



3 Menu Items

Menu Items

3.1 Edit

3.1.1 Edit Menu, Overview

The main menu item "Edit" contains the following components, depending on which editor, dialog or wizard is active or when working in the Project Explorer:

- [Undo, page 89](#)
- [Redo, page 89](#)
- [Cut, page 90](#)
- [Copy, page 90](#)
- [Paste, page 90](#)
- [Delete, page 91](#)
- [Rename, page 91](#)
- [Select All, page 91](#)
- [Find and Replace, page 91](#), (character strings)
- [Bookmarks, page 96](#), bookmarking functionality in text-based editors
- [Insert file as text..., page 97](#),
- [Browse, page 97](#), information about object position and dependencies
- [Input assistance, page 98](#)
- [Declare variable, page 100](#),
- [Advanced, page 102](#), expanded list of commands for working with text-based editors
- ["List components" function, page 103](#)

If required, the menu structure can be reconfigured via the **IndraWorks ► Tools ► Customize ► Commands** dialog.

3.1.2 Undo

Icon:

Default shortcut: <Ctrl>+<Z>

Menu: **Edit ► Undo**.

The "Undo" command undoes the editing step most recently carried out in the currently active editor.

Repeated execution of the "Undo" command undoes all of the actions carried out since the editor window was opened, one by one. This applies to all actions in the function block editors.

Use "[Redo](#)" to restore the result of the editing step just undone.

3.1.3 Redo

Icon:

Default shortcut: <Ctrl>+<Y>

Menu: **Edit ► Redo**.

The "Redo" command restores the result of the editing step that was just [undone](#).

Menu Items

3.1.4 Cut

Repeated execution of the "Redo" command restores all actions previously undone, one by one.



Icon:

Default shortcut: <Ctrl>+<X>

Menu: **Edit** ► **Cut**.

This command transfers the current selection (object, character string) to the clipboard and **removes it from the previous position** in the Project Explorer or editor. In the Project Explorer, this applies to the currently selected object. Several items can be selected at once.



Note that not all editors support this command and that its use is limited in some editors.

The possibilities for selection differ depending on the editor:

In text editors, for example, the selection can consist of a character string or a single character.

In graphical editors, for example, the selection can include one or more elements inside a selection frame.

To insert the contents of the clipboard use the "Paste" command.

To transfer a selection to the clipboard without removing it from its current position, use the [Copy](#) command.

To remove a selection without changing the clipboard contents, use the "Delete" command.

3.1.5 Copy



Icon:

Default shortcut: <Ctrl>+<C>

Menu: **Edit** ► **Copy**.

The description for the "Cut" command also applies to this command, except that when copying, the selection is **not removed from its current position**.

To transfer a selection to the clipboard and remove it from its current position, use the "Cut" command.

3.1.6 Paste



Icon:

Default shortcut: <Ctrl>+<V>

Menu: **Edit** ► **Insert**.

This command inserts the contents of the clipboard at the current cursor position.

Not all editors support "Paste" and its use can be limited in others. Graphical editors only support the "Paste" command if a correct construct is established when the insertion is made.

In the Project Explorer, this command applies to the currently selected object.

Menu Items

Several items can be selected at once. Depending on the current position, a dialog might open in which to select whether the object on the clipboard should be inserted above or below the position.

To transfer a selection to the clipboard without removing it from its current position, use the "Copy" command.

To remove a selection without changing the clipboard contents, use the "Delete" command.

3.1.7 Delete

Icon:

Default shortcut:

Menu: **Edit ▶ Deleting.**

This command deletes the selection from the editor or from the Project Explorer. The contents of the clipboard remain unchanged.

In the Project Explorer, this command applies to the currently selected object. Several items can be selected at once.

The possibilities for selection are the same as for the "Cut" command.

To remove a selection from the editor or object tree and simultaneously transfer it to the clipboard, use the "Cut" command.

3.1.8 Rename

Default shortcut: <F2>

Menu: **Edit ▶ Rename.**

The command allows to "rename" an object. In addition to changing the name of the object itself, the names of its dependent objects can also be changed automatically.

Example:

Renaming an axis causes the names of the address constants stored in the global variable list "MlcVarGlobal" to be changed as well.

3.1.9 Select All

Default shortcut: <Ctrl>+<A>

The command can be used in editors to select the entire contents at once.

3.1.10 Find and Replace

Overview of Find and Replace

The Find/Replace commands are located in the "Edit" category. They can be used to search an entire project for a character string and to replace certain character strings with other character strings.

By default, the commands are located in **Edit ▶ Find and Replace .**

Commands:

- [Find, page 92](#)
- [Replace, page 95](#)
- [Search down, page 95,](#)
- [Search down \(selection\)95](#)
- [Search up, page 96,](#)
- [Search up \(selection\), page 96](#)

Menu Items

Find

Icon:

Default shortcut: <Ctrl>+<F>

Use this command to search the project for a specific character string. The search is executed in all areas of the objects in the project that can be edited.

