# zenon manual

## Distributed engineering

v.8.10

**COPA·DATA**

# Contents

# 1  Welcome to COPA-DATA help

## ZENON VIDEO-TUTORIALS

You can find practical examples for project configuration with zenon in our YouTube channel (https://www.copadata.com/tutorial_menu). The tutorials are grouped according to topics and give an initial insight into working with different zenon modules. All tutorials are available in English.

## GENERAL HELP

If you cannot find any information you require in this help chapter or can think of anything that you would like added, please send an email to documentation@copadata.com.

## PROJECT SUPPORT

You can receive support for any real project you may have from our Support Team, who you can contact via email at support@copadata.com.

## LICENSES AND MODULES

If you find that you need other modules or licenses, our staff will be happy to help you. Email sales@copadata.com.

# 2 Distributed engineering

The Multi-user functionality of zenon allows multiple users to work on the same project at the same time. The projects are stored on a central SQL Server (on page 7) and can be worked on from different workstations at the same time. In this case, the central storage of the project is called the **Server DB**.
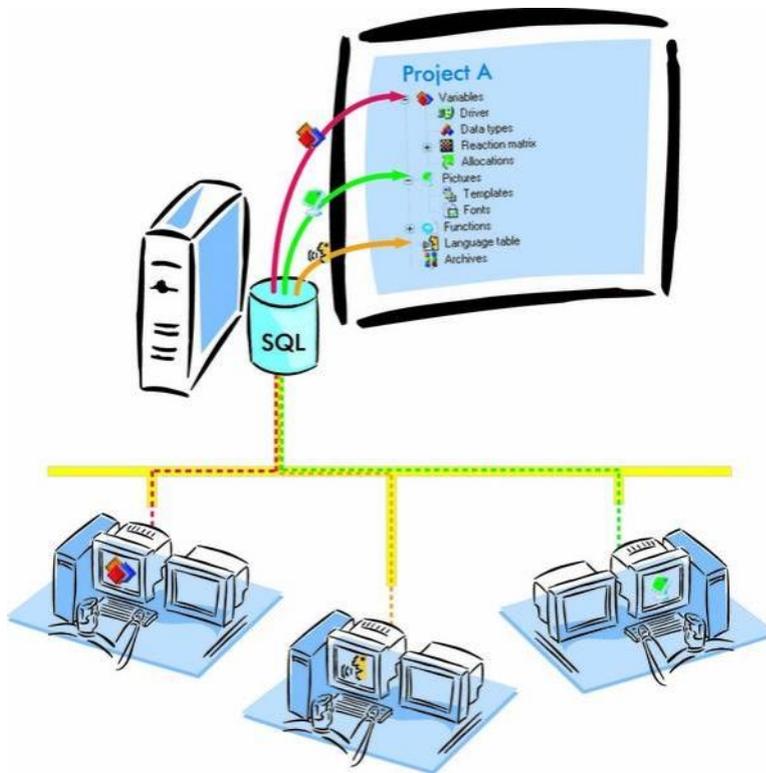


**Example:** One user can create screens while another user can define variables.

> ### ☀ Information
>
> Hint for terminology:
>
> ▸ "Check in" corresponds to the property **Accept changes**
>
> ▸ "Check out" corresponds to the property **Enable changes**

## IP PROTOCOL

The communication in the zenon network can be done via **IPv4** or **IPv6**.

You can find details in relation to this in the Network manual in the Requirements chapter.

# 3 Hardware requirements

The minimum requirements relate to the use of **multi-user projects** with a number of objects (for example >= 100,000 variables).

| Hardware | Minimum requirements |
|---|---|
| **CPU** | Quad Core |
| **RAM memory** | Server: 16GB / Client: 8GB RAM |
| **Harddisk** | SSD is recommended |

**Note:** The **Maximum Server Memory** setting of the SQL server may need to be amended manually.

## ☀ Information

You can find more information under the following link:
https://msdn.microsoft.com/en-us/library/ms178067(v=sql.120).aspx
(https://msdn.microsoft.com/en-us/library/ms178067(v=sql.120).aspx)

# 4 Information about SQL Server

## ZENON AS OF VERSION 7.00 SP0

For **distributed engineering** all participants access a central SQL Server. As of zenon version 7.00 SP0, all participants must use at least the suggested versions:

- ▶ **Editor:** as of 7.00 SP0
- ▶ **DBSrv:** as of 7.00
- ▶ **SQL Server:** SQL Server 2008 R2 Express Edition

> ⚠️ **Attention**
>
> The same password must be used on all computers.
>
> Note:
>
> ▸ The encryption is done via the **Startup Tool**.
>
> ▸ The database setting must be set using the **Startup Tool**.
>
> ▸ The password must also be amended on the SQL server for the **zenOnSrv** user.

### UPDATE

When updating the project server to a higher SQL Server version, all Clients must also be updated.

You can find out more information under additional software.

