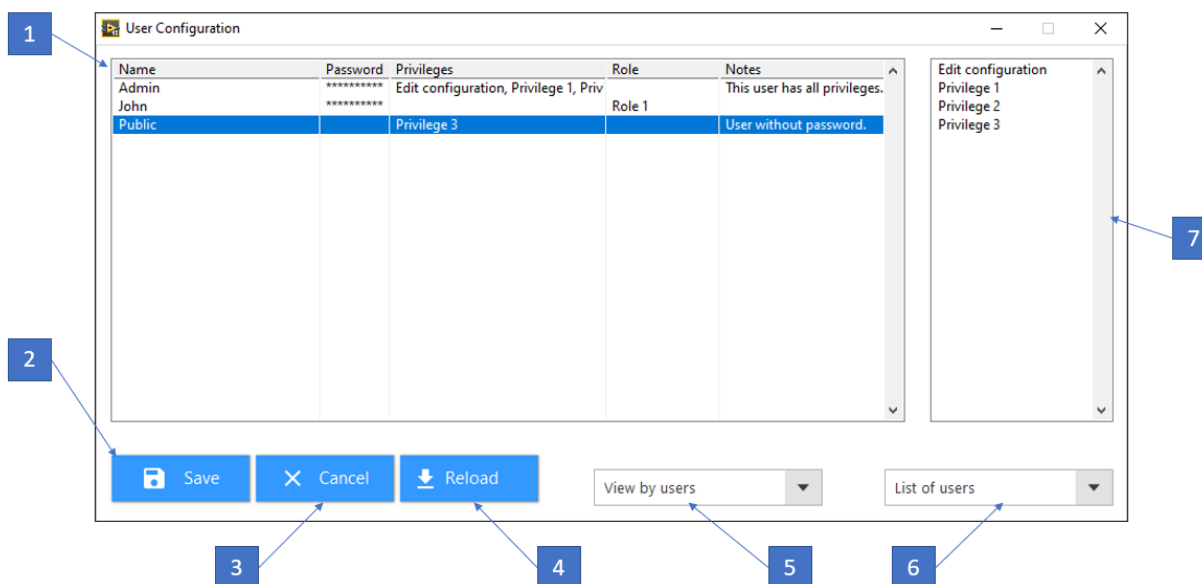
Configuration UI for managing users

Features:

- Add user – right-click menu option, launches edit user UI and allows you to enter data for a new user.
- Edit user – right-click menu option, available when you click on existing user. Launches edit user UI which allows to edit data for this user.
- Remove user – right-click menu option, available when you click on existing user. Removes selected user from configuration.

UI for managing users allows developer to setup different users. Each user should have unique name. You can assign privileges and role to a user, however, both of these parameters are optional. The only restriction is that there always should be at least one user, which possesses privilege to "Edit Configuration", otherwise the configuration could not be saved. This is default privilege of UMT and it is used to edit configuration in API. This restriction ensures that it is always possible to edit UMT configuration in distributed applications by a user.

Other parameters which user can optionally possess are notes, which allow you to store additional info for a given user and password. Password can be optionally setup for each user, but you can also leave this field blank in order to have unprotected user accounts. Note that passwords are encrypted, therefore making sure that the password you enter will be stored safely. For safety reasons, password field does not show actual password in the configuration window, only indicates whether given user has any password (by several "*" characters) , or no password (blank field).