To open the dialog "Search", click in the main menu on **Edit ▶ Find and Replace ▶ Find**.

The "Find" command opens the "Find" dialog:

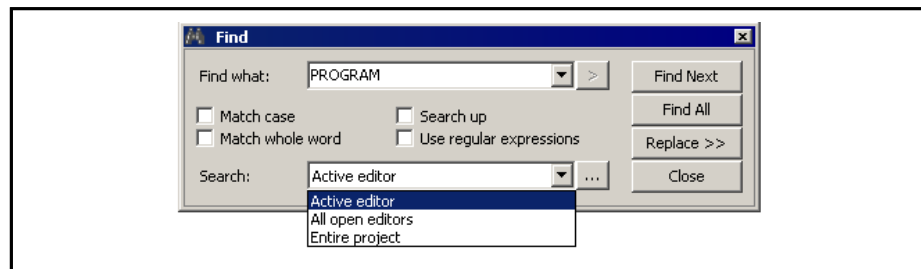


Fig.3-1: "Find" dialog

In the "Find" dialog, enter the string to be found, where the search is to take place and whether the positions found are to be displayed all at once or one at a time. Switch to the "Replace" dialog at any time.

Find what: In the "Find what" input field, enter the string to be found (search string). The selection list associated with the button provides all strings entered since the programming system was started.

Search options Select the desired search options:

- **Match case**
This search considers the case of the search string.
- **Search up**
The search area entered moves upwards. To search in a downward direction again, disable the "Search up" option.
- **Match whole word**
Only strings that exactly match the character string entered are found.
- **Use regular expressions**

Use the button for support when entering regular expressions to search for specific character strings. The most commonly used regular expressions are supported and are sorted into the following submenus:


- Special characters
- Repetitions
- Alternatives
- Groups
- Others

Search in: In the "Search in" selection list enter the objects that are to be searched for and the character string to be found; see the following figure.


Menu Items



Fig.3-2: "Find" dialog

Use the  button to select one of the options from the list. The following can be selected:

- Active editor,
- All open editors,
- Entire project,
- Only selection.

In addition, the  button can open the dialog for defining the search area where the search can be specified, see the following figure.

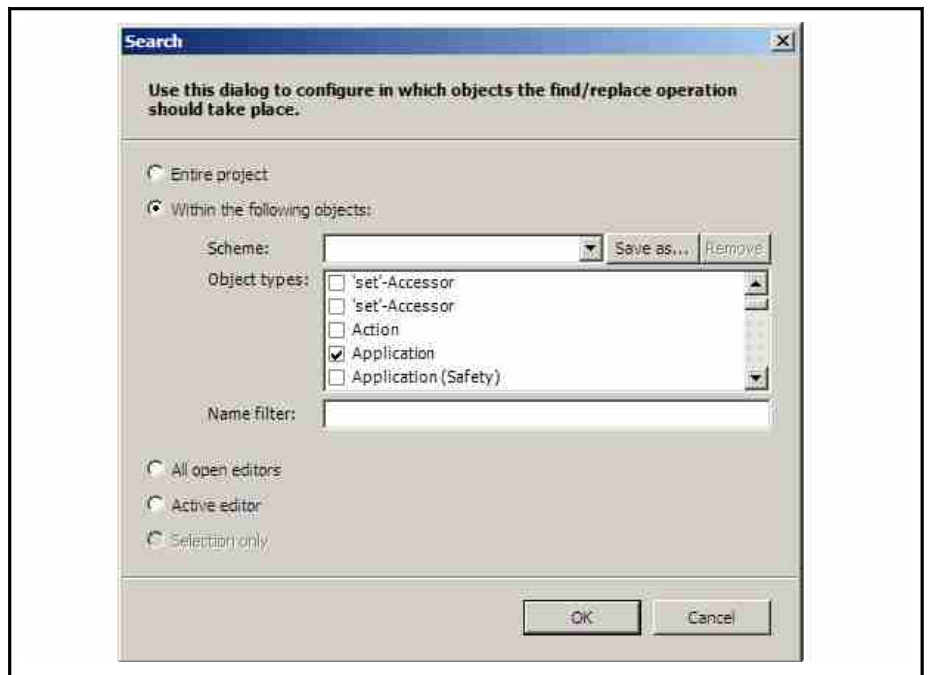


Fig.3-3: Dialog for entering the search area

- Entire project
Select the "Entire project" option if the search for the character string should include the entire project.
- Within the following objects
Select the "Within the following objects" option if the search is to be limited to the selected object types.

Proceed with the following steps:

1. **Object types:**In the **Object types** selection field, specify which object types are to be included in the search for the character string

Menu Items

by placing a checkmark next to the object type(s) in the selection list.

2. **Name filter:** Optionally, set a name filter for the objects to be searched whereas placeholders "*" can be used. Example: If all objects with "CAN" in their name are to be searched, enter "**CAN" as filter here.
3. **Scheme:** You can save the current search configuration. Press <Save...> and enter a name for the configuration in the "Save Scheme" dialog.