# 5 Possibilities for using distributed engineering

**Distributed engineering** can be used for:

▸ Standalone projects (on page 8)

Changes can be tried out without risk

▸ a local Server (on page 9) with several clients

▸ a network Server (on page 9) with several clients

## 5.1 Local Server

In this case the server project AND a local project copy are stored on the same PC.

A **multi-user project** can also be created on an individual computer. This makes sense if, for example, changes need to be checked in advance.

Example: You can

▸ Allow changes for a screen with **Check out**

▸ Edit the screen

▸ Check the changes in the Runtime

You can then either accept or discard changes. The original screen stays unchanged, until the changes to the screen are accepted.

## 5.2 Local server for access to several clients

The server database and a local project copy are on the same PC. The clients also work with a local project copy. Changes in the local project copy are transferred to the server database, when the changes are accepted. The local project copy is also edited on the server.
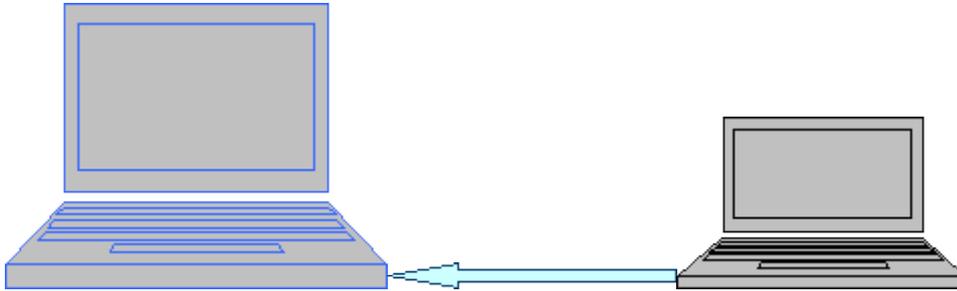


## 5.3 Network server for access by several clients

The server database is on the central computer. Multiple Clients connect to this network server. Local project copies are created on each client.
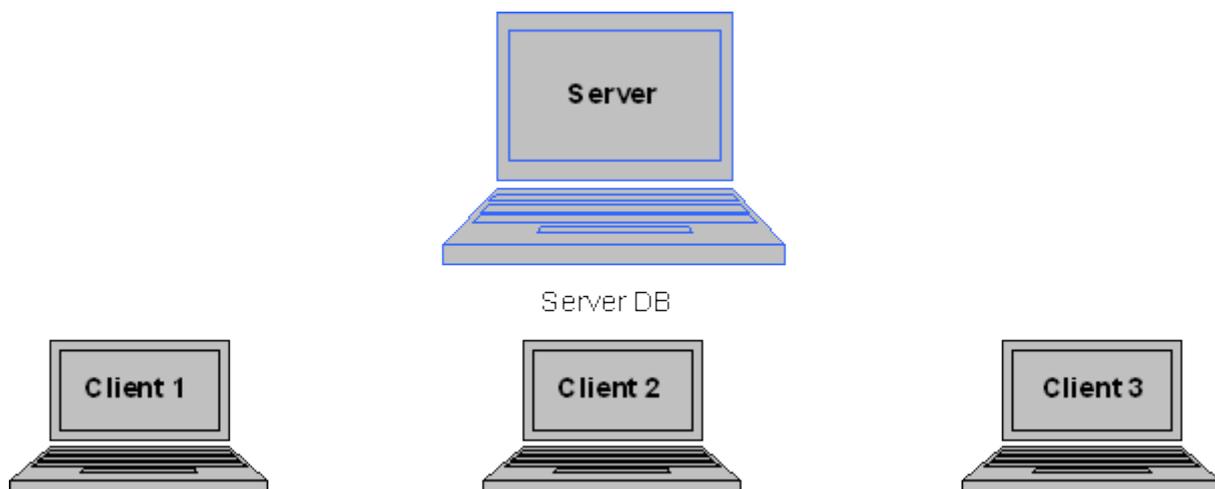


Local project copies 1-3 of the Server DB

> ✦ **Information**
>
> Various configurations of the example above are also possible. Each computer has, for example, one or more server databases for different projects installed and at the same time only one local project copy of another server database.

## SERVER WITHOUT LOG INS

For Servers in which no user is logged in, the **zendbsrv.exe** must run as a Service.

The **zendbsrv** service starts the **zensyssrv** service for logging tasks**.** For the operating system Windows 7 this is not permitted via **Remote Transport** due to security reasons.

The file must to be registered manually:

*zendbsrv.exe -service*

**Hint:** Also register file **zensyssrv.exe** as service: *zensyssrv.exe -service*

---

### 💡  Information

Once both files have been registered as services, **Remote Transport** can no longer be used from Windows 7. Runtime can thus not be started with **Remote Transport**.

---

# 6  Creating a new multi-user project

When a new **multi-user project** is created, the **Multi-user project** checkbox must be activated.

