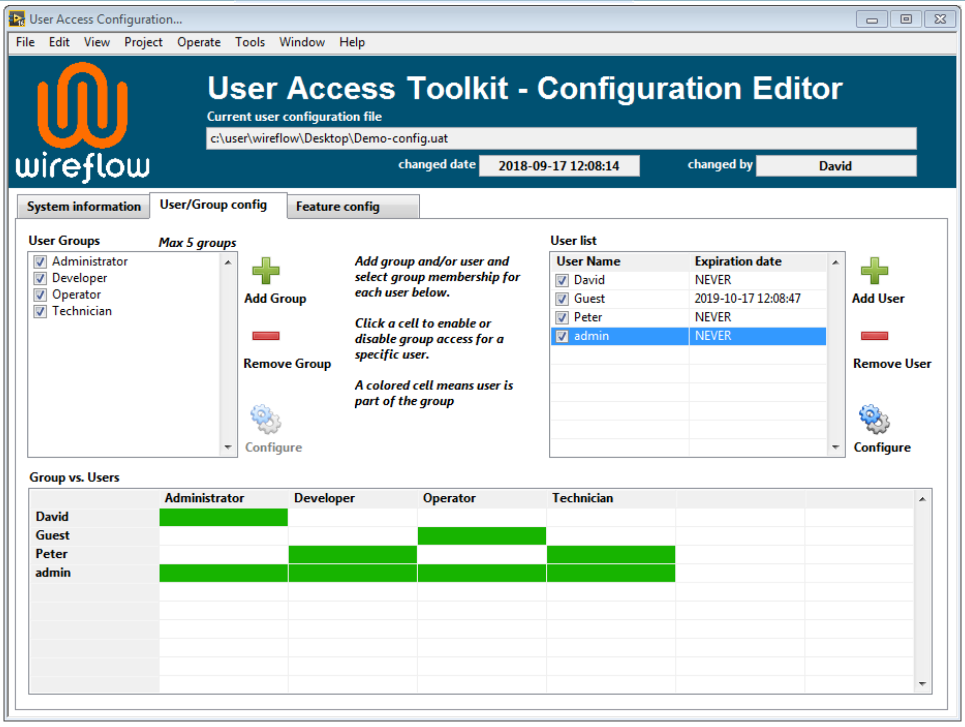
AC0062-002 Rev D

User Access Toolkit User Manual



User Access Configuration...

User Access Toolkit - Configuration Editor

Current user configuration file: c:\user\wireflow\Desktop\Demo-config.uat

changed date: 2018-09-17 12:08:14 changed by: David

User Groups *Max 5 groups*

- ☑ Administrator
- ☑ Developer
- ☑ Operator
- ☑ Technician

Add group and/or user and select group membership for each user below.

Click a cell to enable or disable group access for a specific user.

A colored cell means user is part of the group

User list

User Name	Expiration date
☑ David	NEVER
☑ Guest	2019-10-17 12:08:47
☑ Peter	NEVER
☑ admin	NEVER

Group vs. Users

	Administrator	Developer	Operator	Technician
David	🟩			
Guest			🟩	
Peter		🟩		🟩
admin	🟩	🟩	🟩	🟩

Contents

Support information

Technical support and Product information

www.wireflow.se

https://www.wireflow.se/product/wf-user-access-toolkit/

support@wireflow.se

WireFlow headquarters

WireFlow AB
Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

Please see appendix “Technical support and Services” for more information.

Important information

Copyright

The User Access Toolkit is Copyright © 2014, WireFlow AB.

Install VI Package

The VI package requires the program VI Package Manager (VIPM) from JKI to be installed. Once this is installed, just double-click the .vip-file and installation will be opened in VIPM.

If the installation fails, please try to run VIPM/LabVIEW with administrative rights.

Requirements

The toolkit can be used with standard login dialogs, but can also be used with WireFlow security dongles or WireFlow fingerprint readers. In this case the following hardware requirements apply

- WF2007 or WF20018 for security dongle usage
- WF2111 for fingerprint reader usage

In addition to the optional hardware, the toolkit requires the following software to be used

- Windows (>= XP)
- LabVIEW version 2012 or later
- Third Party Licensing & Activation Toolkit (TPLAT)
    - This NI toolkit is needed to correctly register and license the toolkit
- Driver for WF2111
    - If fingerprint reader support is needed
    - Can be installed automatically as a dependency from VIPM
- Driver for WF2007/WF2008
    - If WF security dongle support is needed
    - Can be installed automatically as a dependency from VIPM
- NI-VISA with USB passport support
    - Only needed if fingerprint reader and/or security dongles will be used.

For more information regarding software requirements for dongles or fingerprint readers, please see the documentation for the specific hardware. This information can be found at

https://www.wireflow.se/product/wf-2111-usb-fingerprint-reader-for-labview/

and

https://www.wireflow.se/product/wf-2008-usb-security-dongle-for-labview/

EULA

END-USER LICENSE AGREEMENT FOR

WireFlow Security Suite: User Access Toolkit (AC0062)

IMPORTANT PLEASE READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE CONTINUING WITH THIS PROGRAM DOWNLOAD/INSTALL: WireFlow's End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and WireFlow, for the WireFlow software product(s) identified above which may include associated software components, media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this EULA. This license agreement represents the entire agreement concerning the program between you and WireFlow, (referred to as "licenser"), and it supersedes any prior proposal, representation, or understanding between the parties. If you do not agree to the terms of this EULA, do not download, install or use the SOFTWARE PRODUCT.

The SOFTWARE PRODUCT is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. The SOFTWARE PRODUCT is licensed, not sold.

1 GRANT OF LICENSE

The SOFTWARE PRODUCT is licensed as follows:

1.1 Installation and Use

WireFlow grants you a personal, non-transferable and non-exclusive right to use the copy of the Software provided with this EULA on your computer running a validly licensed copy of the operating system for which the SOFTWARE PRODUCT was designed.

1.2 Backup Copies

You may also make copies of the SOFTWARE PRODUCT as may be necessary for backup and archival purposes.

1.3 Evaluation Version

For clarity in the case of Trial Licenses, if You do not pay the applicable license fees prior to the conclusion of any applicable Trial Period, you have no right or license, express or implied, to further use the SOFTWARE PRODUCT in any manner thereafter.

2 DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS

2.1 Maintenance of Copyright Notices

You must not remove or alter any copyright notices on any and all copies of the SOFTWARE PRODUCT.

2.2 Distribution

You may not distribute registered copies of the SOFTWARE PRODUCT to third parties. Evaluation versions available for download from WireFlow's websites may be freely distributed.

2.3 Prohibition on Reverse Engineering, Decompilation, and Disassembly

You may not reverse engineer, decompile, or disassemble the SOFTWARE PRODUCT, except and only to the extent that such activity is expressly permitted by applicable law notwithstanding this limitation.

2.4 Rental

You may not rent, lease, or lend the SOFTWARE PRODUCT.

2.5 Support Services

WireFlow may provide you with support services related to the SOFTWARE PRODUCT ("Support Services"). Any supplemental software code provided to you as part of the Support Services shall be considered part of the SOFTWARE PRODUCT and subject to the terms and conditions of this EULA.

2.6 Compliance with Applicable Laws

You must comply with all applicable laws regarding use of the SOFTWARE PRODUCT.

2.7 Export Laws

The export of the SOFTWARE PRODUCT from the country of original purchase may be subject to control or restriction by applicable local law. Licensee is solely responsible for determining the existence and application of any such law to any proposed export and for obtaining any needed authorization. Licensee agrees not to export the SOFTWARE PRODUCT from any country in violation of applicable legal restrictions on such export.

3 TERMINATION

Without prejudice to any other rights, WireFlow may terminate this EULA if you fail to comply with the terms and conditions of this EULA. In such event, you must destroy all copies of the SOFTWARE PRODUCT in your possession.

4 COPYRIGHT

All title, including but not limited to copyrights, in and to the SOFTWARE PRODUCT and any copies thereof are owned by WireFlow or its suppliers. All title and intellectual property rights in and to the content which may be accessed through use of the SOFTWARE PRODUCT is the property of the respective content owner and may be protected by applicable copyright or other intellectual property laws and treaties. This EULA grants you no rights to use such content. All rights not expressly granted are reserved by WireFlow.

4.1 Third party software.

The SOFTWARE PRODUCT may include software under license from third parties ("Third Party Software" and "Third Party License"). Any Third Party Software is licensed to you subject to the terms and conditions of the corresponding Third Party License. Generally, the Third Party License is located in a separate file such as license.txt or a readme file.

5 NO WARRANTIES

WireFlow expressly disclaims any warranty for the SOFTWARE PRODUCT. The SOFTWARE PRODUCT is provided 'As Is' without any express or implied warranty of any kind, including but not limited to any warranties of merchantability, noninfringement, or fitness of a particular purpose. WireFlow does not warrant or assume responsibility for the accuracy or completeness of any information, text, graphics, links or other items contained within the SOFTWARE PRODUCT. WireFlow makes no warranties respecting any harm that may be caused by the transmission of a computer virus, worm, time bomb, logic bomb, or other such computer program. WireFlow further expressly disclaims any warranty or representation to Authorized Users or to any third party.