All saved schemes are then contained in the selection list ( The currently selected scheme can also be deleted again using the <Remove> button).



Fig.3-4: "Save Scheme" dialog

- **All open editors:** Select the "All open editors" option if only the open editors are to be searched for the desired character string.
- **Active editor:** Select the "Active editor" option if only the active editors are to be searched for the desired character string. The active editor is the editor for which the working area is displayed in the foreground.
- **Selection only:** The "Selection only" option is ### being prepared ###




Use the "OK" button to confirm the settings.

Once all search options are defined, press the button...

Find next Click on the "Find Next" button to jump to the next position where the string was found. The respective editors are opened and the string found is highlighted in the display.

Search all In order to get a list of all positions found in the message window. The progress of the search process is shown in the status bar and can be canceled early by means of the <Cancel> button displayed in the status bar if necessary.

After the search is complete, the following information is indicated regarding each position:

- **Priority:** Message icons.
Messages can include
Errors 
Warnings  or
Just information .
- **Description:** Expression that contains the search string.
- **Origin:** Path where the item searched for was found.
- **Position:** Position (e.g. line number) within the objects; "Decl" in brackets stands for the declarations or "Impl" for the implementations of the editor.

Menu Items

The following is displayed below the list:

- Total number of positions found
- Number of objects in which the string was found
- Total number of objects searched

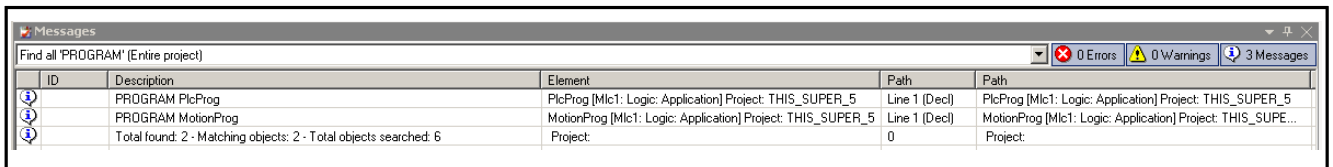


Fig.3-5: Message window with positions found

To replace this string with another string, click on the [Replace, page 95](#) button to open the "Replace" dialog.

Replace

Icon:

Default shortcut: <Ctrl>+<H>

The "Replace" command opens the "Replace" dialog, which is an extended "Find" dialog; see the following figure.

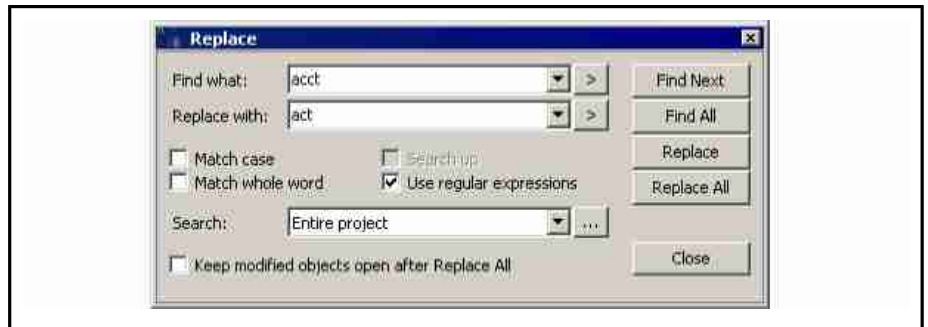


Fig.3-6: "Replace" dialog

As in the "Find" dialog, select the options for finding the character string to be replaced. In addition, enter the new character string in the "Replace with:" field and then click on one of the "Replace" buttons; see the following table:

| Button | Function |
|-------------|---|
| Replace | The "Replace" button highlights and replaces the next string found in the editor (step by step replacement). |
| Replace all | The "Replace all" button replaces all of the strings found at once without displaying them in the respective editor(s). |

Fig.3-7: "Replace" function commands

Search down

Icon:

Default shortcut: <F3>

Menu: **Edit** ► **Find and Replace** ► **Search Down**.

The "Search down" command can be used to move to the next position found after using the "Find" or "Replace" commands to search for a specific character string in the project.

Search down (Selection)


Default shortcut: <Ctrl>+<F3>

Menu Items

Menu: **Edit** ► **Find and Replace** ► **Search Down (selection)**.

The "Search down (selection)" command can be used to move to the next position found in the project for a character string that corresponds with the string currently highlighted in the editor.

Search Up

Icon: 

Default shortcut: <Shift>+<F3>

Menu: **Edit** ► **Find and Replace** ► **Search Up**.

The "Search Up" command can be used to move to the position previous to the one found after using the "Find" or "Replace" commands to search for a specific character string in the project.

Search Up (Selection)

Default shortcut: <Shift>+<Ctrl>+<F3>

Menu: **Edit** ► **Find and Replace** ► **Search Up (selection)**.

The "Search up (selection)" command can be used to move to the position previous to the one found in the project for a character string that corresponds with the string currently highlighted in the editor.