In the **Project server** field, select the server for the **Multi-user project** or enter the server name using the keyboard.

Note:

▸  If a server is stated in the **Project server** field, the **Multi-user project** is activated automatically.

▸ If there is no entry in the **Project server** field, the project and thus the server database is created on the local PC. A local project copy is created automatically with which you can work directly in the Editor.
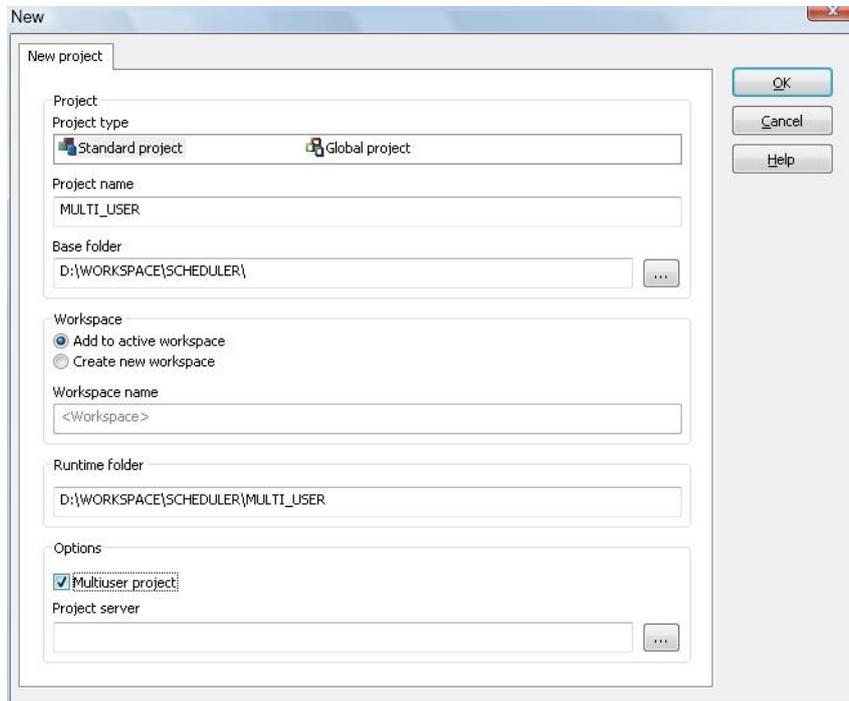


**REQUIREMENTS:**

▸ Project database server (**zendbsrv.exe**) must be started:

  ▸ On the server

  ▸ On the client computer

▸ Firewall setup:

  ▸ The **zendbsrv.exe** and **sqlservr.exe** programs must be enabled

  ▸ The SQL browser must be enabled on the server and the client

▸ Necessary open ports:

  ▸ 1103

  ▸ 1433 for SQL Server: this port can change

    The SQL Server process must therefore be exempted from the firewall completely.

  ▸ 1434 for SQL Browser

▸ The name resolution in the network must work

▸ Reverse lookup must deliver the same computer name both at the server and at the client

> ⚠ **Attention**
>
> zenon workspaces are not multi-user compatible. A workspaces of the zenon Editor is always tied to the computer on which the zenon Editor has been started.
>
> For this reason, the project hierarchy can be configured differently on different computers. The users have to communicate directly so that the hierarchy is the same on all stations.

## CREATING A CLIENT FOR A MULTI-USER PROJECT

Carry out the following steps to create a client for a **multi-user project**:

1. Go to **File** and **Add existing project...**.
   The **project selection** dialog is opened.

2. In this dialog, select the name of the computer under **Computer name**.
   To do this, enter the name of the desired computer in the network or click on the **...** button to show a list of computers in your network. The reading of the computer list using the **...** button can take some time depending on the size of your network.

3. After selecting or entering a computer name, the **project selection** list is updated.
   In this list, all zenon projects available on the computer are listed.

4. Select under **Project selection**, from the list of available projects, the project that you want to integrate on your local computer in your zenon **Workspace**.

5. Confirm your selection by clicking on the **OK** button.
   The project is listed in your zenon **Workspace**.

## NOTES ON CONFIGURATION WITH ZENON LOGIC

Note when configuring with zenon Logic:

▶ In the zenon Logic Workbench, lines can be created in the **All projects** node **MultiSpy** and the global **Binding** Editor can be called up.

　 The following is applicable for these settings: They are

　　▶ Not saved as multi-user

　　▶ Not transferred to the project server

　　▶ Removed again in the next synchronization

▶ If new zenon Logic projects are created in the zenon Logic Workbench via **File** and **Add new project**, the following is applicable: The projects are

　　▶ Not transferred to the project server

▸    Deleted again in the next synchronization

**Attention:** Multi-user zenon Logic projects can only be created using the zenon Editor.

# 7  Convert single-user projects to multi-user projects

To convert a **standalone project** into a **multi-user project**:

1.   Make sure that the global project is a multi-user project.
     If that is not the case, remove the existing global project from the workspace and

     create a new, multi-user-compatible global project:

     a)   In the **project manager**, right-click on the **Workspace**.