6 HIGH RISK ACTIVITIES

The SOFTWARE PRODUCT is not fault-tolerant and is not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems, in which the failure of the SOFTWARE PRODUCT could lead directly to death, personal injury, or severe physical or environmental damage ("High Risk Activities"). WireFlow and its suppliers specifically disclaim any express or implied warranty of fitness for High Risk Activities.

7 LIMITATION OF LIABILITY

In no event shall WireFlow be liable for any damages (including, without limitation, lost profits, business interruption, or lost information) rising out of 'Authorized Users' use of or inability to use the SOFTWARE PRODUCT, even if WireFlow has been advised of the possibility of such damages. In no event will WireFlow be liable for loss of data or for indirect, special, incidental, consequential (including lost profit), or other damages based in contract, tort or otherwise. WireFlow shall have no liability with respect to the content of the SOFTWARE PRODUCT or any part thereof, including but not limited to errors or omissions contained therein, libel, infringements of rights of publicity, privacy, trademark rights, business interruption, personal injury, loss of privacy, moral rights or the disclosure of confidential information.

8

CONTACT

All questions about this EULA shall be directed to: info@wireflow.se.
WireFlow AB
Theres Svenssons gata 10
SE-417 55 Göteborg
Sweden

Introduction

The WireFlow User Access toolkit is a LabVIEW add-on that is mainly intended to be used with the Security tokens from WireFlow, but it is also offered with a simple login-dialog module.

The general idea of the User Access Toolkit is to be able to limit or enable features depending on the current user, and to allow easy reconfiguration of the features using an easy to use configuration User Interface.

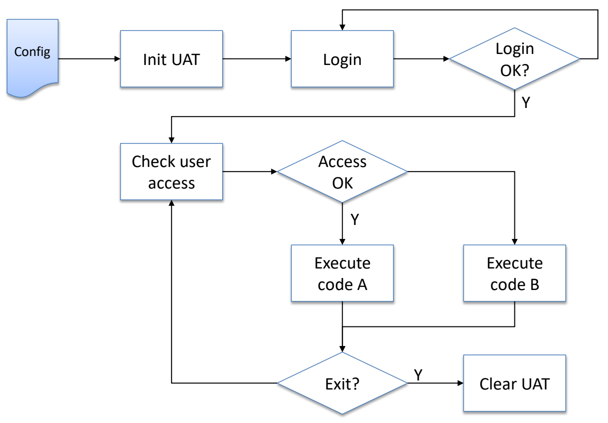


Figure 1. Basic program flow using the User Access Toolkit

The basic program flow when using the User Access Toolkit (UAT) is shown in Figure 1, where different codes are executed depending on the current user.

The general flow is to initialize the configuration from file, identify user (login dialog or through a security token), and then check access configuration to limit or enable features for the current user.

The access configuration defines a number of groups, users and optionally application features. There is always one admin user and one admin group (these cannot be removed), but we can basically add as many users or groups as we want, as long as they have unique names.

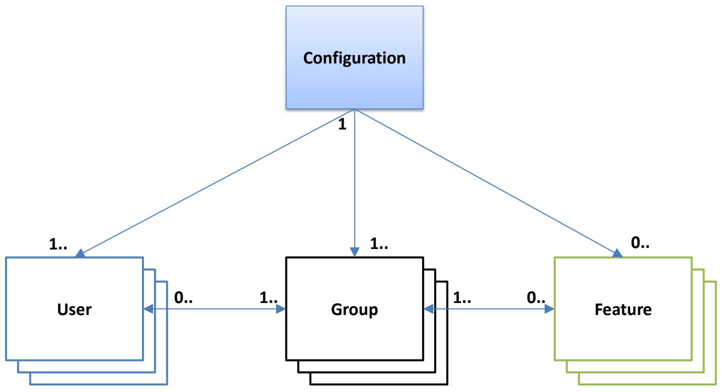


Figure 2. Basic configuration layout

A user can be defined to be part of zero or more groups, e.g. User A can be both "Operator" and "Technician", but User B might only be "Operator".

A feature is basically a string that can be active or inactive for one or more groups.

Access checks

Once a user is logged in, the access for a user can be limited by a number of different checks; the simplest check is to check if a user is a member of the admin group.

The access check types that are supported are;

- - Admin check
  - Checks if the current user is a member of the admin group
  - Mostly used to prevent main reconfiguration of the system, other setup tasks can be delegated to other groups
- - Group check
  - Checks if the user is a member of one or more of the specified groups.
  - A group check is useful if a specific part of the code is limited to one type of group, e.g. calibration panels.
- - Feature check
  - Checks if the current user is granted access to one or more specific features in the system.
  - If more than one group has access to specific features, e.g. menu items, buttons etc, it is more convenient to check feature than group. It is also easier to grant temporary access to a specific feature in the system if we don't have to check the group ownership.

User attributes

Using the user attributes an application can perform additional checks, like require the user to use a specific finger sequence for elevated security. It can also be used to read the features stored inside of a WireFlow dongle or to have users logged out automatically after a certain period of time.

Table 1. Authentication features

Attribute	**Authentication module**	**Comment**
LoginTime	All	The timestamp of the user login. Date/Time string in the format YYYY-mm-dd_HH:MM:SS
ValidationTime	All	Updated/Added when a user is validated. Same type of time string as "LoginTime"
SerialNumber	WF dongle	Hexadecimal string with the dongle serial number.
DataField1	WF dongle	Hexadecimal string with the content of the dongle DataField X. Only available if activated in the configuration and is only updated at login, not at validation.
DataField2	WF dongle	See DataField1.
DataField3	WF dongle	See DataField1.
MatchingFinger	WF fingerprint	String with the name of the matching finger
Confidence	WF fingerprint	String containing a decimal number with the confidence of last fingerprint detection
Image	WF fingerprint	Flattened string of a LabVIEW picture with the fingerprint raw image.
WeakImage	WF fingerprint	Returns the string "True" if the finger print image has low contrast. "False" otherwise

Some attributes are available regardless of the authentication method used, e.g. LoginTime and ValidationTime, but others are specific to the authentication method. In Table 1 the different attributes are listed against the different authentication methods.

Quick Start

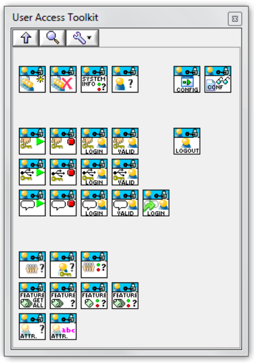
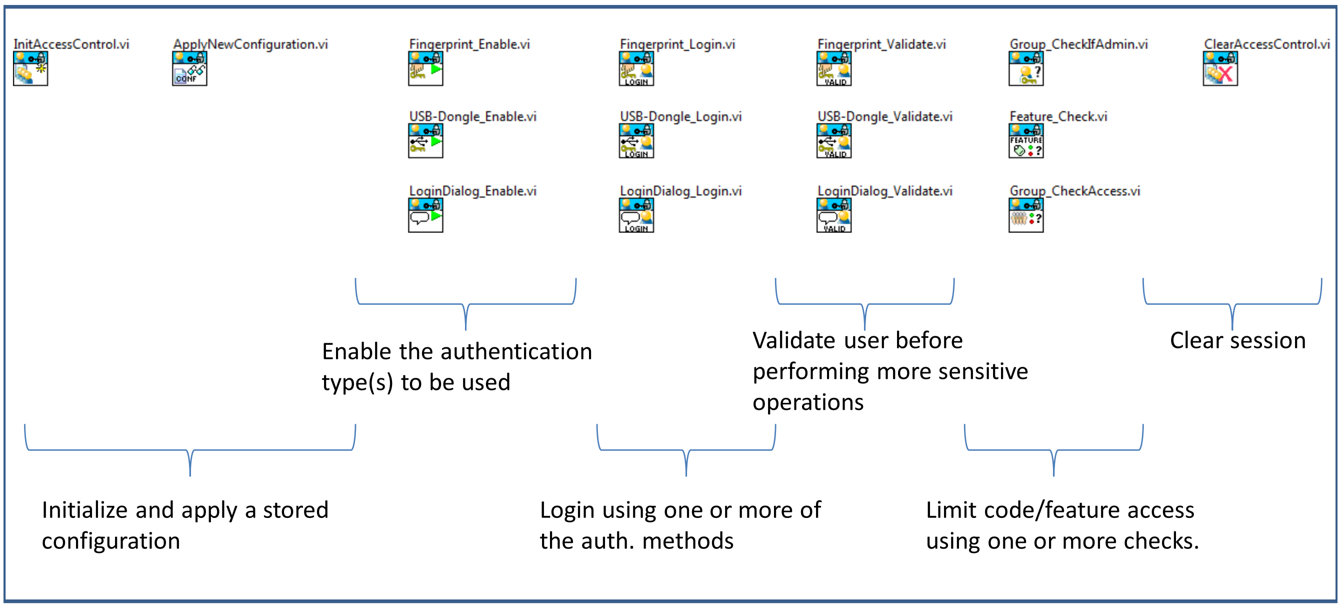


Figure 3. User Access Toolkit palette'

Once the UAT has been installed we have access to all methods in the LabVIEW palettes (for more information regarding palettes or methods, please see chapter "Toolkit VIs").

