User management: Update information for device manufacturers

Version: 19

Inhalt

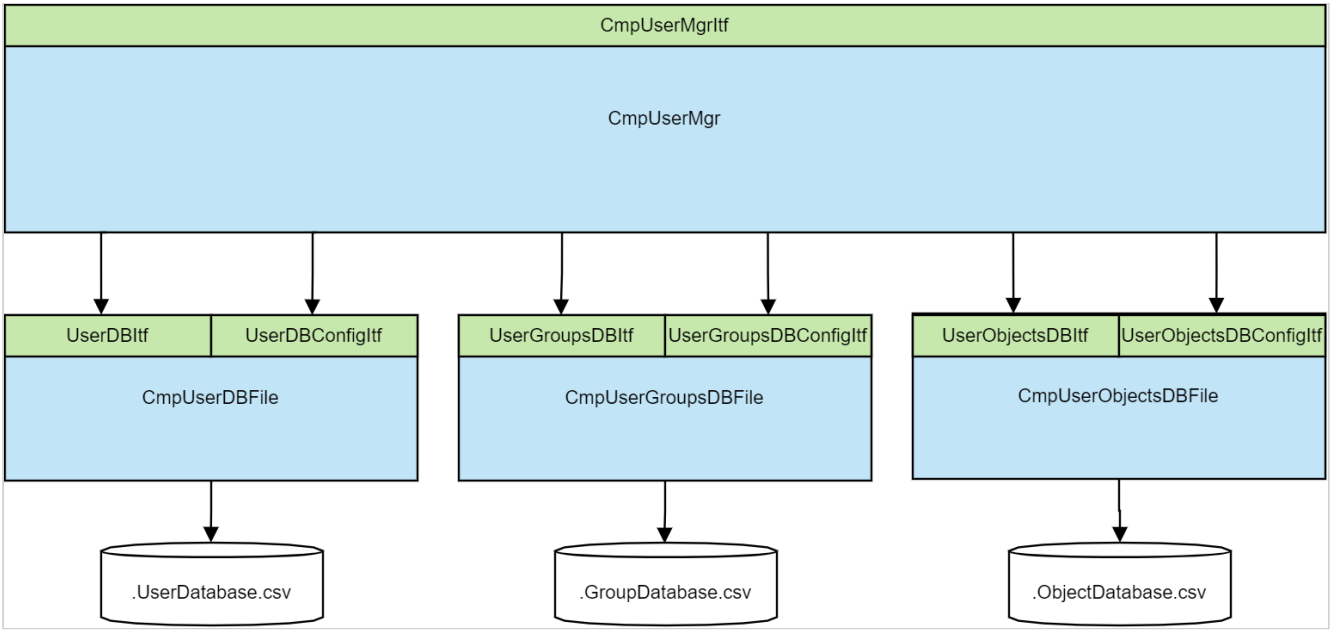
1 New component and interface structure of the user management 3
1.1 CmpUserMgr 3
1.2 Available Interfaces 3
1.2.1 CmpUserMgrItf 3
1.2.2 CmpUserDB 4
1.2.3 CmpUserDBConfig 4
1.2.4 CmpUserGroupsDB 4
1.2.5 CmpUserGroupsDBConfig 4
1.2.6 CmpUserObjectsDB 4
1.2.7 CmpUserObjectsDBConfig 4
1.3 Usage of back ends 5
1.4 Activation of the user management 5
1.5 Available back end components 5
1.5.1 CmpUserDBFile 5
1.5.2 CmpUserGroupsDBFile 5
1.5.3 CmpUserObjectsDBFile 5
1.5.4 Back end Templates 5
2 Security settings involved within the CmpUserMgr 6
2.1 SECURITY.UserLogin_AuthenticationType 6
2.2 SECURITY.UserLogin_RSAKeyLen 6
2.3 SECURITY.UserMgmtEnforce 6
2.4 SECURITY.UserMgrEditTimeout 6
3 Updating existing implementations 7
3.1 You have integrated your own user database 7
3.2 You have integrated your own user and group database 7
3.3 Your database have been configurable by CODESYS 8
4 Connecting the user management to your own user management 9

With CODESYS V3.5 SP16 the user management of the runtime system has been redesigned to fulfil current security requirements. Many parts have been implemented in a compatible way, but for security reasons it was necessary to do some incompatible changes. The fixed vulnerabilities and the user compatibility information is documented in the CODESYS Security Advisory 2019-08, which is available on the CODESYS Security web site.