          The context menu is opened.

     a)   Go to **New project**.

2.   Convert the **standalone project**: select the **Multi-user** command in the context menu of the
     project tree    and then **Convert to multi-user project**

3.   You receive a warning that this step cannot be undone.
     A **multi-user project** can however be newly created (on page 13) as a **standalone project** using a
     project backup.

4.   Confirm the warning by clicking on **Yes**.

> 💡  **Information**
>
> The computer on which the conversion is carried out is defined as the database
> server for the project. This property cannot be changed later.

# 8  Convert multi-user projects to single-user projects

To convert a **multi-user project** into a standalone project:

1.   Make sure that all project changes are taken over by all clients:

     ▸    **Go to Project -> Context menu -> Multi-user -> Synchronize**

2.   Create a project backup:

     ▸    **Node project backup -> Context menu -> Create backup**

     ▸    Enter a name in the **create project backup** dialog under **Filename** and a brief project
          description under **Description**.

**Note:** You can also back up a local project copy.

3. Create a new project from the backup:

   ▶ Workspace -> Context menu -> Restore project backup.

   ▶ Select the desired project backup.

   ▶ Activate checkbox **Create new project**.

   ▶ Confirm your selection by clicking **OK**.

   ▶ Enter a new project name.

4. The project is created as single-user project.
   The **multi-user project** also still exists.

**Note:** To convert the local project copy to a single-user project, you can also select the option **Project** and **Save under ...** in the Editor in the context menu of the project.

# 9 Working with a multi-user project

**Multi-user projects** enable access to all modules.

Note the following when configuring **multi-user projects**:

▶ the options, Enable changes and Accept changes (on page 20) and their handling of changes in several modules

▶ Deleting (on page 25) elements

▶ Synchronizing (on page 25) elements

▶ the exceptions of VBA (on page 27) and RGM (on page 26)

## RENAME PROJECT

Projects cannot be renamed locally with the **Save as** command, because it is always the local project copy that is always saved.

To save a server project under a new name:

1. Make a backup of the server project.

2. Select **Restore backup as a new project** in the context menu.

3. Enter the new chosen name.

   The project is now available with the new name.

## 9.1 Extend and synchronize project

If a new element (such as a variable or a screen) is created, this element is initially only available locally and changes to the element are automatically enabled.

If the changes are:

▸ Applied: Another client must synchronize in order for the amended elements to also be available on its computer.

▸ Discarded, after a new element has been created: In this case it would be as if this element had never existed.

The difference between a new variable and a new screen is, that after a synchronization the variable is not visible on the other computers, if the changes have not yet been accepted. After synchronization, a screen is visible as **in progress** on the other computers.

See also chapter Screens / Frame editor.

After another user has added or deleted an element, the project has to be synchronized in order to receive the changes. This synchronization is done with the entry **Synchronize** in the context menu of the project. To do this, select the project in the zenon Editor. Call up the context menu and select in the entry **multi-user**, the entry **Synchronize**.

**CONTEXT MENU**

The following options are available in the context menu:

| Parameter | Description |
|---|---|
| **Multi-user/synchronize** | The project is synchronized. |
| **Multi-user/activate offline mode** | The offline mode (on page 29) is activated. Changes take effect only locally and are not taken over centrally as long as the project remains offline. |
| **Multi-user/activate online mode** | The online mode (on page 30) is activated. If elements have been edited and taken over by other users in the meantime, they overwrite changes from the offline mode. |
| **Update local version** | The local copy is updated with the Server DB. The project is reloaded into the Editor.<br>**Attention:**Excluded are those elements/modules, that are checked out by this user. |

---

### 💡 Information

The elements that have been checked out by the same user who has performed the synchronization will not be overwritten by the synchronization. The local changes are protected from being overwritten by the checkout status.

**Exception:**
User A checks out elements offline.

User B checks out elements online.

If user A reconnects to the server, the elements checked out by both users are overwritten with the data of user B. If from a team one member is out of office, we recommend not the change the project in this offline phase!

## 9.2   Changing existing elements

If changes to an element are required, it must first be checked out.

This prohibits more than one user gaining access to the same element.

Note:

▸   Elements that are checked out cannot be changed by other users.

▸   The Element can only be changed by the user that has it checked out. This user has the exclusive change rights. For all other users this element is prohibited and locked.

▸   In order to allow changes to an element, the user has to select the entry **allow changes** in the context menu of the element. It is possible, to check out only single elements, e.g. one variable. To do this, select the single variable in the detailview and select entry Allow changes in the context menu. It is also possible to check out all variables in the project at once via the module node.

If a project is a **multi-user project** the entry **Under construction** is added to the Project Manager. In the detail view, the elements of the current project are listed.