The palette is logically split in three groups of VIs, from top to bottom:

- General VIs
    - General system methods like Init, Clear, Apply Configuration etc.
    - Specific method to launch the Configuration panel, where all access parameters/user management is defined.
- Authentication VIs
    - Contains all methods necessary to authenticate using one of the supported authentication methods.
- Checks
    - Contains all necessary VIs to perform runtime checks of the access level for the current user.



Enable the authentication type(s) to be used

Validate user before performing more sensitive operations

Clear session

Initialize and apply a stored configuration

Login using one or more of the auth. methods

Limit code/feature access using one or more checks.

Figure 4. Basic usage of the toolkit

The general usage of the toolkit is displayed in Figure 4, and involves:

- Init
    - Start a session, and apply the configuration to be used
- Activation of auth. Methods
    - A session can use one or more of the different auth. types, e.g. combining dongle and login dialog for elevated security.
- Login/Validation
    - Login using the active auth. methods, and validate user for more protection of sensitive features.

- Checks
    - Once a user is authenticated, the different checks are used to grant or deny access to features in the application.
- Cleanup
    - Calling clear method cleans up all buffers and references.

Toolkit VIs

The User Access Toolkit API is located in the palette “addons/WireFlow/User Access Toolkit”

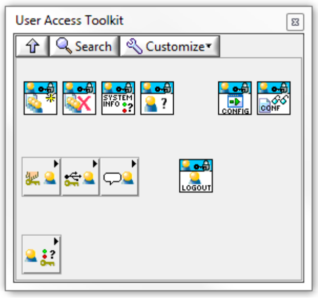


Figure 5. User Access Toolkit palette view.

The palette contains all the methods in the API, and is grouped by functionality. The different groups are described in more detail in the following sub-chapters.

General

General methods initialize, clears and returns general information for the current session.

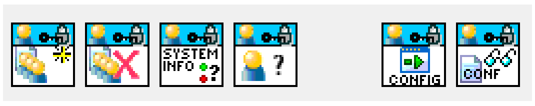


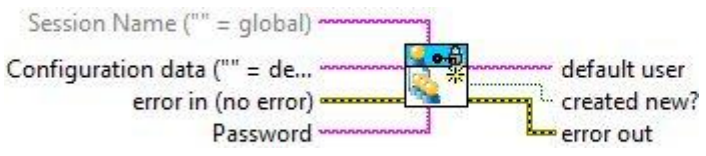
Figure 6. General functions VIs

InitAccessControl.vi

Initializes a User Access Toolkit session.

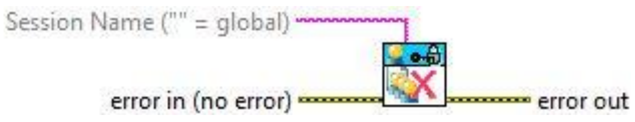
If Configuration data is specified, a valid password to decrypt the configuration data has to be supplied

N.B.: If configuration data is an empty string, the default configuration is used. It also means that admin user is assumed, and that configuration is necessary.



ClearAccessControl.vi

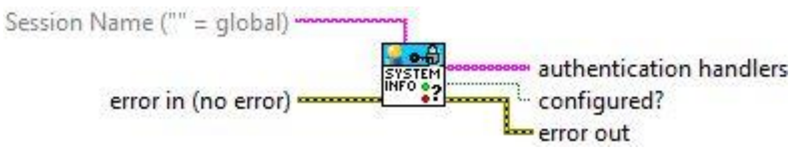
Closes the specified User Access Control session.



System_GetInfo.vi

Returns system information;

- configured = TRUE if the system is running with a valid configuration.
- authentication handlers = name of the classes that manages the authentication.

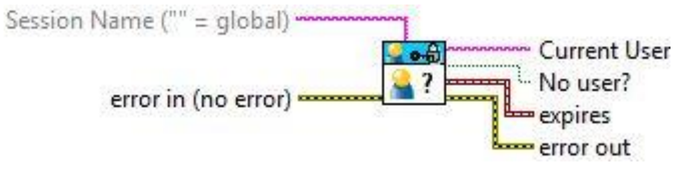


CurrentUser.vi

Returns the current user in the specified session

Outputs:

- Current user - name of the user account that is currently logged in
- No user? – if TRUE no user is currently logged in.
- expires - this is the expiration time for the user, if unlimited expires will be 0



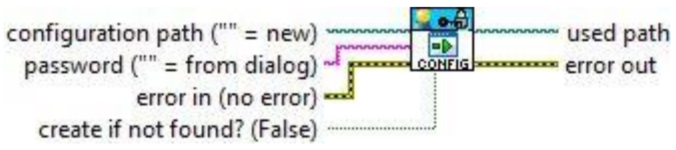
OpenConfigApplication.vi

Opens the configuration UI (currently only on Windows) and lets the user reconfigure the system. If path doesn't exist but "create if not found" is TRUE, the file is created with the specified password

The following table describes the different modes for the configuration application

	Valid path	**Bad path**	**empty path**
Valid pwd	* No password dialog * Not possible to change configuration file	* Error dialog. * Return error	* Create new config or open existing with supplied pwd * No pwd dialog
Bad pwd	* Error dialog. * Return error	* Error dialog. * Return error	* Error dialog. * Return error
Empty pwd	* Not possible to change configuration path * Prompt for pwd * If bad password, return with error	* Error dialog. * Return error	* Create new config or open existing * Pwd dialog for each new file

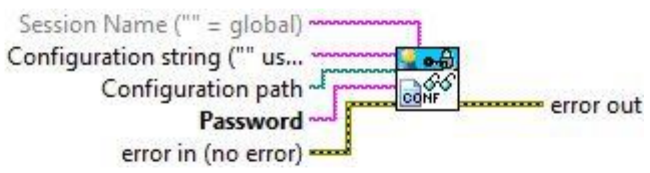
N.B. configuration files are saved encrypted, and the password used in the application must match the password used in the configuration application to be valid.


configuration path ("" = new)
password ("" = from dialog)
error in (no error)
create if not found? (False)
CONFIG
used path
error out

ApplyNewConfiguration.vi

This method applies a new configuration from either configuration string or configuration path. Configuration data is encrypted, and "Password" must be used to decrypt the data.

N.B. When a new configuration is applied, the current user is automatically logged out.



Authentication methods

These VIs handle the different authentication methods in the toolkit; enable, disable authenticate etc.

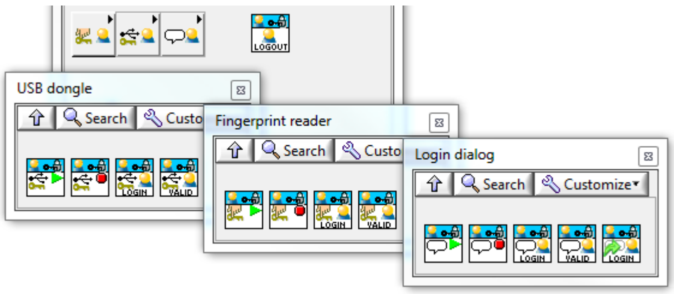
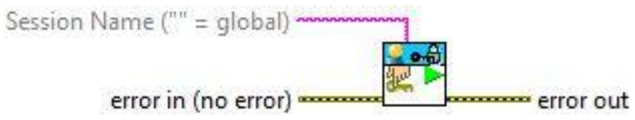


Figure 7. Authentication methods and palettes

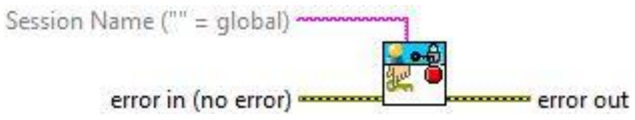
Fingerprint_Enable.vi

Enables the authentication type associated with the current VI.



Fingerprint_Disable.vi

Disables the authentication type associated with the current VI



Fingerprint_Login.vi

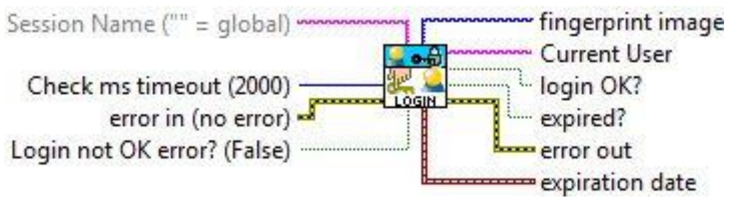
Switch user, by invoking the authentication check associated by the current VI. For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication is in place before this VI is called.