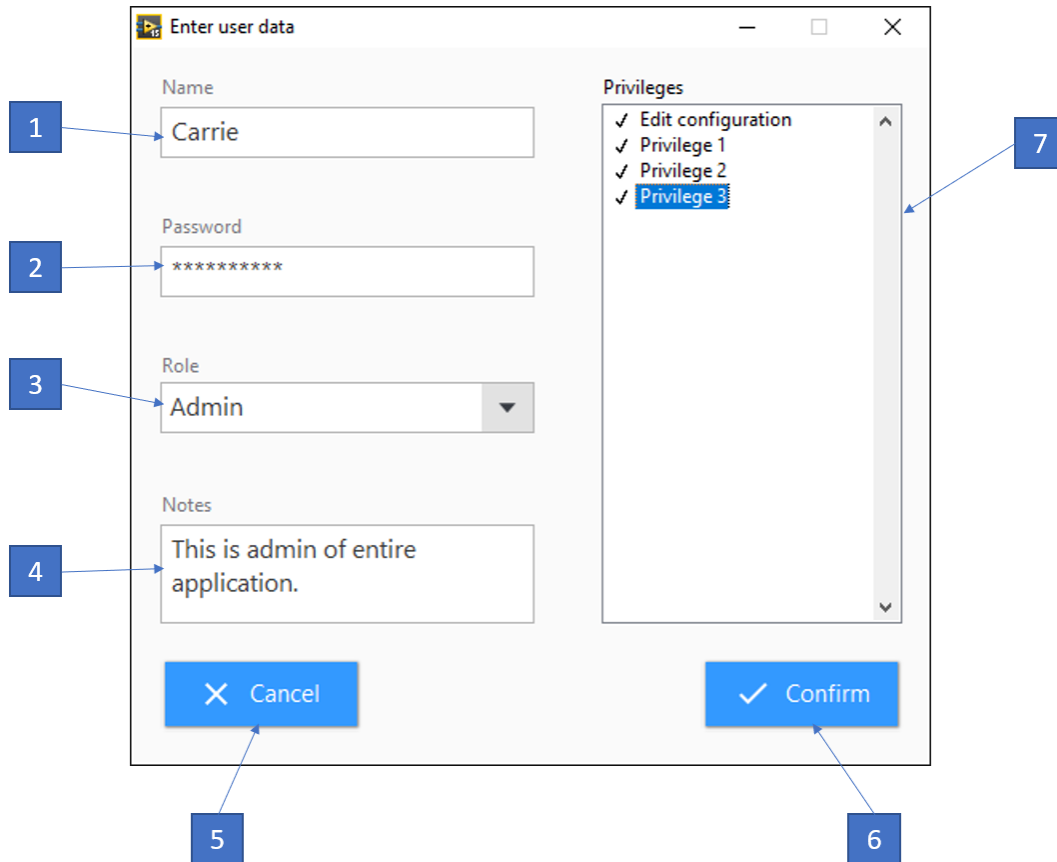


Configuration UI for managing users.

1. Table containing all users and their data.
2. Save button – saves configuration to data storage.
3. Cancel button – discards performed changes, closes configuration window.
4. Reload button – discards performed changes, reloads configuration from data storage.
5. Change user view – changes the way users are displayed. Can be set to either view by list of users, or group users by either roles of privileges.
6. Change window – changes displayed window between Configuration UIs for managing users, privileges or roles.
7. List of privileges for user – displays all privileges for currently selected user. Contains same data as the cell in “Privileges” column of selected user, but in more orderly manner in order to make it easier to view.

UI for managing users can also be switched by different views – you can either view list of users, or users grouped by either privileges or roles. This simplifies management if you use larger number of users. For example, if you have specific privilege and need to know which users possess this privilege, simply switch view to “View by privileges” and look at the list of users who are listed under given privilege. These alternative views also offer simple way to add or remove users from a specific role or privilege.

For editing of user data, separate window is used. It allows you to enter username, password and notes and select role and privileges for user from existing roles and privileges.