The following columns are shown by default:

▸   Status

▸   Element

▸   Objective

▸   Module

Additional columns can be added in the context menu of the column headers with the entry 'field selection'. With the additional columns, the list shows an overview of which elements have been edited by which user and on which computer.
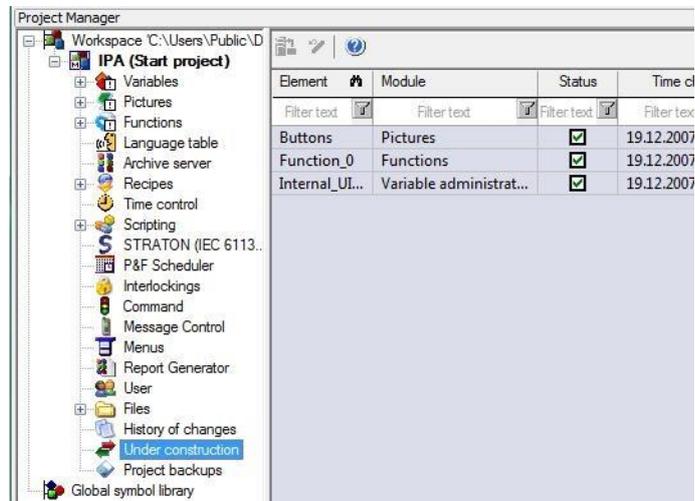


💡 **Information**

If you want to use subprojects in a multi-user environment, make sure, that both projects have the entry Allow changes (on page 20) activated before dragging the subproject under the integration project.

An element/module can only be 'read' but not edited, as long as the user has not fetched the authorization for it. The authorization can be fetched with the context menu of the element **Allow changes** or directly at opening the element.

## 9.3 Deleting projects

If a project is deleted, only the local project copy is deleted. The server DB remains available and other clients can still work with it. It is still possible to add the project to the workspace. Here a new local project copy is created again.

If a user deletes his local project copy, all check out information and thus all changes of this user are lost.

## 9.4 Disconnect client

With the option **disconnect client** it is possible to disconnect the local project copy of a user from the server database. All check out information of this user is subsequently deleted. Changing the elements locked by this user again is possible.

It could be necessary to disconnect a client, if the user is not in the office and has checked-out elements, which urgently have to be edited or the computer of this user is defective.

> ⚠️ **Attention**
>
> If a client of a user has been disconnected and the user opens the project copy, the project is offline. It is no longer possible to go online. First the local project copy has to be deleted and again copied from the server.

**REQUIREMENTS:**

A client can only be disconnected, if the computer, where the local project copy has been edited, cannot be reached.

With the main menu **File** -> **Insert existing project** the **Project Database Manager** can be opened. Under multi-user and connected local project copies you will find all users, which have (had) a local project copy.
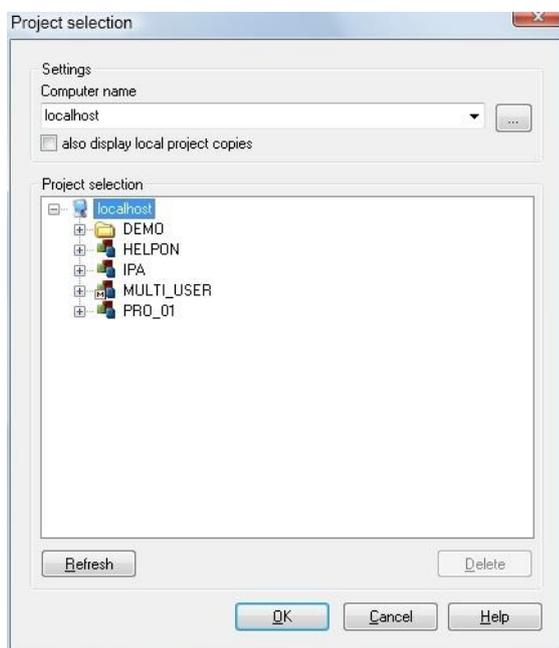


With disconnect the entry for this user is deleted from the server database. After the client has been disconnected, the dialog can be closed.

> ⚠️ **Attention**
>
> This action cannot be undone! If the user on this computer again wants to access the multi-user project, the local project copy has to be deleted and again be inserted.

## 9.5 Backup mechanism

In general, the following should be noted for the backup mechanism with **distributed engineering**:

▶ Only local project states can be backed up.

▶ As long as they have not been synchronized, restoring is possible.

### 9.5.1 Project backup

The project backup is executed in the project manager in the node **Project backups**.

### 9.5.1.1 Project backups of local project copies

A project backup can be made using the context menu of the **Project backups** module. To make a local project backup, the **Create local project copy** check box must be activated in the **Save project backup** dialog box.

Local project backups can only be made within the same offline session as the backup.

> ☀ **Information**
>
> You can find further information in the project backup manual in the Project backup detail view tool bar chapter.