Inputs:

- Login not OK error = If TRUE and the authentication fails an error will be returned

Outputs:

- Current user = name of the user currently logged in
- login OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 ... means never)
- fingerprint image = image of last read fingerprint



Fingerprint_Validate.vi

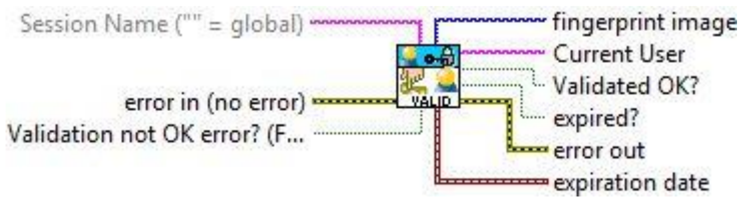
Verifies the identity of the current user, using the authentication check associated by the VI.
For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication token is in place before this VI is called.

Inputs:

- Verification not OK error = If TRUE and the authentication fails an error will be returned

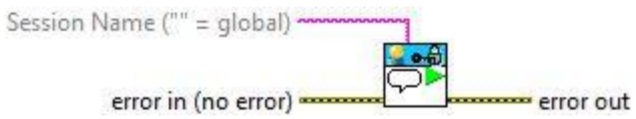
Outputs:

- Current user = name of the user currently logged in
- Validated OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 ... means never)
- fingerprint image = image of last read fingerprint



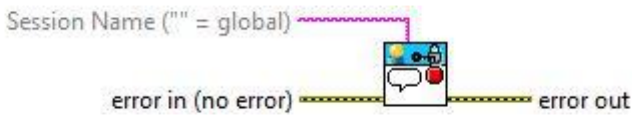
LoginDialog_Enable.vi

Enables the authentication type associated with the current VI.



LoginDialog_Disable.vi

Disables the authentication type associated with the current VI



LoginDialog_Login.vi

Switch user, by invoking the authentication check associated by the current VI.

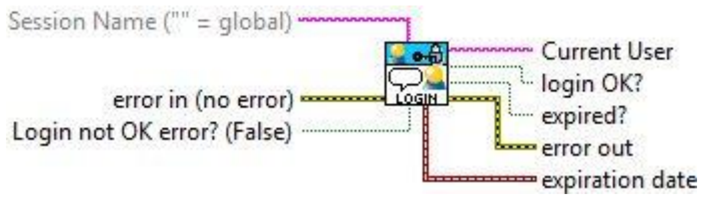
For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication is in place before this VI is called.

Inputs:

- Login not OK error = If TRUE and the authentication fails an error will be returned

Outputs:

- Current user = name of the user currently logged in
- login OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 … means never)



LoginDialog_Validate.vi

Verifies the identity of the current user, using the authentication check associated by the VI.

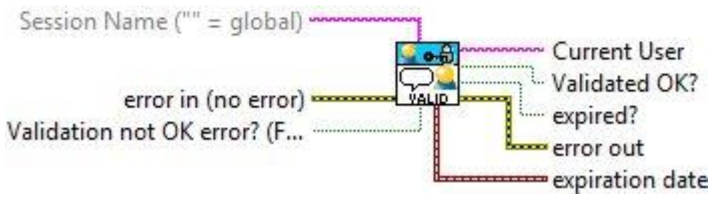
For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication token is in place before this VI is called.

Inputs:

- Verification not OK error = If TRUE and the authentication fails an error will be returned

Outputs:

- Current user = name of the user currently logged in
- Validated OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 … means never)



LoginDialog_ProgrammaticLogin.vi

Switch user by programmatic login.

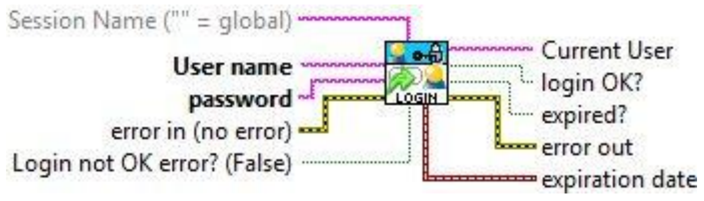
User name and password has to be specified as inputs, and then the credentials are checked like with the login dialog.

Inputs:

- Login not OK error = If TRUE and the authentication fails an error will be returned
- User name = user name to login
- password = password of the user to login

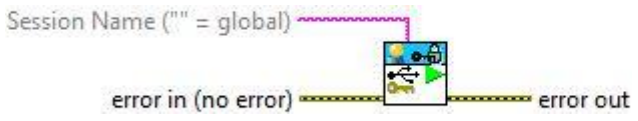
Outputs:

- Current user = name of the user currently logged in
- login OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 ... means never)



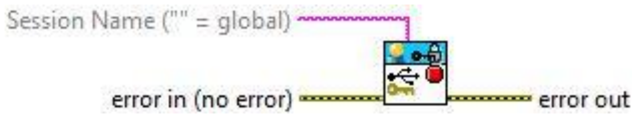
USB-Dongle_Enable.vi

Enables the authentication type associated with the current VI.



USB-Dongle_Disable.vi

Disables the authentication type associated with the current VI



USB-Dongle_Login.vi

Switch user, by invoking the authentication check associated by the current VI.

For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication is in place before this VI is called.

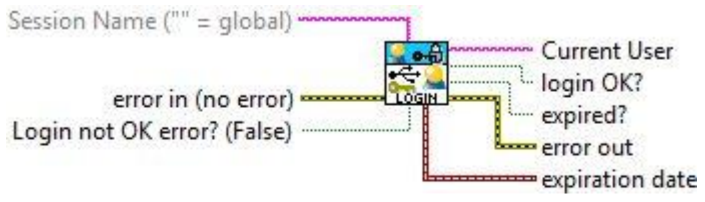
Inputs:

- Login not OK error = If TRUE and the authentication fails an error will be returned

Outputs:

- Current user = name of the user currently logged in

- login OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 ... means never)



USB-Dongle_Validate.vi

Verifies the identity of the current user, using the authentication check associated by the VI.

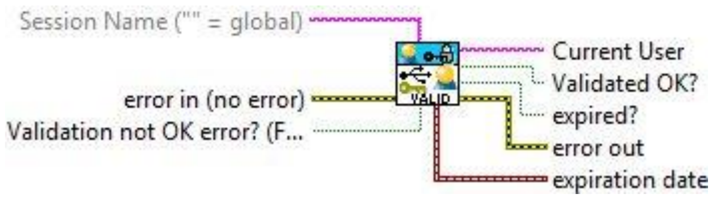
For headless authentication methods like fingerprint reader or USB dongle, it is assumed that the authentication token is in place before this VI is called.

Inputs:

- Verification not OK error = If TRUE and the authentication fails an error will be returned

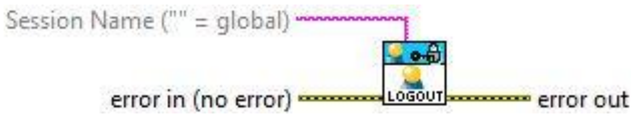
Outputs:

- Current user = name of the user currently logged in
- Validated OK? = set to TRUE if the authentication is OK
- expired = indicates if the user account has expired
- expiration date = date-time when the user account expires (00:00:00 ... means never)



Logout.vi

Logout the current user from the session.



Group access

The “group access” contains methods to check if the current user is part of the requested group or if the user is an admin user.

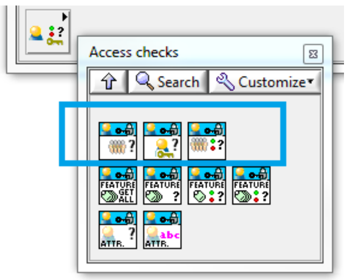
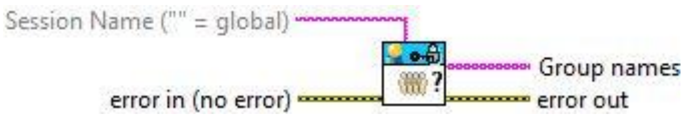


Figure 8. Group access control VIs

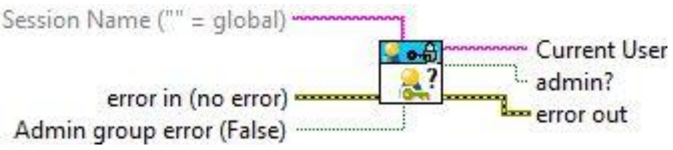
Group_GetActiveList.vi

Returns a list of active groups in the system (for all users).



Group_CheckIfAdmin.vi

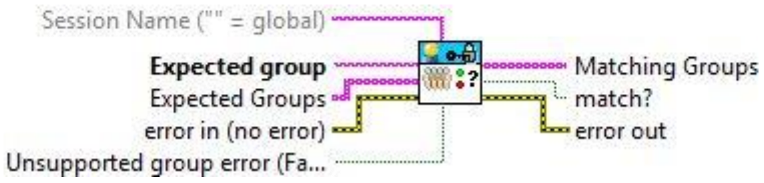
Checks if the current user is a member of an admin group.