This document contains additional information for device manufacturers and gives advice on updating existing custom user databases and creating your own.

1 New component and interface structure of the user management

The interfaces and components have been restructured. The existing interfaces have been split up into seven interfaces and four components. See the following picture as an overview:



The basic idea was to separate parts that are not related to each other into different interfaces which can be implemented in different ways to provide an easy way to overload some parts of the user management. The default implementation provided by us contains four components which handle the user management. These components will be explained in brief here. The interfaces used by these components are explained in a separate chapter.

1.1 CmpUserMgr

This component contains the main implementation of the user management. It implements the interfaces used by other components such as the OPC UA server. In addition this component implements the different login protocols used by the CODESYS online services. It is not recommended to replace this component as this would require a complete compatible implementation of the CmpUserMgrItf, as well as implementations of the different login protocols supported by the CODESYS online communication.

With the release of V3.5 SP16 we added a new plugin like way to extend the CODESYS login protocols with new implementations such as a X.509 based logins.

However, to support an easy adoption of the user management of different use cases (such as using the systems user management or a device manufacturer specific user management) it is possible to replace one or more databases by custom implementations. This can be done by implementing the needed interfaces and replacing the corresponding database by this implementation. The functionalities of each database was split up into an access and a configuration interface. This allows an easy implementation of static or unconfigurable back ends.

1.2 Available Interfaces

1.2.1 CmpUserMgrItf

The CmpUserMgrItf and the implementing CmpUserMgr component are now the only way to access the complete user management functionality from other components. This interface should not be replaced by a device manufacturer.

1.2.2 CmpUserDB

This small interface is the essential part for user authentication. An implementation of this interface allows to authenticate users and to list all available users. The list of users is shown in the CODESYS "User and Groups" page. This interface must be implemented to add a new back end to handle the user authentication. As the interface gets the actual credentials it does not matter how the credentials are stored, or if the credentials are forwarded to another user management component. It should be quite easy to implement this interface.

If you have your own group configuration as well and don't want to edit the users or groups from within CODESYS you can skip the implementation of GetFirstUser and GetNextUser user.

This interface is called within each login process to check if a user has entered valid credentials.

An implementation of this interface is essential for a functioning user management.

1.2.3 CmpUserDBConfig

This interface must be implemented if it should be possible to configure the user database through CODESYS online services. This interface handles all database configuration related parts such as export/import the database, deletion of all users, as well as adding and changing users and their credentials.

An implementation of this interface is optional for a functioning user management. But an implementation is needed for a configurable user management.

1.2.4 CmpUserGroupsDB

This interface provides the assignment of users to groups to the CmpUserMgr. This assignment is used to check the access rights of a specific user on a specific object. Additionally this interface provides the possibility to list all groups and group configurations to CODESYS using online services. The list of groups is used to build up the "Access Rights" page within CODESYS. The group configuration is used to fill up the group section of the "User and Groups" page within CODESYS.

An implementation of this interface is essential for a functioning user management.

1.2.5 CmpUserGroupsDBConfig

This interface must be implemented if the user groups should be configured using the CODESYS online services. It handles all database configuration related parts such as export/import the database deletion of all groups, as well as adding and changing groups and adding/removing users to/from groups.

An implementation of this interface is optional for a functioning user management. But an implementation is needed for a configurable user management.

1.2.6 CmpUserObjectsDB

This interface is used by the CmpUserMgr to check access rights to different objects using the group association of a user, as well as to upload the different objects to CODESYS in order to show and configure the access rights. As the objects used in CODESYS are highly specific to the CODESYS runtime you should select one of the provided implementations or templates of this interface. This guaranties that the object database is correctly implemented and the access right checks fulfill the needed requirements.