3.1.11 Bookmarks

Bookmark, Overview



The menu item provides the bookmarking functionality for text editors. It can be accessed in **Edit** ► **Bookmark** or via the "Bookmarks" toolbar.

Bookmarks can be added to lines to facilitate navigation in long programs.



Bookmarks remain when closing the editor window.


If IndraWorks Engineering is closed, all bookmarks are removed.

The commands described below allow the setting of bookmarks and the navigation with them.

Commands:

- [Switching Bookmarks, page 96](#),
- [Next Bookmark, page 97](#),
- [Previous Bookmark, page 97](#)
- [Deleting Bookmark, page 97](#)

Switching Bookmark

Icon: 

Menu: **Edit** ► **Bookmark** ► **Set/Remove Bookmark**

The command can set a bookmark in a text editor to set a bookmark in the current line or remove an existing bookmark. A set bookmark is displayed as cyan blue square in front of the line.


```

11 ivar_err := ivar_err +1
12 strvar_err := INT_TO_ST
13 bInc := FALSE;
14 END_IF
15
16 IF ivar_err > 3 THEN
17   ivar_err := 0;
18 END_IF

```

Fig.3-8: Example of a bookmark in the ST editor

Next Bookmark

Icon:

Menu: **Edit ▶ Bookmark ▶ Next Bookmark**

Use the command to jump to the next bookmark (down) in the text editor.

Previous Bookmark

Icon:

Menu: **Edit ▶ Bookmark ▶ Previous Bookmark**

Use the command to jump to the previous bookmark (up) in the text editor.

Deleting Bookmark

Icon:

Menu: **Edit ▶ Bookmark ▶ Delete bookmark**

Use this command to delete all bookmarks in the active editor window.

3.1.12 Insert File as Text...

Use the "Insert file as text" command to insert a copy of the contents of a text file into the currently active text editor. Open the "Insert file" dialog in order to open a file.

To open the dialog "Insert file", click in the main menu on **Edit ▶ Insert file as text...**

Find the desired file which has to be available in text format. The file contents are then inserted at the current cursor position.

3.1.13 Browsing

Browsing, Overview

Functions to navigate through the source text are available in the "Edit" category. These are used to search the source text for information on the positions and dependencies (call tree) of a function block or a function currently edited in an editor.

The functionalities for navigating source text are located in the main menu **Edit ▶ Browse**.

Commands:

- [Go to Definition, page 97,](#)
- [Display Cross References, page 98.](#)

Go to Definition

Icon:

Menu Items

Menu: **Edit ▶ Browse ▶ Use to definition**

The "Go to definition" command can be used if the cursor is currently pointing to an identifier in an editor window.

It is used to search the project for the position that contains the "definition" of the respective variable or function block. Then, the function block, line or network found is opened in the corresponding editor.

For example, if there are several instances of the selected object in the project, the "Select online status" dialog opens to select whether the function block should be opened in offline or online mode and - for function block instances - whether an instance or the implementation should be displayed.

Alternatively, the command is available in the context menu.

Example The following function block contains "fbinst", a function block definition, "prog_y", a program call, and "fbinst.out", a function block call.

Example "Go To Definition"


```
VAR
    fbinst:fb1;
    ivar:INT;
END_VAR

prog_y();
ivar:=prog_y.y;
res1:=fbinst.out;
```

Cursor locations after using the "Go to definition" command

- Place the cursor on "prog_y" and the command opens the "prog_y" program in its editor.
- Place the cursor on "fbinst" and the command sets the focus to the declaration window to the line "fbinst:fb1;"
- Place the cursor on "out" and the command opens the "fb1" function block in its editor.

Displaying Cross References

Icon: 


The "Display cross references" command displays the cross reference list currently shown in the cross reference list window in the message box. To open the dialog "Display cross references", click in the main menu on **Edit ▶ Browse ▶ Display Cross References**. Alternatively, "Display Cross References" is available in the context menu.



Cross references in the cross reference list can also be automatically updated.

An automatic update of the cross reference list can be activated in the main menu via **Tools ▶ Options ▶ IndraLogic 2G ▶ Smart coding ▶ Automatically update cross references when changing selection** (see also "[Cross reference list](#)", page 107).

3.1.14 Input Assistance...

Icon: 

Default shortcut: <F2>

Menu: **Edit ▶ Input Assistance...**

Menu Items

The "Input assistance..." command, which opens the input assistance dialog, is only available if the cursor is currently in a programming language editor window.

The dialog provides all programming elements that can be inserted at the current cursor position.

Alternatively, the command is available in the context menu.