Group_CheckAccess.vi

Checks if the current user is part of any of the requested groups, optionally returning an group access error.

If Expected groups is an empty array all groups for the current user is returned.



Feature access

Feature access contains methods to check if the current user is configured to have access to one or more specific features.

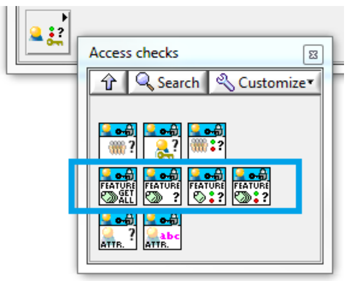
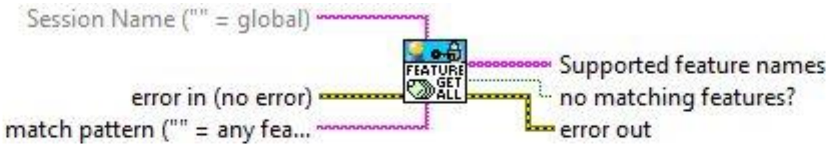


Figure 9. Feature control VIs

Feature_GetAll

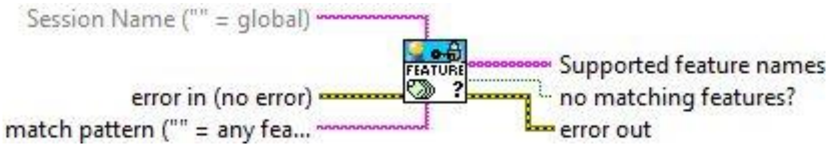
Returns a list of the currently defined features that matches the specified pattern (LabVIEW match pattern). Matching is case-insensitive.

N.B. This method returns features defined in the system, and is not bound to the current user.



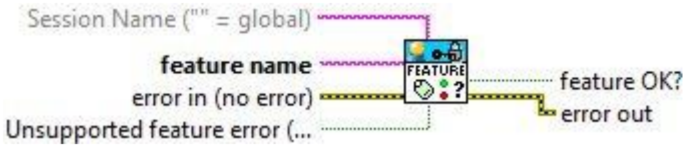
Feature_GetActive.vi

Returns a list of active features for the current user that matches the specified pattern (LabVIEW match pattern). Matching is case-insensitive.



Feature_Check.vi

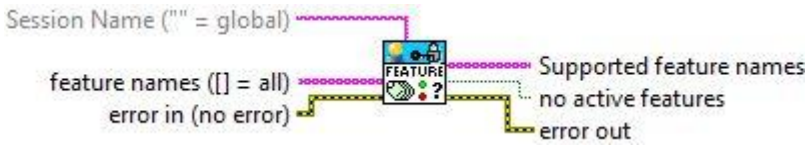
Checks if a named feature is active for the current user.



Feature_CheckMultiple.vi

Checks if the named features are active for the current user.

If feature names array is empty, all supported features are returned.



Attribute access

These VIs access the attributes for the currently logged in user. Different authentication methods have different set of attributes, see Table 1.

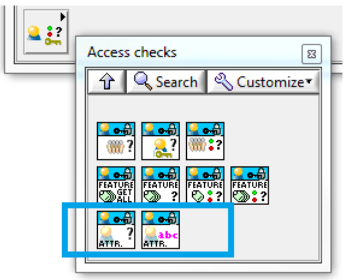
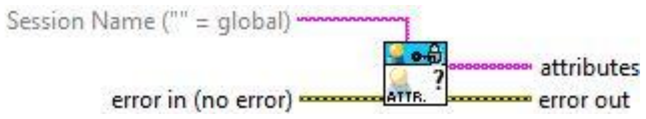


Figure 10. Attribute access VIs

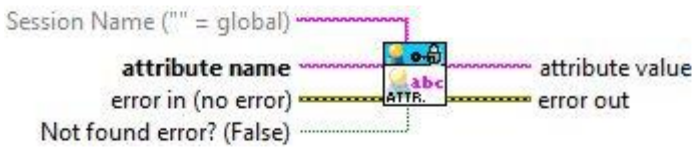
Attribute_GetNames.vi

Returns the attributes for the current user authentication



Attribute_GetValue.vi

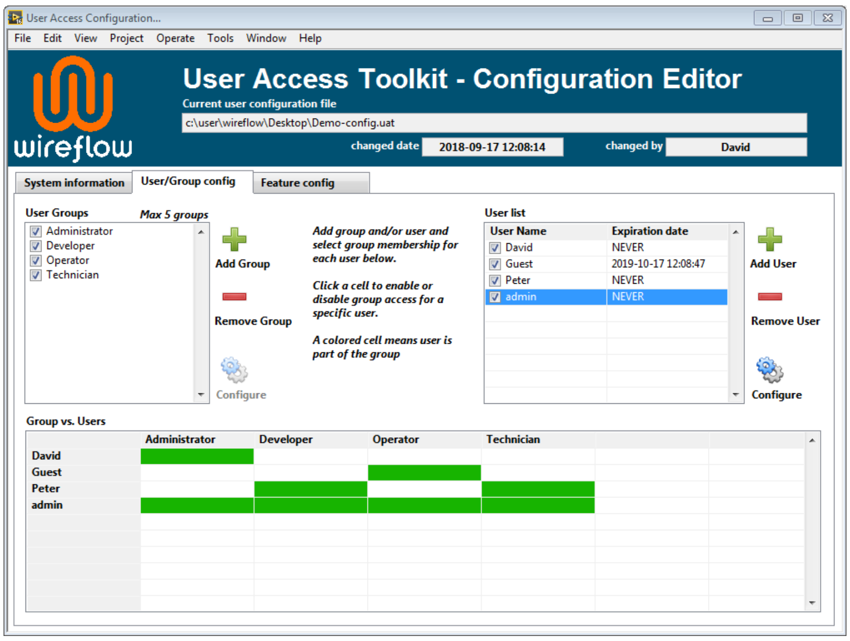
Returns the value for a named attribute of the current user, optionally returning an error if the attribute was not found.



Configuration UI

The configuration UI is a separate application, but can be invoked by the API method

OpenConfigApplication.vi (see 0). This application can edit the configuration in terms of authentication methods, users and groups.



User Access Toolkit - Configuration Editor

Current user configuration file: c:\user\wireflow\Desktop\Demo-config.uat

changed date 2018-09-17 12:08:14 changed by David

User Groups *Max 5 groups*

- ☑ Administrator
- ☑ Developer
- ☑ Operator
- ☑ Technician

Add group and/or user and select group membership for each user below.

Click a cell to enable or disable group access for a specific user.

A colored cell means user is part of the group

User list

User Name	Expiration date
☑ David	NEVER
☑ Guest	2019-10-17 12:08:47
☑ Peter	NEVER
☑ admin	NEVER

Group vs. Users

	Administrator	Developer	Operator	Technician
David	🟩			
Guest			🟩	
Peter		🟩		🟩
admin	🟩	🟩	🟩	🟩

Figure 11. The configuration UI

The configuration panel has three configuration pages

- System information
    - General system info and configuration overview
    - Import and export of configurations
- User/Group config
    - Add, remove, enable or disable users
    - Configure expiration and credentials for users.
    - Add, remove, enable or disable groups
    - Configure user group access
- Feature config
    - Add or remove features
    - Configure feature access for different groups

Using these pages it is possible to configure access for users in the system.

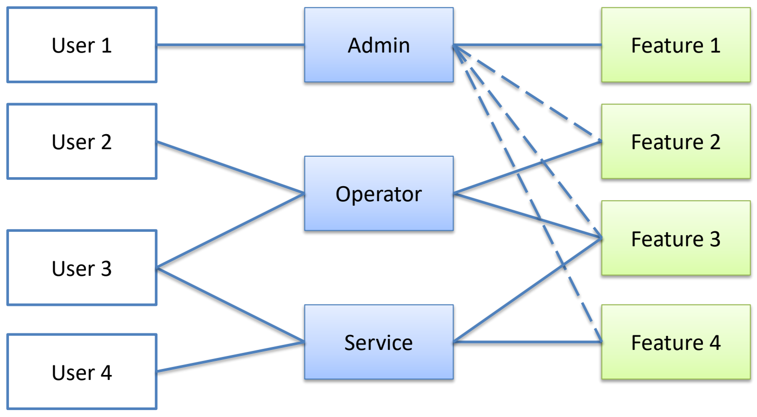


Figure 12. Schematic example configuration

The configuration outlined in Figure 12, defines three groups, 4 users and 4 features.

Each group specifies one or more features, and one feature is activated for more than one group (Feature 3). User 3 is member of both the Operator and the Service group, and gets access to the combined set of features.