### 9.5.1.2 Project backup of server projects

A backup of a server project can be executed from each client, if the checkbox **Save local project** is not active.

> ☀ **Information**
>
> You can find further information in the project backup manual in the Back up and export project chapter.

### 9.5.2 Restore project backup

A project backup can be restored using the context menu of the **Project backups** module.

Local project restores can only be carried out within the same offline session as the backup.

> 💡 **Information**
>
> You can find further information in the project backup manual in the Project backup detail view tool bar chapter.

### 9.5.2.1 Restore project backup of local project copies

Restoring a local project copy is only possible, as long as no synchronization has been executed. If a project has been synchronized since the last backup, the user only has the possibility to Restore backup as new project. Thus the project is restored as a single user project.

> 💡 **Information**
>
> You can find further information in the project backup manual in the Restore project backup chapter.

## 9.6  Allow, accept and reject changes

To make changes to a project, the respective elements (functions, variables, screens, ...) must be enabled for change (**Enable changes**). After editing, the elements must be protected from change and must be transferred to the Server (**Accept changes**).

**Note:** New and deleted elements are always transferred at the next synchronization. If the user deletes an element, it can still be edited on another computer. The element is finally deleted after the next synchronization.

| Parameter | Description |
|---|---|
| **Update local version** | If other users made changes in the server project, the local project copy can be updated with the menu entry.<br><br>Updates only single existing Elements/Modules. New elements at the Server are therefore not available, synchronization is required. |
| **Enable changes** | In order to be able to change elements, the status 'Allow changes' has to be activated. This can be done in the context menu of the selected element. |

| Parameter | Description |
|---|---|
|  | In order to work quicker, the user can enable changes for the whole variable module. This means that all variables, drivers and data types are locked for all other users. |
| **Accept changes** | The changes are transferred into the server project. The element/module is then unlocked for other users.<br>If changes were allowed for a complete module, they only can be accepted for the complete module.<br>This means, that all variables, drivers and datatypes are transferred and unlocked for other users. Accepting a single element is not possible in this case.<br>The changes in the local project copy are rejected and overwritten by the server data. The element/module is then unlocked for other users. |
| **Discard Changes** | All variables, drivers and datatypes are rejected and unlocked for other users. Rejecting a single element is not possible in this case. |
| **All modules** | ▶ **Update local version:** Updates the local version.<br><br>▶ **Enable changes:** Enables changes.<br><br>▶ **Accept changes** Accepts changes made.<br><br>▶ **Undo changes:** Rejects changes made. |

💡 **Information**

When you change the project name, **Undo changes** is not possible. The changed project name remains.

## CHANGES ON SEVERAL MODULES AT THE SAME TIME

If a module is enabled for changes and if elements of other modules are newly created and linked with this module, all affected elements/modules must be set to **Accept changes**. Enable the newly created elements/modules first and then enable the already existing ones. If the user enables the module which was originally disabled, recent changes may be lost.

**Example:**

▶ On one Client the RGM is checked out.

▶ A new recipe group is created. Then a new recipe.

▶ An existing variable is added to the recipe group.

▶ Three new variables are created and added.

‣ Interlockings are checked out and a new interlocking is added.

‣ A new keyboard screen is created with a new frame.

‣ In the recipe group the new created interlocking is assigned.

‣ The keyboard screen is assigned.

‣ A new variable for the visibility is created and linked with a variable in the recipe group.

If only the recipe group is set to **Accept changes** and another Client is synchronized:

‣ the other Clients gets:

‣ a recipe group with a linked variable

‣ and a recipe which contains the variable

‣ all other information is lost

If this Client disables, edits and enables the recipe group, all newly engineered information is lost.

For correct acceptance **Accept changes** must be carried out for:

‣ RGM

‣ Recipe group

‣ Recipe

‣ Variables

‣ Interlockings

‣ Keyboard screen

‣ Frame for keyboard screen

## 9.6.1 Status of the elements and the modules

If an Element has been checked out, the Symbol in the project manager or the detail view changes. The user who has checked out the element gets a checkbox with a green check mark displayed. All other users get a checkbox with a red **X**. The **!** symbol means that single elements within a module are checked out.

See chapter Legend (on page 24).

In the special case of the variable administration, symbols have a slight transparency, if the module has been checked out.

📄 Example

All variables are checked out by a user. Another user cannot change these variables, but can add an additional variable.

Exception: Module checkout

The following actions are only possible if the whole module is set to **Enable changes**:

- ▶ creating array variables
- ▶ creating structure data types
- ▶ copying and pasting array and structure variables
- ▶ creating variables based on a structure data type with array
- ▶ changing the array settings of a variable
- ▶ changing array settings in zenon Logic of a variable with zenon profile

### 💡 Information

If a variable is created without the whole module **Variable** set to **Enable changes**, the dimensions are automatically set to *0*. This prevents creating a structure variable unintentionally.