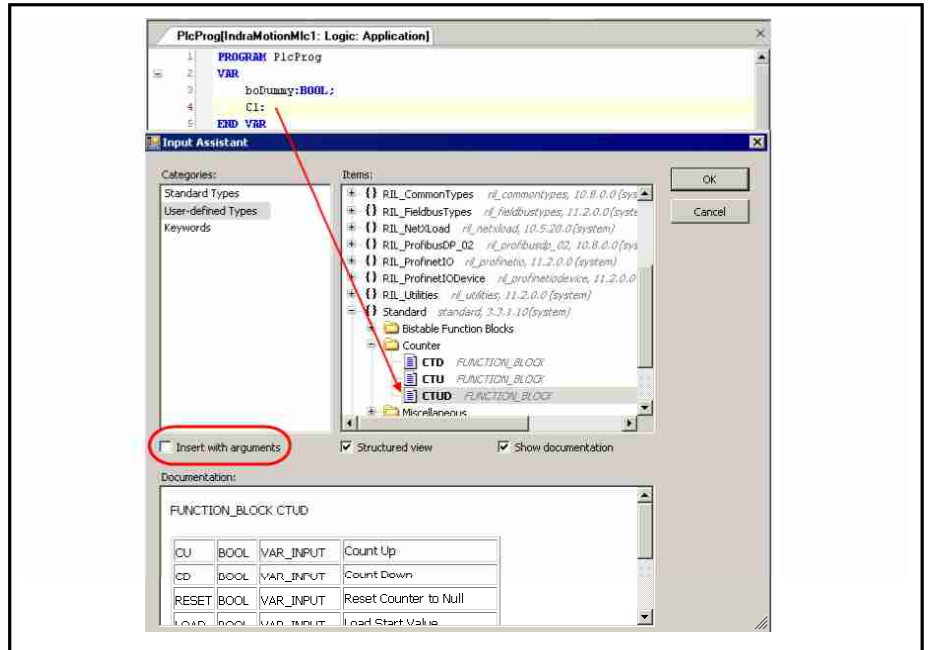


Fig.3-9: "Input assistance" dialog: Declaration of an instance (without inserting an argument)

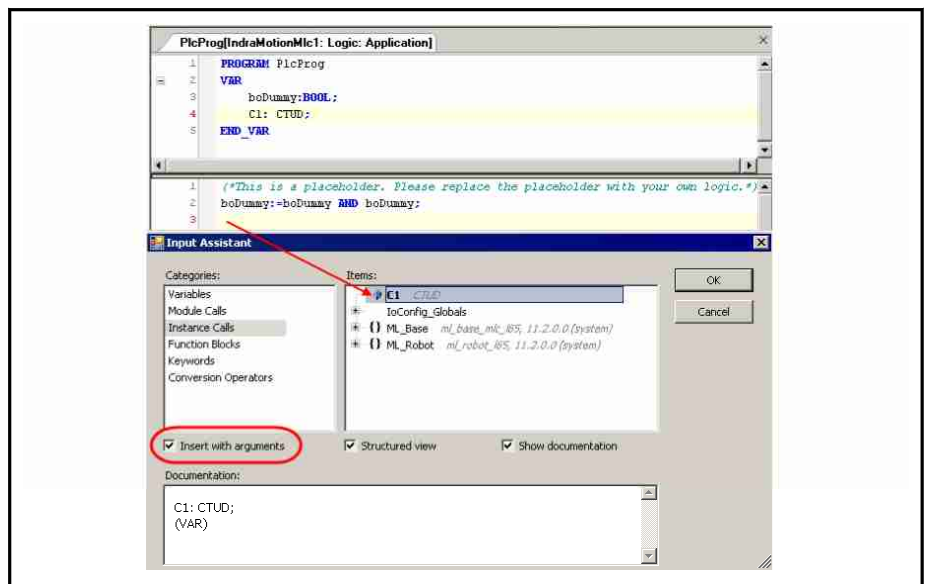


Fig.3-10: "Input assistance" dialog: Implementation of an instance (with inserting an argument)

The elements are sorted according to category. For the "Variables" category, additionally set a **filter** for the variable type, like "Local variables", "Global variables", "Constants", etc.

Menu Items

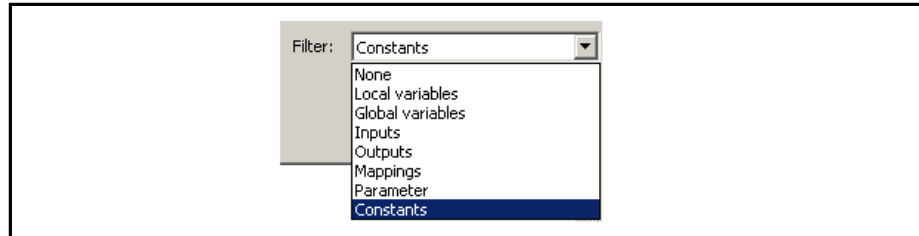


Fig.3-11: "Filter" selection window

Insert with arguments

If the "Insert with arguments" option is selected, elements that have arguments, e.g. functions, are inserted with their arguments at the cursor position.

Implementation, Inserting with Arguments, C1: CTUD;

```
C1 (
  CU:= ,
  CD:= ,
  RESET:= ,
  LOAD:= ,
  PV:= ,
  QU=> ,
  QD=> ,
  CV=> );
```

Structured view

If the "Structured view" option is selected, the elements are arranged in a tree structure based on subject and supplemented by icons.