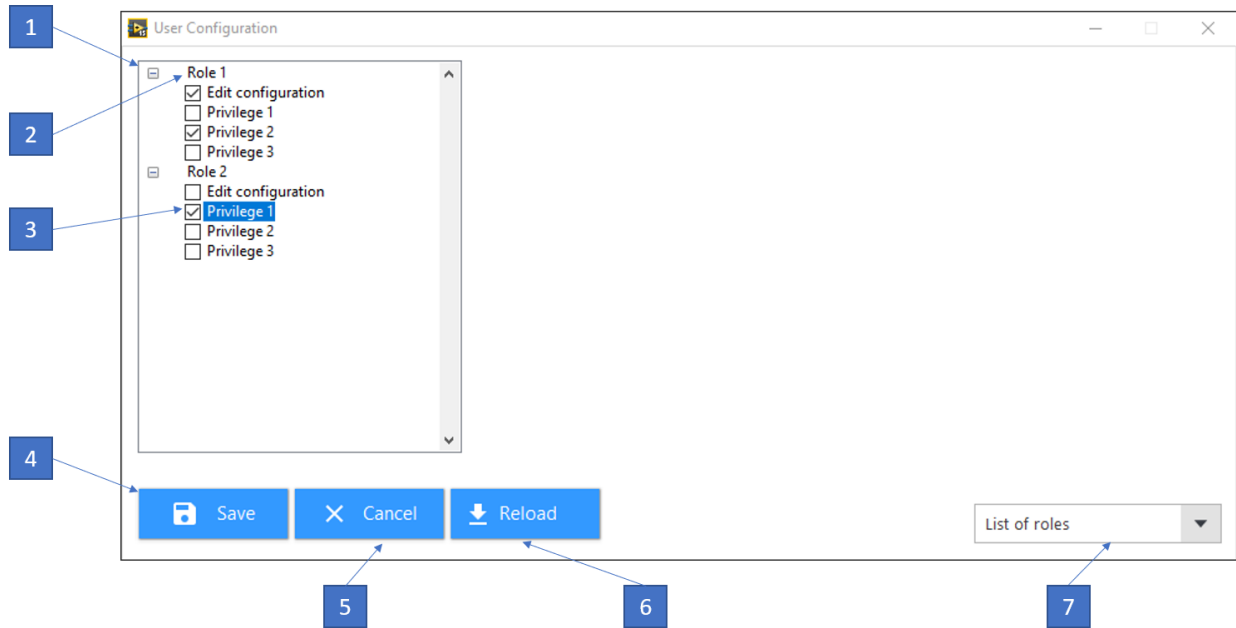
UI for entering user data.

1. Username – this field is required and needs to be unique for every user.
2. Password – optional field.
3. Role – optional field. You can select an existing role to assign it to a user.
4. Notes – optional field, can be used to leave notes for a user.
5. Cancel button – discards performed changes to a user.
6. Confirm button – finishes editing user, accepts performed changes.
7. Privileges list – indicates privileges which current user possesses. Left click a privilege to select/deselect it.

Configuration UI for managing user roles

Features:

- Add role – right-click menu option, allows to add a new role to the list.
- Remove role – right-click menu option, available when clicked on an existing role. This option removes selected role from configuration.
- Rename role – left clicking on a role allows you to rename selected role. Note that role names must be unique.
- Add/remove privilege – adds/removes selected privilege from a role. This is performed by checking/unchecking box left to the selected privilege.



Configuration UI for managing user roles.