An implementation of this interface is essential for a functioning user management.

1.2.7 CmpUserObjectsDBConfig

This interface is needed if the runtime should support dynamic creation of objects and the configuration of access rights using CODESYS online services. If you can stick with static access rights and a small set of available objects, an implementation of this interface is not necessary.

An implementation of this interface is optional for a functioning user management. But an implementation is needed for a configurable user management.

1.3 Usage of back ends

To hide all the different back ends from access through other components than the CmpUserMgr, these interfaces are defined as DEF_CLASSITF_API and implemented using the singleton design pattern. Additionally this interface does not provide an interface function to create instances of the database. Instead this is done by the component itself. This guaranties that no other component except the CmpUserMgr can trigger calls to the back ends. To allow calls from CmpUserMgr to the back ends the CmpUserMgrItf has a set of interface functions to register the different back ends. The provided template shows how this behaviour is implemented in the back end components.

1.4 Activation of the user management

With V3.5 SP16 we also changed the way how the user management is activated. Before V3.5 SP16 there was an "Anonymous" User in the user database. This user was used if a login with empty credentials was received. With V3.5 SP16 we removed this user and the deactivated user management is completely handled within the CmpUserMgr without calls to the back ends.

The user management is activated with V3.5 SP16 as soon as the first user database back end registered itself to the CmpUserMgr. If this back end has no users available, no login will be possible. With this step all session activated with empty credentials will receive access errors on every object. After a re-login with valid credentials these sessions can be used again. CODESYS will automatically show a login dialog in this case.

1.5 Available back end components

1.5.1 CmpUserDBFile

This back end implements a user database that supports user name / password credentials and stores the database within the file system. For a secure storage of the user passwords this component uses SCRYPT to generate the password hashes. The database file is not compatible with the existing user database. However, the component imports an existing user database on start-up and converts the hashes into SCRYPT hashes. This component relies on implementations of SysFileItf and CmpCryptoItf. The component allows full configuration of users and implements the CmpUserDBItf and CmpUserDBConfigItf.

1.5.2 CmpUserGroupsDBFile

This back end implements a user group database and stores the database within the file system. This component relies on implementations of SysFileItf. The component allows full configuration of user groups and user to group association and implements the CmpUserGroupsDBItf and CMpUserGroupsDBConfigItf.

1.5.3 CmpUserObjectsDBFile

This back end implements a full object database and allows full configuration of the access rights to the different objects. This component stores the configuration into the file system. This component relies on implementations of SysFileItf and implements itself the CmpUserObjectsDBItf and CmpUserObjectsDBConfigItf.

1.5.4 Back end Templates

We provide a set of template implementations for the different interfaces. This helps to get started with your own implementation.

2 Security settings involved within the CmpUserMgr

2.1 SECURITY.UserLogin_AuthenticationType

The security setting can be used to configure the supported login protocol types for the CODESYS online protocol.

- ONLY_ASYMMETRIC: The passwords are encrypted using RSA keys. Only supported by clients >= V3.5.16.0.
- AUTO_NEGOTIATE: This will select the RSA encryption for clients >= V3.5.16.0 if the runtime supports the needed encryption algorithms. The legacy encryption is used for older clients or if the runtime is not able to provide RSA keys.
- ONLY_LEGACY: Only support the legacy login protocol.

2.2 SECURITY.UserLogin_RSAKeyLen

Use this setting to configure the size of the RSA key used to encrypt the password.

2.3 SECURITY.UserMgmtEnforce

This setting can be used to enforce the user management. If the user management is enforced a user must be created before any other service can be sent. There is no default administrator available as it is in older CODESYS runtimes. The user of the PLC has to create the first administrator user with a name and password selected by himself. The user will be guided through this process.

2.4 SECURITY.UserMgrEditTimeout

This setting forces an administrator or any other user that is allowed to configure the user management to re-authenticate, if the user was logged in, but has not taken any action while the configured time-out time. This prevents the user management from being changed if a user leaves the configuration PC for some time without logging off.