If it is not activated, the elements are listed one after the other in alphabetical order. The function block to which each belongs is also listed (e.g. "GVL1.gvar1").

Show documentation

If the "Show documentation" option is selected, the dialog is expanded to include the "Documentation" field. Here a help text is shown for the currently highlighted element. This help text is automatically generated from a comment that was entered when the element was created.

3.1.15 Declare Variable...

Default shortcut: <Shift>+<F2>

The command opens the "Declare variable" dialog, which supports the [Declaration of a variable, page 503](#).

To do this, the cursor has to be positioned in a line of the programming section of the editor that contains a variable that has not yet been declared or a variable that has already been declared has to be highlighted.



If the dialog is to be opened automatically as soon as a line that contains a variable not yet declared is exited, the "Auto Declare" function has to be selected in the options for smart code editing. More information on "Auto Declare" or "Smart code editing" is located in "[Options, Smart Coding](#)", [page 205](#), "Settings".

To open the "Declare variable" dialog, enter the variable name and click in the main menu on **Edit ▶ Declare Variables...** or press <Shift>+<F2>.

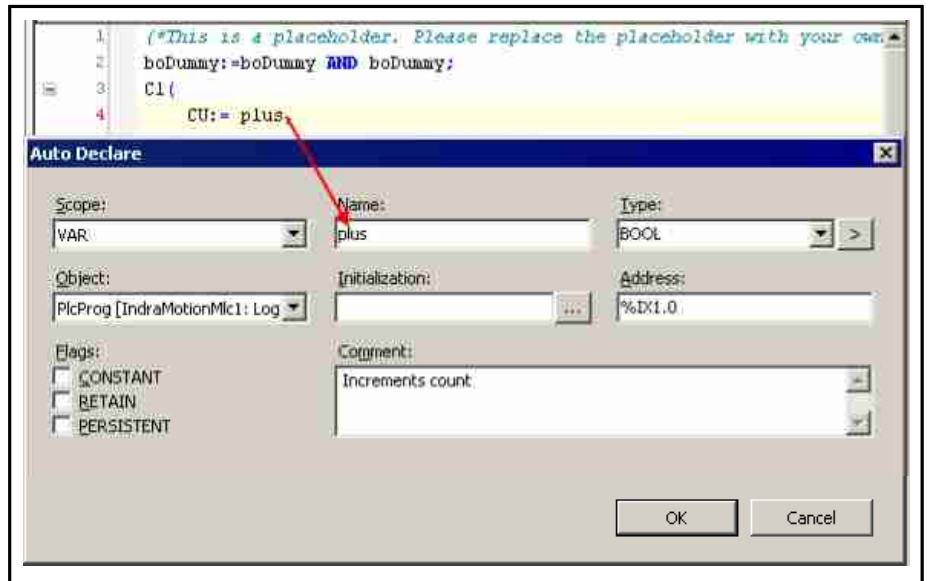


Fig.3-12: Dialog for variable declaration

A few fields in the dialog are automatically filled with default values, but these can be edited: Name, Data type, Object

- Visibility** In the "Visibility" selection list, "VAR" is the default entry for "local variables". Alternatively, another validity range can be set in the selection list.
- Name** The default entry in the "Name" field is the name of the variable that has not yet been declared as it was entered in the editor.
- Data type** The default entry in the "Data type" selection list is "INT" if this variable is the first one to be declared in the editor line.
However, if there is already a declared variable in the line, the data type for this variable is listed, here "BOOL". To change this setting, use the [...] button to access input assistance which provides all data types, see [Input assistance...](#), page 98.
- Object** The default entry in the "Object" selection list is the name of the [Project Object](#), page 28 that is currently edited. To enter a different object, in which the variable is to be declared, select one from the selection list. If a global variable list should be declared, it contains for example (visibility: VAR_GLOBAL) all available global variable lists of the project.
- Initial value** The "Initial value" field is empty by default. A valid value, which corresponds to the data type of the variable and which is used to initialize the variable, can be entered here. If the field is left empty, it is initialized with the default value.
The [...] button opening the "Input assistance" dialog.
- Address** The "Address" field is empty by default. In the "Address" field, an IEC address can be directly assigned to the variable ([AT Declaration](#), page 512).
Automatically created declaration line

```
plus AT %IX1.0: BOOL;
```

- Flags** No flags are enabled by default. "CONSTANT", "RETAIN" or "PERSISTENT" can be selected. Define whether the variable is a constant or a ["remanent variable"](#), page 519, by selecting the corresponding "flags". The corresponding attribute keyword is then inserted into the declaration after the keyword defined in the "Visibility" field, e.g. "VAR CONSTANT".

Menu Items

- Comment** The "Comment" field is empty by default. Enter a text to be added to the declaration as comment.
- Breaks can be inserted with the shortcut <Ctrl>+<Enter>.
- The comment is then displayed in the declaration editor in the line above the variable declaration.
- Click on the "OK" button to close the "Declare variable" dialog. The variable declaration appears in the declaration editor according to the IEC syntax.