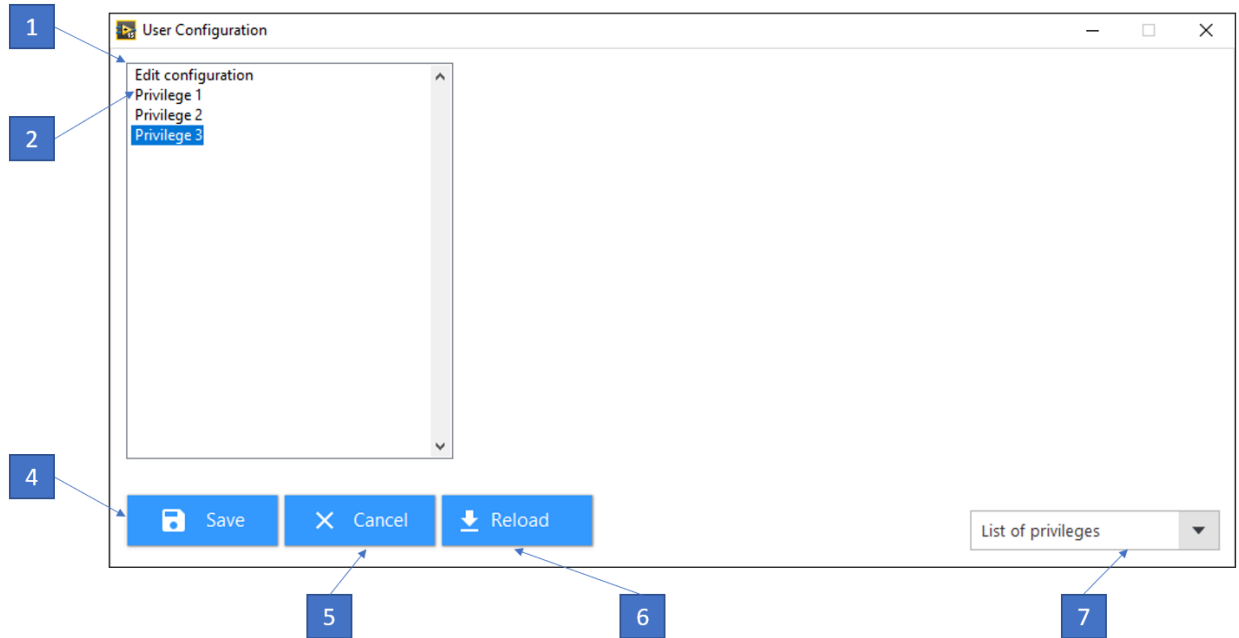
1. List of roles and privileges assigned to them.
2. Specific role in the list.
3. Privilege, which can be either assigned or unassigned to a role by left-clicking the privilege.
4. Save button – saves configuration to data storage.
5. Cancel button – discards performed changes, closes configuration window.
6. Reload button – discards performed changes, reloads configuration from data storage.
7. Change window – changes displayed window between Configuration UIs for managing users, privileges or roles.

UI for management of user roles is a list with all the roles available in the configuration paired with all privileges. It allows you to add and remove roles and select which privileges specific role should possess.

Configuration UI for managing privileges

Features:

1. Add privilege – right-click menu option, allows to add a new privilege to the list.
2. Remove privilege – right-click menu option, available when clicked on an existing role. This option removes selected privilege from configuration. Default privilege “Edit configuration” cannot be removed.
3. Rename privilege – left clicking on a privilege allows you to rename selected privilege. Note that privilege names must be unique. Default privilege “Edit configuration” cannot be renamed.



Configuration UI for managing privileges.

1. List of privileges.
2. Specific privilege o the list.
3. Save button – saves configuration to data storage.
4. Cancel button – discards performed changes, closes configuration window.
5. Reload button – discards performed changes, reloads configuration from data storage.
6. Change window – changes displayed window between Configuration UIs for managing users, privileges or roles.

UI for management of privileges is, similarly to UI for management of roles a list, which possesses all available privileges. You can add and remove new privileges here. By default, “Edit configuration” privilege is present. It allows users to edit configuration in distributed applications. This privilege cannot be deleted or modified.

User Management Basics

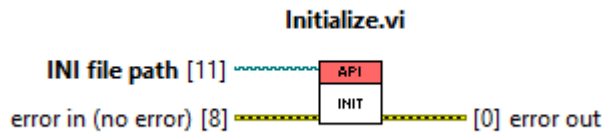
User Management Toolkit allows developers to easily manage users in their applications. In this chapter, we describe fundamental terminology and concepts of the toolkit.

User management is based on creation of users with different rights, which each user can possess. UMT uses name “privilege” for rights used by the application. Each user can have several rights assigned at the same time. Furthermore, there are “roles”, which act as set of multiple privileges. Each user can have a specific role, which gives him all privileges that the role contains. Roles allow to group users based on common privileges that users possess. This means, that there are two ways to assign a privilege to a user – implicitly by directly assigning privilege, or explicitly through a role.