The following elements/modules can only be set to **Enable changes** or **Accepts changes** as a whole:

- ▶ Archives
- ▶ User
- ▶ Menus
- ▶ Reaction matrix
- ▶ Standard recipes
- ▶ Recipegroups
- ▶ Scripts
- ▶ Fonts
- ▶ Interlockings
- ▶ Allocations

## DISPLAY CHANGED ON DATE IN THE DETAIL WINDOW

In the detail view there are two columns available which display information about the last changes that were made:

- ▶ **Last change accepted**: Displays in all lists regarding distributed engineering, the time of the last update in the database or of the last check in.
- ▶ Last modified: Displays a time stamp of the last change to the files.

If the columns are not displayed, you can activate them:

1.  Right-clicking on the column header.

2.  In the context menu select menu item **Column selection**.
    It depends on the module which columns are available.)

3.  Select the desired columns in the selection list on the left-hand side (multi-select is possible)

4.  Add the columns by clicking button **Add** to the right-hand list

5.  The Up and Down buttons allow the user to select the desired position

6.  Confirm and close the dialog by clicking **OK**.

## 9.6.2 Keys of the symbols

The following symbols indicate the status of the elements in multiuser projects.

(for example: A screen can be changed by a user ☑)

**ELEMENTS IN THE DETAIL VIEW OF THE PROJECT MANAGER:**

| Symbol | Description |
| --- | --- |
| ⊞ | New element has been added in the local project copy. This is only transferred when **checking in** (Check-In) the element on the server. |
| ⊞ | New module has been added in the local project copy. This is only transferred when **checking on** (Check-In) the module on the server. |
| ⊠ | Elements have been marked as deleted in the local project copy. This is only deleted when **checking in** (Check-In) the element on the server. |
| ⊠ | Module has been marked as deleted in the local project copy. This is only deleted when **checking in** (Check-In) the module on the server. |
| ☑ | Element can be edited by the current user. |
| ☑ | Module can be edited by the current user. |
| 🔒 | Element is locked by another user. |

**DISPLAY IN THE PROJECT TREE:**

| Symbol | Description |
|---|---|
| | Module can be edited by the current user. |
| | Module is edited by another user and thus is locked for the current user. |
| | At least one element of the module is marked for editing (not the entire module). |

## 9.7 Deleting elements

A distinction is made between:

▸ Database elements

▸ File elements

Data base elements are only displayed as being deleted at first.

**Note:** As soon as a database element in the database of a multi-user project is marked as deleted, this database element is also no longer available in the Editor's selection window.

On other clients, no more changes can be made to elements marked as deleted. These are marked as locked. If changes are applied, the elements are deleted for good. It is possible to undo the deletion with "undo changes".

> 📄 Example
>
> This is true for variables for instance: The changes have to be accepted to allow deleting.

File elements such as screens, files (language table, graphics) etc. are directly deleted when a confirmation is done on the executing computer. New and deleted elements are always transferred at the next synchronization. If the user deletes an element, it can still be edited on another computer. However it is finally deleted after the next synchronization.

## 9.8 Synchronization of the multi-user project

After another user has added or deleted an element, the project has to be synchronized in order to receive the changes. This is done in the context menu of the project with the entry **Project -> Context menu -> Multi-user -> Synchronize** or **All modules/update local version**.

The following options are available in the context menu:

| Parameters | Description |
|---|---|
| **Synchronization** | The local copy is updated with the Server DB. The project is |

| Parameters | Description |
|---|---|
| | reloaded into the Editor.<br>**Attention:** Exception: The Elements/Modules, that are checked out by this user. |
| **Update local version** | See synchronize |
| **Enable changes** | Checks out Elements/Modules |
| **Accept changes** | Checks in Elements/Modules |
| **Undo changes** | The changes of the element are not accepted but discarded. After **Undo changes** the project is overwritten by the Server database again.<br><br>**Attention:** If **changes are undone** at an element that has not yet been "accepted", the changes that depend on this element are also undone. |

New and deleted elements are always transferred at the next synchronization.

## 💡 Information

The following are ignored during synchronization:

▸ Elements and modules that are checked out by the same user who is carrying out the synchronization.

▸ Empty folder.
A folder must contain at least one element in order to be synchronized. An empty folder is created, for example, in the **Help** folder if a new language is added to the language file.

## 9.9 Recipegroup Manager

### FILTER DIALOG IN THE RUNTIME

If the **Show dialog in the Runtime** property is activated in the options of the recipe value list in the screen switch, there are only restricted options available in Windows CE:

▸ Recipe Filter

▸ Column settings (for recipe table)

**Recipe selection**, **Equipment modeling** and column settings for the **recipe list** are not available.

## 9.10 VBA and VSTA

Select the entry **Open VBA Editors** or **Open VSTA Editor** in the context menu of **Programming interfaces**, then the Editor is checked out and changes are enabled.