If we display this configuration in a table;

Table 2. Example access configuration

User	**Supported groups**	**Supported features**
User 1	Admin	Feature 1 (all features indirectly)
User 2	Operator	Feature 2 Feature 3
User 3	Operator Service	Feature 2 Feature 3 Feature 4
User 4	Service	Feature 3 Feature 4

For most systems it might be enough to be able to check if a user is a member of a specified group, but many times it is more convenient to define features since this means that we can grant access to a BD feature for more than one group without explicitly defining all the allowed groups on the block diagram.

The menus

Opening, saving and creating new configurations is done through menu actions.

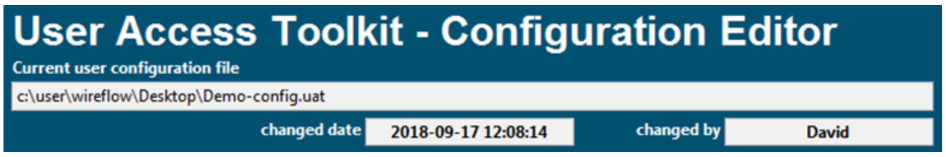
This section briefly describes the available menu selection

File:New…	Creates a new User Access Toolkit configuration from scratch (with only the root admin user at the start)
File:Open…	Opens an existing configuration, prompting the user for a password to load the file. If the password is not correct the configuration is not loaded
File:Save…	Saves the current configuration to file, prompting the user for a password to save the file. N.B. the password must match the password used in the application, otherwise the application cannot read the configuration data.
File:Save As…	Saves a copy of the current configuration to a new file, prompting the user for a password to save the file. N.B. only available if configuration UI launched with empty path.
File:Exit	Exits the application
Edit:Revert	Reverts all edits to the last loaded state
Help:Show Context Help	Opens the context help window

System information

At the top of the panel, in the WireFlow banner, general file information is displayed;

- Current user configuration file = path to the configuration currently edited.
- changed date = date/time when the configuration file was saved.
- changed by = user (in the system) that last saved the configuration


User Access Toolkit - Configuration Editor

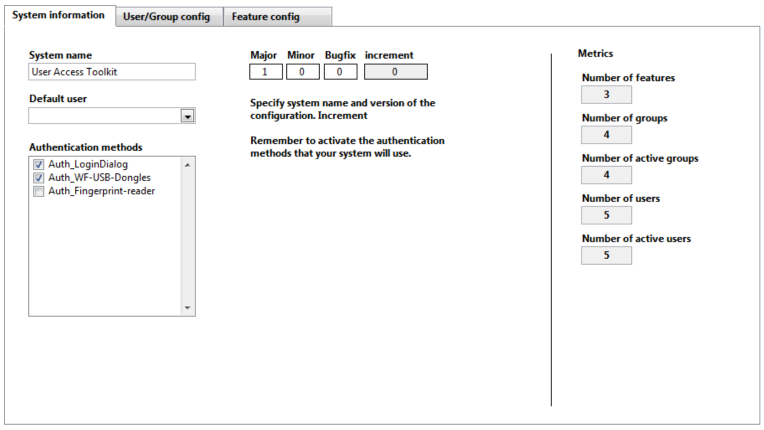
Current user configuration file

c:\user\wireflow\Desktop\Demo-config.uat

changed date 2018-09-17 12:08:14 changed by David

Figure 13. UAT WireFlow banner field

The first thing to do when creating a new configuration is to define what type of authentications the application should use.



System name: User Access Toolkit

Default user:

Authentication methods:
- ☑ Auth_LoginDialog
- ☑ Auth_WF-USB-Dongles
- ☐ Auth_Fingerprint-reader

Major	Minor	Bugfix	increment
1	0	0	0

Specify system name and version of the configuration. Increment

Remember to activate the authentication methods that your system will use.

Metrics

Number of features	3
Number of groups	4
Number of active groups	4
Number of users	5
Number of active users	5

Figure 14. System information page

When an authentication method is enabled, the checkbox to the left is checked, e.g. ☑ Auth_LoginDialog , and the authentication option is enabled in the user configuration panel (see Figure 17).

System name is only informational, but can be read by the API for information to the end user.

If default user is set, this will be the user that is activated when the configuration is applied when a session is initiated.

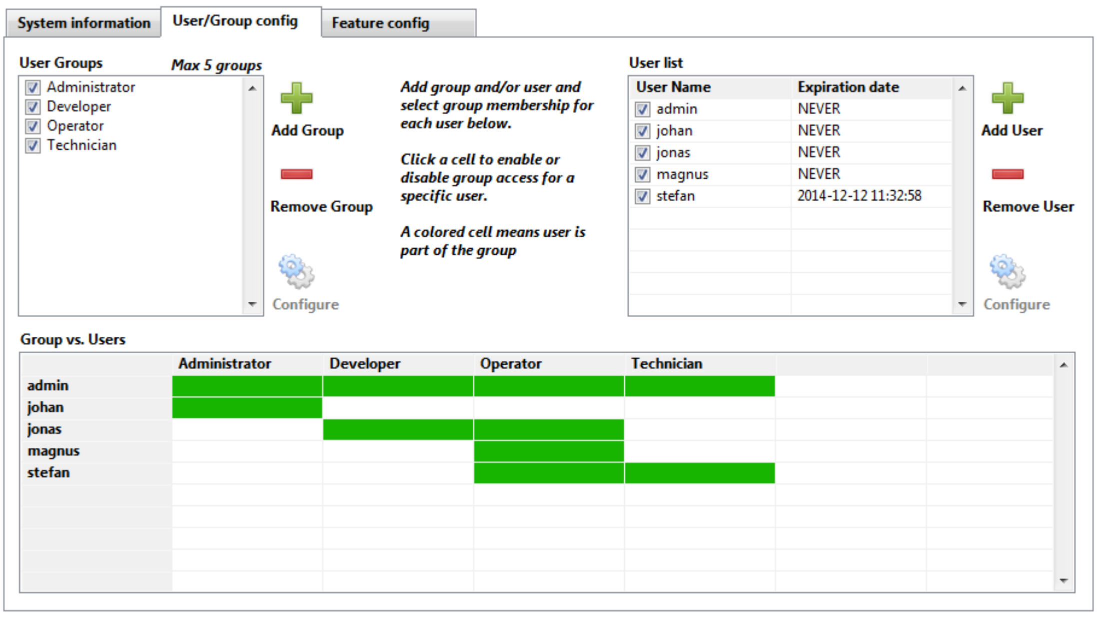
The default user should never be an admin user, and will not be activated when configuration is applied.

The right hand side of the information page shows some metrics for the system, like number of groups, users and features.

User configuration

Without a valid user configuration the UAT cannot perform login or access checks.

Configuring a user involves giving the user a unique name, set the user credentials and optionally setting an expiration date as well as group membership.



System information | User/Group config | Feature config

User Groups *Max 5 groups*

- ☑ Administrator
- ☑ Developer
- ☑ Operator
- ☑ Technician

Add Group

Remove Group

Configure

Add group and/or user and select group membership for each user below.

Click a cell to enable or disable group access for a specific user.

A colored cell means user is part of the group

User list

User Name	Expiration date
☑ admin	NEVER
☑ johan	NEVER
☑ jonas	NEVER
☑ magnus	NEVER
☑ stefan	2014-12-12 11:32:58

Add User

Remove User

Configure

Group vs. Users

	Administrator	Developer	Operator	Technician
admin	🟩	🟩	🟩	🟩
johan	🟩			
jonas		🟩	🟩	
magnus			🟩	
stefan			🟩	🟩

Figure 15. User configuration

Figure 15 shows the configuration of user and groups;

- The “User Groups” shows the groups in the system and their active state.
- The “User list” shows the current users in the system and their active state as well as the expiration date
- “Group vs. User” shows and defines the group membership for each user in the system. Groups are in columns and users are in rows.

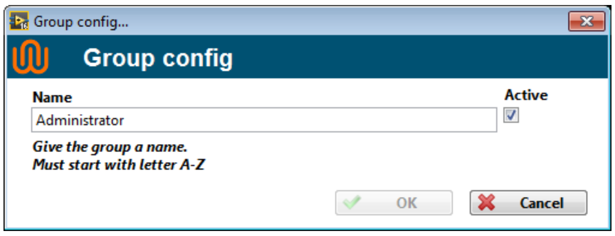
The checkboxes to the left of a user or group name can be used to disable or enable the selected item (disabling is not possible for the default admin user or admin group):

- Disabling a user means that the user cannot access the system, but that the user information is still stored in the configuration data.
- Disabling a group means that all users that are member of that group loose the access right defined for that group.

Exactly how a group/user can be configured is described in more detail in the following subchapters.

Adding a new group

To add a new group, press Add Group and give the new group a name and optionally set the group inactive.



Administrator

Give the group a name.
Must start with letter A-Z

Figure 16. Group configuration dialog.