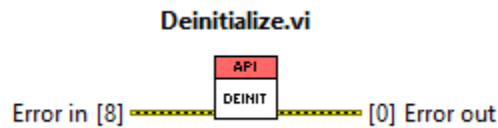
Toolkit API

Toolkit API is used to programmatically translate VIs with existing translation configuration. After toolkit installation, API can be used from LabVIEW Functions palette located at ANV/ANV User Management Toolkit. Following VIs are available to user, grouped by subpalettes:

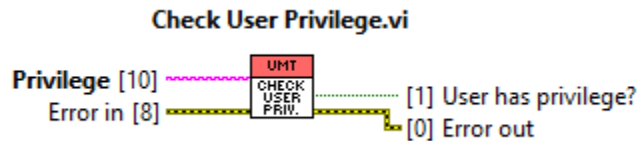
Main palette



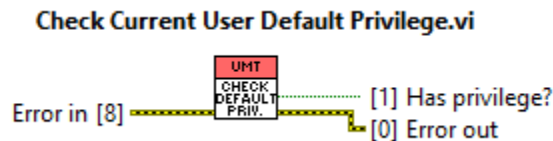
This VI initializes necessary API references. It needs to be used before any other API VI. "INI file path" is path of INI file for user management configuration.



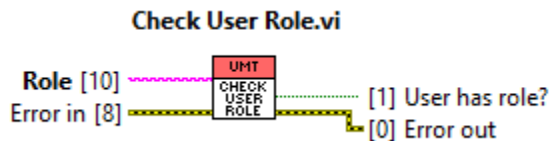
This VI deinitializes API for User Management Toolkit. This should be used at the end of application to properly close references for login events, logout user etc.



This VI checks, whether currently logged user possesses wired privilege.

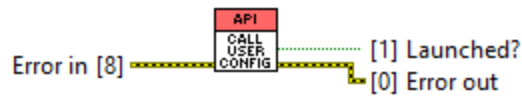


This VI checks, whether currently logged user possesses default privilege.



This VI checks, whether currently logged user has the wired role.

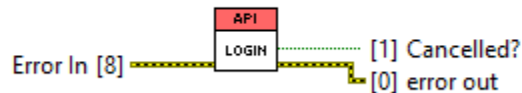
Call Users Configuration UI.vi



This VI checks currently logged user and if he possesses privilege to edit configurations, configuration window for user management is launched.

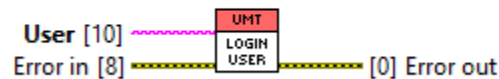
Login subpalette

Login UI.vi



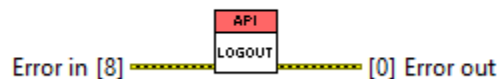
Opens UI for user login when called. After confirmation, entered data are compared with the data from user management toolkit storage and if match is found, login event for selected user is generated.

Login User.vi



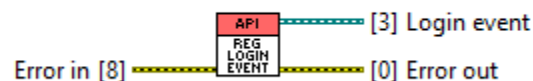
This VI generates login event with data "login" and name of user which is being logged in. If you wish to handle this event, "Register Login Event" VI should be used to register user event to dynamic event terminal of selected event structure.

Logout User.vi



This VI generates login event with data "logout" and the name of currently logged user. If you wish to handle this event, "Register Login Event" VI should be used to register user event to dynamic event terminal of selected event structure.

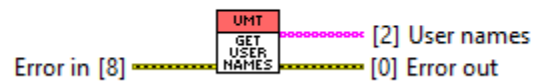
Register Login Event.vi



Registers login user event to an event structure. Wire output of this VI to dynamic terminal of event structure where you wish to handle login events.

Data subpalette

Get All Users.vi



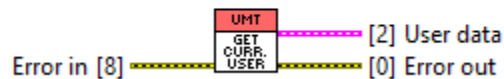
This VI outputs all user names from the configuration storage.

Get All Users Data.vi



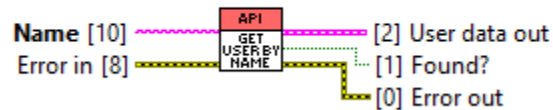
This VI gets all data related to all users from configuration storage.

Get Current User Data.vi



This VI gets user data for current user.