3.1.16 Advanced Advanced Overview

There are additional commands for editing text in the project under "Edit".
By default, these commands are located in **Edit ▶ Advanced**.

Commands:

- [Go to..., page 102](#)
- [Switch to upper case, page 102](#)
- [Switch to lower case, page 103](#)
- [Corresponding bracket, page 103](#)
- [Highlight up to corresponding bracket, page 103](#)

Go to...

Default shortcut: <Ctrl>+<G>

Menu: **Edit ▶ Advanced ▶ Go To...**

Jump to a specific line within the active text editor using the command. The command opens the "Go To Line" dialog to enter the desired line number (jump target).

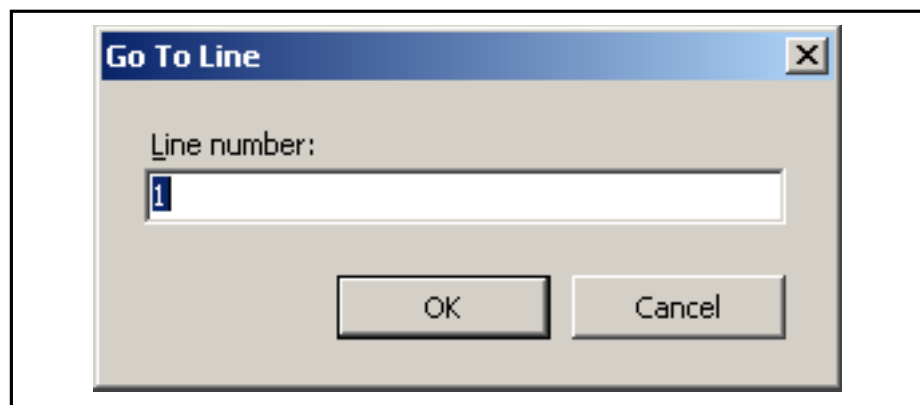


Fig.3-13: Dialog – Go to line

Then close the dialog with the "OK" button. The cursor is placed at the beginning of the corresponding line.

Switch to Upper Case

Default shortcut: <Ctrl>+<Shift>+<U>

Menu: **Edit ▶ Advanced ▶ Switch to upper case**

Use the command to display the currently highlighted text area in upper case letters.

See also [Switch to lower case, page 103](#).

Switch to Lower Case

Default shortcut: <Ctrl>+<U>

Menu: **Edit ▶ Advanced ▶ Switch to lower case**

Use the command to display the currently highlighted text area in lower case letters.

See also [Switch to upper case, page 102.](#)

Corresponding Bracket

Menu: **Edit ▶ Advanced ▶ Corresponding bracket**

When the cursor is positioned at a bracket, the command can be used to jump to the next related opening or closing bracket. This applies to brackets within program lines and to "parenthesis enclosures", [page 212](#) that extend across several lines.

Highlight up to Corresponding Bracket

Menu: **Edit ▶ Advanced ▶ Highlight up to corresponding bracket**

When the cursor is positioned at a bracket, the command is used to highlight all of the code lines up to the corresponding closing or opening bracket. The highlighted area can extend upward or downward. This applies to brackets within program lines and to "parenthesis", [page 212](#) across several lines.

3.1.17 "List Components" Function

Entering text is supported in the context of the IEC 61131-3 standard.

The IEC 61131-3-text editor also provides a type of "Intellisense" functionality that can be enabled or disabled using the dialog in **IndraWorks ▶ Tools ▶ Options ▶ IndraLogic 2G ▶ Smart coding.**

"List components" makes it easier to enter valid identifiers:

- If only a point "." is entered in a position where a global identifier can be entered, a selection list of all available global variables appears. A variable can be selected and inserted after the period.

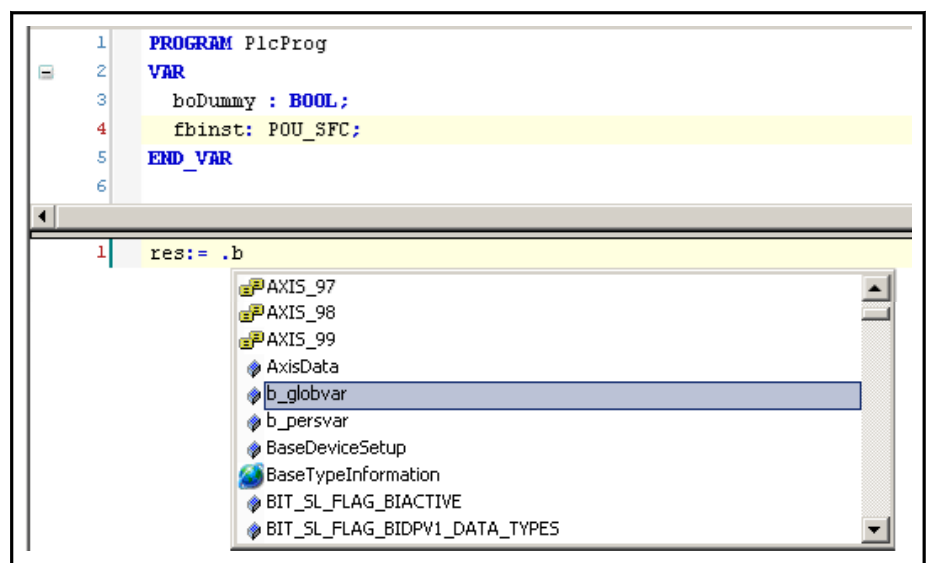


Fig.3-14: "List components" provides global variables (starting with 'b')