Removing an existing group

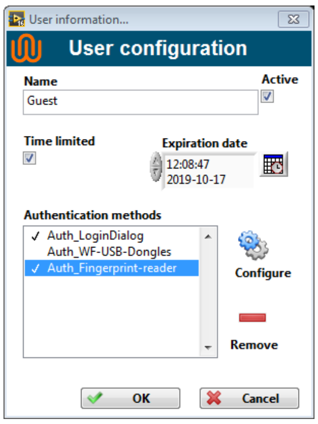
To remove an existing group, select one or more groups in the group list and press **Remove Group** and either accept or cancel the deletion. The default admin group cannot be removed.

Renaming a group

To rename a group, double-click the group name or press the Configure button next to the group list, and change the name in the Group config dialog.

Adding a new user

To add a new user, press **Add User** and fill in the information in the pop-up. A checkmark to the left of an authentication type means that the authentication is configured



Guest

12:08:47
2019-10-17

✓ Auth_LoginDialog
Auth_WF-USB-Dongles
✓ Auth_Fingerprint-reader

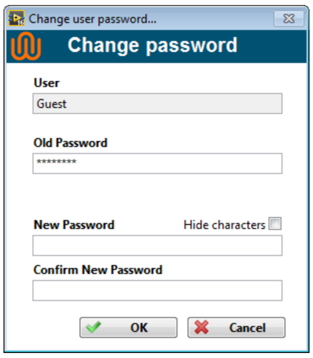
Figure 17. User configuration dialog

- Name
  - Must be a unique name and can only contain the characters: a-z, A-Z, 0-9, _-().@
- Active
  - Sets the user to be active or inactive. Can also be set from the user list
- Time limited

  - If the user account is time limited, set this checkbox and fill in the Expiration date
- Expiration date
  - Expiration date is only active if the “Time limited” checkbox is set.
- Authentication methods
  - List of the currently active authentication methods and their status. A checkmark before an authentication indicates that it is configured and ready to be used.
- Configure
  - Opens up a dialog to configure the selected authentication method.

Login dialog authentication

Login dialog is the most basic user validation. To configure a user when the login dialog session is active (activated in the System information tab), we have to specify the password.



User: Guest
Old Password: ********

Figure 18. Specifying user password for login dialog

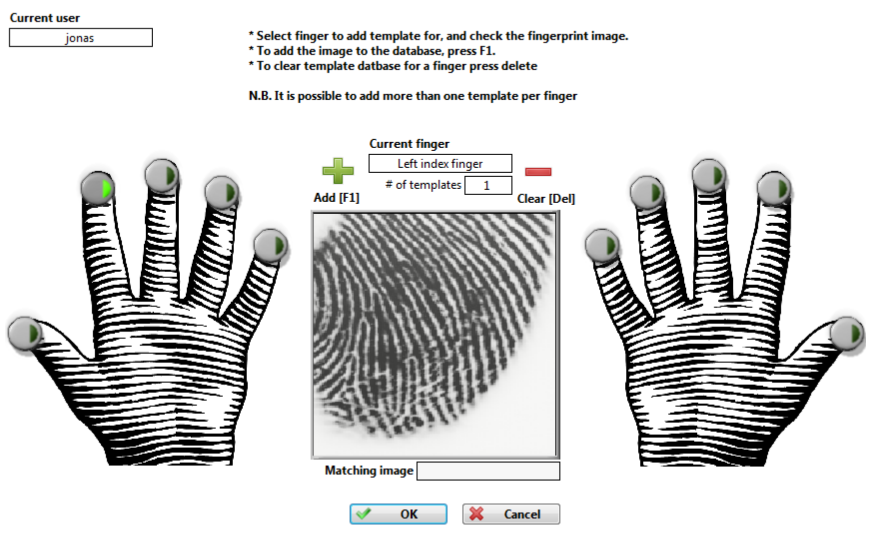
The configuration dialog we have to specify the new password, and if “Hide characters” is selected we have to specify the password twice. When the “Confirm New Password” and “New Password” matches (or if Hide characters is unchecked) the OK button is activated.

The password is hashed using the SHA-256 algorithm, and the only thing that is saved to configuration is the username and the hash. Since the SHA-256 algorithm is one-direction, it is impossible to reveal the password from the hash value.

Fingerprint reader credentials

Even if the fingerprint authentication works on many LabVIEW platforms, the configuration of the user has to be performed on a desktop version of LabVIEW. Configuration requires a connected WF fingerprint reader.

Configuring a user for fingerprint access means that we have to take a number of captures of one or more fingers.


Current user
jonas

* Select finger to add template for, and check the fingerprint image.
* To add the image to the database, press F1.
* To clear template datbase for a finger press delete

N.B. It is possible to add more than one template per finger

Current finger
Left index finger
# of templates 1

Matching image

Figure 19 Fingerprint authentication setup

The setup panel in Figure 19 is used to configure the fingerprint access for a user in the User Access Toolkit. The basic usage is;

1. Select finger to use by pressing a Boolean at one of the fingertips.
2. Once the image is good enough the Add [F1] button will be activated. Press the button (or press F1 on the keyboard) to add a new template image to the database
3. The "# of templates" counter indicates how many templates that is currently stored for the selected finger.
    a. Adding more templates can result in better image matching, but can also result in more false positives.
4. Once a finger has a template added, the finger will be automatically detected and displayed in the Matching image string indicator

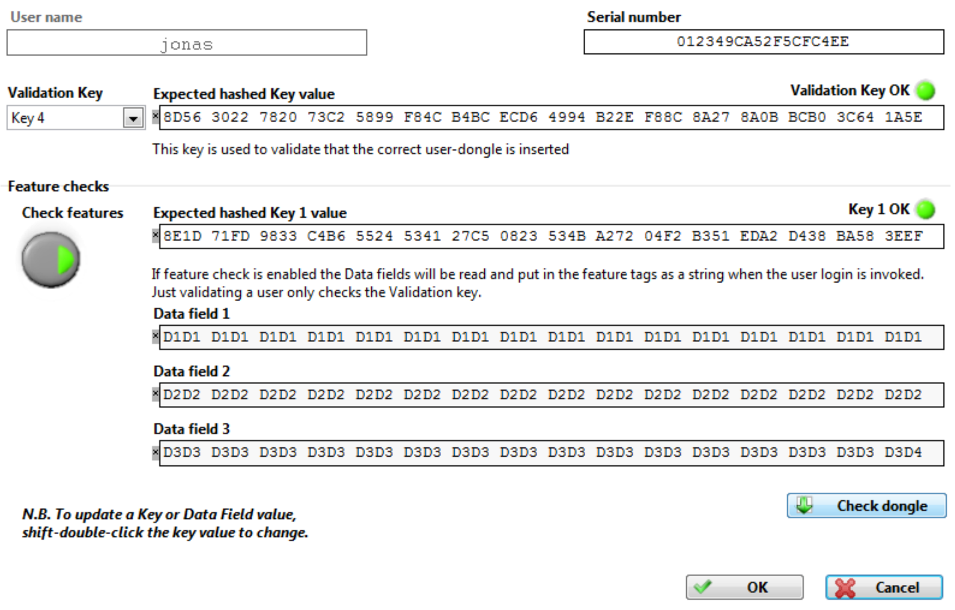
The authentication in User Access Toolkit returns the matching finger as an attribute, see 0, and this can be used to perform additional checks against a specific finger.

To delete the template(s) for a specific finger, press Clear [Del] .

WF security dongle authentication

Like the fingerprint authentication, the dongle authentication has to be configured on a desktop platform.

Since many dongles can be configured to have the same Key values, a dongle can be used to identify user group, e.g. administrators can have a specific dongle to unlock additional features, or a testsetup can have a specific dongle for operators and another for service personnel.



User name
jonas

Serial number
012349CA52F5CFC4EE

Validation Key
Key 4

Expected hashed Key value
8D56 3022 7820 73C2 5899 F84C B4BC ECD6 4994 B22E F88C 8A27 8A0B BCB0 3C64 1A5E

Validation Key OK ●

This key is used to validate that the correct user-dongle is inserted

Feature checks

Check features

Expected hashed Key 1 value
8E1D 71FD 9833 C4B6 5524 5341 27C5 0823 534B A272 04F2 B351 EDA2 D438 BA58 3EEF

Key 1 OK ●

If feature check is enabled the Data fields will be read and put in the feature tags as a string when the user login is invoked.
Just validating a user only checks the Validation key.

Data field 1
D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1 D1D1

Data field 2
D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2 D2D2

Data field 3
D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D3 D3D4

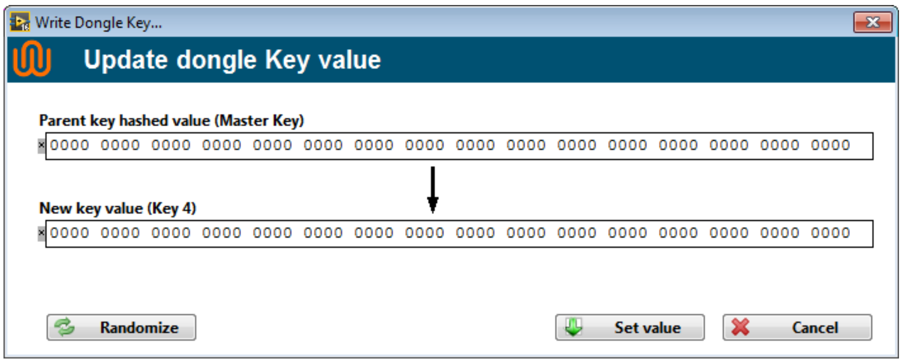
N.B. To update a Key or Data Field value,
shift-double-click the key value to change.