Get User by Name.vi



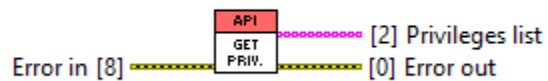
This VI gets user data for a user from configuration storage.

Get List of Roles.vi



This VI outputs list of all roles from configuration storage.

Get Privileges.vi



This VI gets all privileges available from configuration storage.

This VI outputs all user names from the configuration storage.

Get Privileges for Role.vi



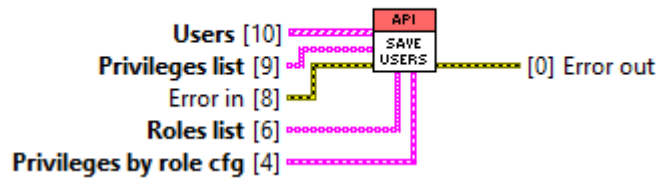
This VI checks selected role in configuration storage and outputs all privileges belonging to this role.

Get Roles by Privilege.vi



This VI checks selected privilege in configuration storage and outputs all roles, which contain given privilege.

Save Data.vi



This VI takes all configuration related data (list of users, privileges, roles and cluster of role - privilege list pairs) and writes them to configuration storage.

Sample Project

Sample project is provided for the toolkit, to demonstrate API usage to user. Sample project is available from among LabVIEW examples after installing toolkit for selected LabVIEW version. In order to use the project, open User Management Toolkit Example.vi in the project and follow front panel instructions.

Configuration Files

This section contains information about configuration files used by ANV User Management Toolkit. Purpose of configuration files is to store data related to settings and translation created by user.

Configuration Scope

Configuration relates to project. Each project requires its own configuration.

Configuration File Types

- a. UMT Storage Cfg.ini – this configuration file is used to store storages configuration: selected storage type, and its specific configuration.

Section Name	Description
General Storage Configuration	Selected storage type
JSON Storage Configuration	JSON storage type configuration
Binary Storage Configuration	Binary storage type configuration
SQLite Storage	SQLite storage type configuration
MS SQL Storage	MS SQL storage type configuration

- b. UMT paths.ini – this configuration file contains configuration of project path vs folder location, which contains storage configuration file.

Configuration Files Location

File name	Default location
UMT paths.ini	C:\Program Files (x86)\National Instruments\LabVIEW XXXX\vi.lib\ANV\ANV User Management Toolkit\Configuration Files
UMT Storage Cfg.ini	<Project folder path>\UMT

Location of “UMT paths.ini” file is fixed.

Location of “UMT Storage Cfg.ini” file is changeable.

Configuration Files Content

Configuration files have stored file paths values in special format, using “path mask”. It means, that file path value in configuration file could be (but not must be) written as <path mask><specific part of file path>. Path mask allows to match different locations on different PCs. For example for Windows “Admin” user, path to Documents system folder is “C:\Users\Admin\Documents”, for “Operator” user this path is “C:\Users\Operator\Documents”. Path mask allows to store these paths in configuration file as “%User Documents%”, and software will load specific file path based on user logged to the system. Below is list of path masks and their description.

Such approach allows to transfer configuration files along with storages to different target machines, without configuration file editing as long as file path mask is the same.

For example, in development environment storages are located in folder “<Project Folder>\UMT Cfg”. And storage will be saved as “%Project Path%\UMT Cfg\storage.json” in the configuration file.

While building executable, folder “UMT Cfg” could be added as “Always Included”, so it will be located along with executable file.

Then, when application (running as executable) will read configuration file path and based on mask “%Project Path%” will use “<Exe Folder>\UMT Cfg”.

Path mask	Description
%Project Path%	Path to project's root folder
%User Desktop%	C:\Users\ <i>user name</i> \Desktop
%User Documents%	C:\Users\ <i>user name</i> \Documents
%User Application Data%	C:\Users\ <i>user name</i> \AppData\Local
%Public Documents%	C:\Users\Public\Documents
%Public Application Data%	C:\ProgramData
%Application Files%	C:\Program Files (x86)
%User Home%	C:\Users\ <i>user name</i>