**Attention:** In general in VBA changes are possible even if the element has been set to **Enable changes** on another computer.
Therefore consider the following before making changes via VBA:

1. Before you make changes: check whether you are dealing with a multi-user project.

2. If yes: check whether the element is added by another Client

3. If yes: check whether the element has the local status **Enable changes** (check out)

    a) If yes: The object can be changed,

    b) If no: The object must be set to **Enable changes**

    ‣ Check again whether the object has the **Enable changes** status

    ‣ If yes: the object can be changed in VBA

4. After the changes the object must be set to status **Accept changes** (check in) via VBA

> ⚠️ **Attention**
>
> In multi user projects the VBA Editor must not be opened with Alt+F11. If VBA is not checked out, changes are not accepted. When the VBA Editor is opened, a warning message is displayed that indicates which projects are affected.

## 9.11 Closing the Editor

Each user should accept the changes to their objects before closing one of the following applications.

‣ Editor

‣ Project

‣ Workspace

If the changes are not accepted, other users cannot make changes to these objects. You can find details in relation to this in the Editor chapter, section:    Closing the Editor.

Upon closing the Editor, the workspace or the project, without accepting the changes, a dialog for accepting or refusing the changes is displayed to the user:
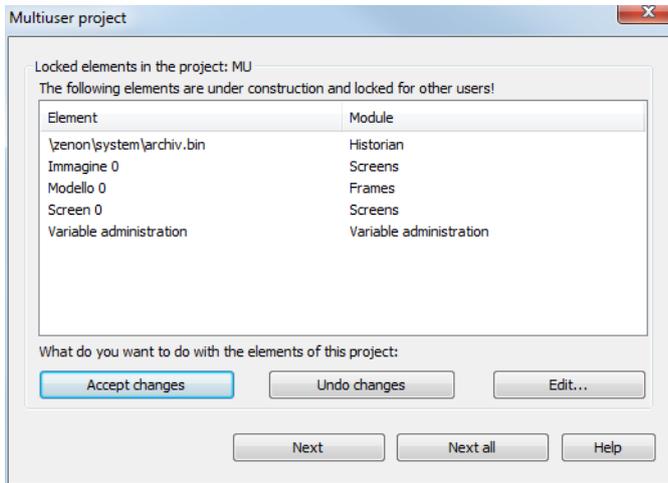


| Parameter | Description |
|---|---|
| List field elements | Lists all elements which are disabled for editing. |
| | Changes for the displayed elements can be accepted or discarded via the respective button. |
| Accept changes | The changes of all elements are accepted. |
| Discard Changes | The changes of all elements are discarded. |
| Edit... | Closing is cancelled. |
| Next | Continues the closing process for the current project. The elements remain disabled for the other users. If several **Multi-user projects** should be closed, this dialog is displayed for the project which should be closed next. |
| Next (all) | Continues the closing process for all **Multi-user project** in the workspace which should be closed. The elements remain disabled for the other users. |
| Help | Opens online help. |

## 9.12  Style groups and styles

Changes are always enabled for complete style groups only, and not for individual styles. If a style is edited or added in a checked-in style group, a corresponding message for whether changes are to be enabled for the style group is given.

Style groups created from elements or added copies of style groups are not automatically checked out after creation or copying.

# 10 Working offline

While working offline there is no connection to the server, e.g. when the engineer takes the project with him on the laptop.

To switch a project offline, the entry Activate offline mode has to be selected in the context menu of the project. Projects having no connection to the database server will be activated offline automatically.

> ### ⏻ Information
>
> I**Accept changes** and **Reject changes** is not possible during the offline operation.

Only **Allow changes** is available in offline mode, otherwise no changes would be possible in offline operation. **Accept changes** can be activated in the online period.

## 10.1  Switch offline

There might be different circumstances when switching offline

### 10.1.1 Elements/modules are locked by another user

At the moment of switching offline another user edits the element/module. In this case no changes can be done offline.

### 10.1.2 Elements/Modules are checked ot by the same user during offline switch

At the moment of switching offline the same user edits the element/module. In this case no online changes by another user are possible.

### 10.1.3 No Elements/Modules are checked out when switching offline

At the time of switching offline elements/modules are checked out.

#### CASE 1:

Offline changes for an element/module are allowed, but the same is also possible online.

User A changes an element/module offline, user B online.

When user A goes online, it is recognized that user B is just editing the module/element online.

The changes made by user A are lost.

> ⚠ **Attention**
>
> In this case the changes made offline (by user A) are lost!

### CASE 2:

When user A accepts his changes, user B already accepted his changes.

The changes made by user A overwrite the changes made by user B when going online.

> ⚠ **Attention**
>
> In this case the changes made online (by user B) are overwritten!

## 10.2  From offline to online

To switch an Offline-project online again, the entry Project/online has to be selected in the context menu. The project is reloaded into the Editor immediately.

## 10.3  Possibilities in offline operation

| Elements/modules | Create, change, rename, delete |
|---|---|