Figure 20 Dongle authentication setup

- Validation Key = the Key in the WF dongle that will be used to validate the user
- Expected pre-hashed Key value = the hashed value of the Validation
- Check Features = If TRUE the Key 1 Value is used to read the three data fields (after validation has passed). The values in the data fields are put in the attributes for the user.
- Data Field 1, Data Field 2 Data Field 3 = displays the current values of the data fields when "Check dongle", Check dongle, button is pressed.
- Validation Key OK and Key 1 OK are true if check dongle has been pressed and the expected key values are OK.

To change the value of a Validation Key, or a DataField,

Shift-double-click on the field, and enter the new key value and the hashed parent key (needed to allow write operation).


Write Dongle Key...

Update dongle Key value

Parent key hashed value (Master Key)

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

New key value (Key 4)

0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

Figure 21. Changing dongle key values

Press Randomize to generate a SHA-256 hashed random number that can be used to get unique dongles (different user with different dongles). If the Parent key is valid and the New key value is 32 bytes long, "Set value" will update the dongle with the new value.

Removing an existing user

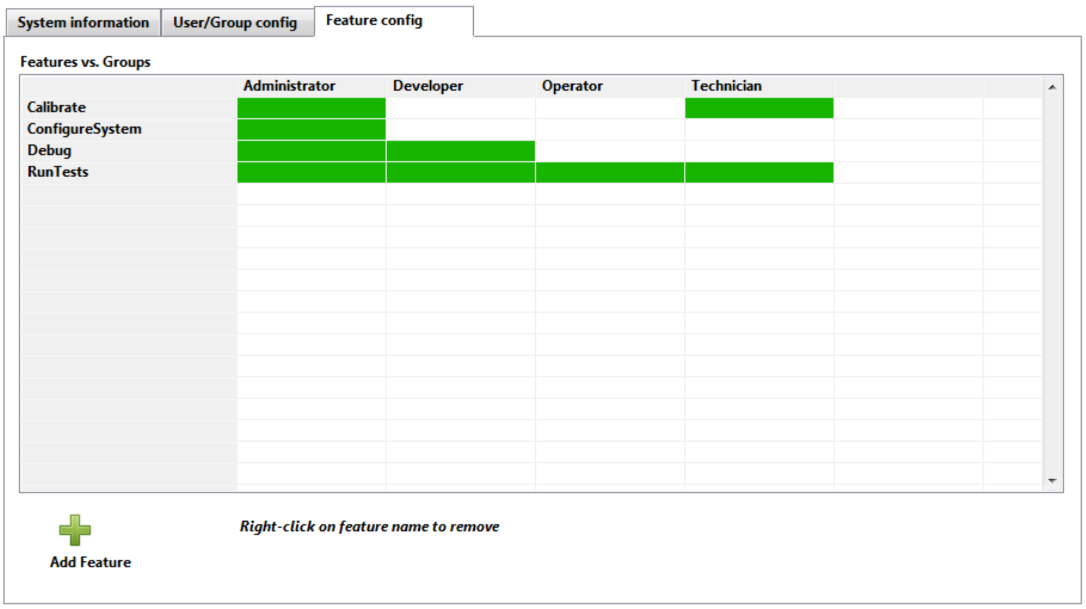
To remove an existing user, select one or more users in the user list and press Remove User and either accept or cancel the deletion. The default admin user cannot be removed.

Reconfigure an existing user.

To change the expiration date as and/or credentials or other user information, press Configure or double-click on the user name, and that will open the same dialog as when a new user is added (see Figure 17).

Feature config

The Feature config page links features to groups.



System information | User/Group config | Feature config

Features vs. Groups

	Administrator	Developer	Operator	Technician
Calibrate	🟩			🟩
ConfigureSystem	🟩			
Debug	🟩	🟩		
RunTests	🟩	🟩	🟩	🟩

Right-click on feature name to remove

Add Feature

Figure 22. Feature configuration page

Features are essentially strings that can be active or inactive for one or more groups. The list in Figure 22 shows a number of features and the groups that have access to each feature.

To enable or disable a feature for a specific group, just click on the corresponding cell in the table. Enabled features have a green cell background.

Adding features is done by pressing Add Feature , and giving the new feature a new name.

Remove a feature by, right-click on the feature and select "Remove feature..."

Basic usage / Examples

The toolkit comes with a number of examples that can be found using the LabVIEW Example finder. In the example finder, just search for WireFlow and open one of the User Access Toolkit examples.

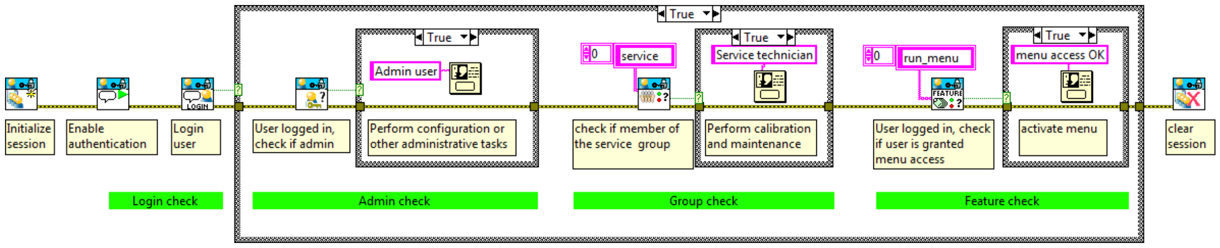

True
Admin user
service
Service technician
run_menu
menu access OK
Initialize session
Enable authentication
Login user
User logged in, check if admin
Perform configuration or other administrative tasks
check if member of the service group
Perform calibration and maintenance
User logged in, check if user is granted menu access
activate menu
clear session
Login check
Admin check
Group check
Feature check

Figure 23. Example of User Access Toolkit usage

The toolkit is created so that it should be easy to add user access control to existing applications, e.g. see Figure 23, and the basic usage is;

1. Initialize a new session
2. Enable the authentication methods that are in use
3. Wait for user to be logged in
4. Once user is logged in, use one or more of the checks to grant user access based on group, feature etc.

Error codes

The toolkit can return the following custom error codes

Code	Source text	Explanation
6400	Invalid block size for cypher!	The cypher algorithm cannot use this block size (internal error).
6401	Invalid Key size for cypher!	The cypher algorithm cannot use this key size (internal error).
6402	Unrecognised configuration format! Make sure a valid configuration is selected.	UAT is initializes with an nnrecognised configuration format! Older versions of UAT might not read new UST config versions.
6403	Invalid configuration data! Check encryption password and make sure a valid configuration is selected.	The configuration data is invalid after decryption, check password and that a valid configuration is selected.
6404	Write Key failed! Check parent key value.	Update of USB dongle key value failed.
6405	User "%s" don't exist!	Unknown user name (internal error)
6406	User access toolkit is not initialized!	UAT has to be initialized before any checks can be performed.
6407	Programmatic login not possible with authentication "%s"!	The current authentication doesn't support programmatic login (internal error)
6408	Group "%s" don't exist!	Unknown user group name (internal error)
6409	Feature name "%s" not found!	Unknown feature name (internal error)
6410	Group "%s" already exists!	The user group name already exists in configuration (internal error)
6411	Invalid feature name "%s"	The name of the feature is invalid (internal error).
6412	User "%s" already Exists!	The user name already exists in the system (internal error).
6413	Authentication failed; User: %s Expired: %s	Optional error if a login fails
6414	Feature "%s" not active for the current user "%s".	Optional error for an unsupported feature.
6415	Validation failed; User "%s" Expired: %s	Optional error if a validation fails
6416	Current user "%s" is not a member of the expected groups.	Optional error for user-group check
6417	Current user "%s" is not an administrator!	Optional error if not an admin user.
6418	Attribute "%s" was not found!	Optional error if attribute not found.
6419	Authentication type is invalid!	The specified authentication type is invalid (internal error)

Troubleshooting

Installation

During the installation progress the program folder is modified (new files are added to the <LabVIEW> directory). On some operating systems or windows installations, it might therefore be necessary to install the driver package with administrator rights.

Missing Authentication methods

If dongle or fingerprint feature is not visible in the dialog when trying to configure a user for fingerprint and/or dongle usage. Please check that the corresponding drivers are correctly installed, and then check that the authentication methods are activated in the System information tab.

Technical support and Professional services

If you need to contact support please include the following information for faster handling

- Driver version (as indicated in VIPM)
- LabVIEW version
- Target platform
- General description of the problem.

If possible, please include sample code that exemplifies the problem.

Please send support questions to support@wireflow.se, and set the subject to "Support AC0062"