3 Updating existing implementations

If you have replaced parts of the user management by your own user management implementations, you will need to do some adaptations for the update to V3.5 SP16. What you need to do depends on what and how you have integrated your own user management.

3.1 You have integrated your own user database

This case applies if you have integrated your own user database into the CODESYS user management, but have used everything else (including the group database) in the standard configuration.

In this case you should select the CmpUserGroupsDBFile and CmpUserObjectsDBFile from the default configuration. The CmpUserDBFile will be replaced by your own implementation. You have to implement the CmpUserDBItf in your own component.

You should find it quite easy to port your existing with the following mapping:

Old CmpUserDBItf	New CmpUserDBItf	Comment
UserDBGetFirst	UserDBGetFirstUser	The behaviour of the interface changed a bit. The new implementation returns the user name instead of a user handle. To provide the iteration there is a iterator handle. The properties of a user are returned while running over the users.
UserDBGetNext	UserDBGetNextUser	
UserDBClose	Not existing	There are no open handles between the back end and the user manager. No need to close handles.
UserDBGetName	Not existing	The user name was known to the CmpUserMgr. No need to get it again from the database.
UserDBOpen	UserDBAuthenticate	The main use case of the CmpUserDBItf is the authentication of users. We decided to combine this three functions into one single interface which is used for user authentication. The interface checks both the user name and credentials and returns the users properties if the authentication was successful.
UserDBGetProperty		
UserDBCheckPassword		

3.2 You have integrated your own user and group database

This applies if you have integrated your group database in addition to the user database into the CODESYS user management. You only used the object database from the default configuration.

In this case you should use the CmpUserObjectsDBFile from the default configuration. The CmpUserDBItf and CmpUserGroupsDBItf have to be implemented in your own component or components. This depends how your system works. It is perfectly fine to implement both interfaces in a single component.

The mapping for the user database can be found in the table of the previous chapter.

See the following table for the group database mapping:

Old CmpUserDBItf	New CmpUserGroupsDBItf
UserDBGroupGetFirst	UserGroupsDBGetFirstGroup
UserDBGroupGetNext	UserGroupsDBGetNextGroup
UserDBGroupOpen	UserGroupsDBGetGroup
UserDBGroupClose	Not existing
UserDBGroupGetName	UserGroupsDBGetName

Old CmpUserDBItf	New CmpUserGroupsDBItf
UserDBGroupGetProperty	UserGroupsDBGetProperty
UserDBGroupHasUser	UserGroupsDBGroupHasUser
UserDBGroupGetFirstUser	UserGroupsDBGetFirstUser
UserDBGroupGetNextUser	UserGroupsDBGetNextUser
UserDBGroupHasMember	UserGroupsDBGroupHasMember
UserDBGroupGetFirstMember	UserGroupsDBGetFirstGroupMember
UserDBGroupGetNextMember	UserGroupsDBGetNextGroupMember

3.3 Your database have been configurable by CODESYS

There you should implement the corresponding configuration interface. The mapping should be straight forward. Please note that we removed some interface functions such as:

- UserDBGetPasswordMD5 and UserDBSetPasswordMD5 for security reasons
- The UserDBSetPassword was replaced by UserDBConfigSetUserCredentials

4 Connecting the user management to your own user management

Decide what you want to integrate into the CODESYS user management:

- Integrate a user database. The database is not configured by CODESYS.
  Implement the CmpUserDBItf
- Integrate a user database. The database should be configured by CODESYS.
  Implement the CmpUserDBItf and CmpUserDBConfigurationItf
- Integrate a group database. The database is not configured by CODESYS.
  Implement the CmpUserGroupsDBItf
- Integrate a group database. The database should be configured by CODESYS.
  Implement the CmpUserGroupsDBItf and CmpUserGroupsDBConfigurationItf.
- You want to use your own Objects database:
  Not recommended. The objects are highly CODESYS specific.