- If a period is entered after a function block instance or structure variable, a selection list provides all of the input and output variables of the func-

Menu Items

tion block or all structure components. With a double-click or <Enter>, insert the variable selected in this list after the period.

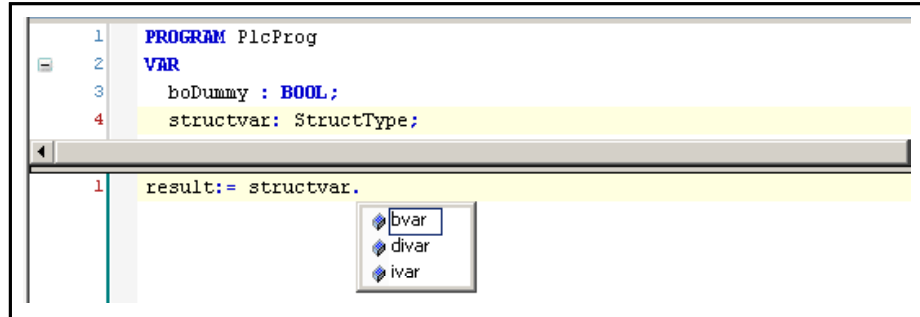


Fig.3-15: "List components" provides structure components

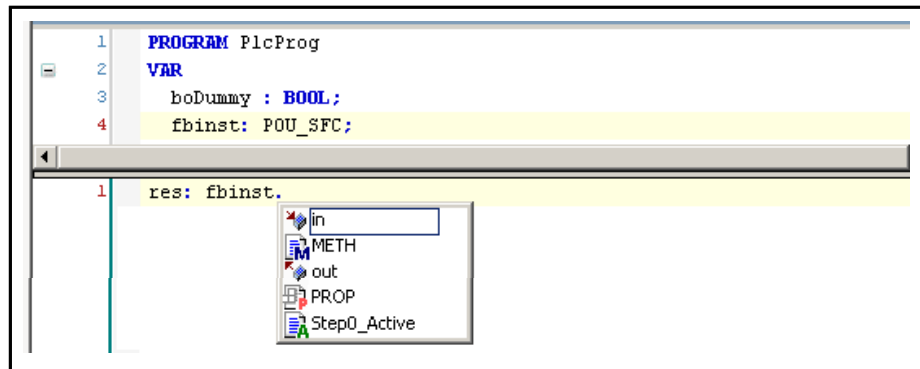


Fig.3-16: "List components" provides function block variables (components)

- If any character string is entered and <Ctrl>+<space bar> pressed, a selection list of all available POUs and global variables appears. The first element in this list that begins with the previously entered character string is automatically selected and can be inserted in the editor with a double-click or by using the <Enter> key.

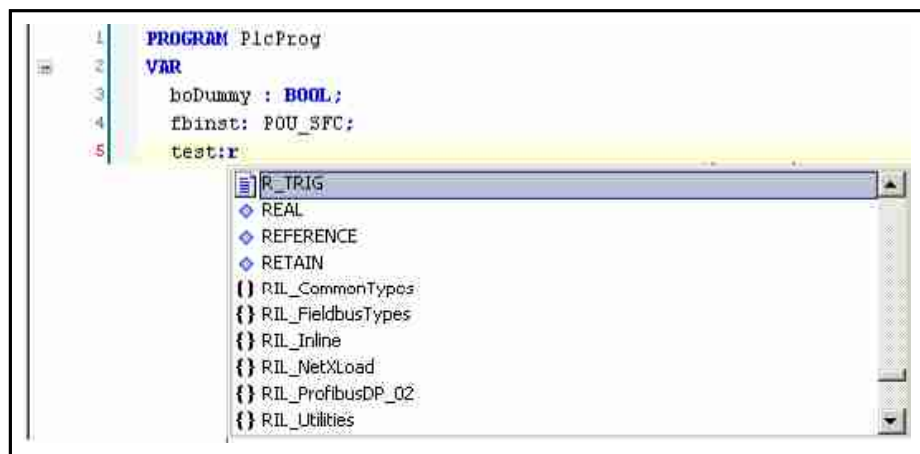


Fig.3-17: "List components" provides list selection (starting with 'r')

3.2 View

3.2.1 View, General Information

The menu item "View" allows the windows and toolbars within IndraWorks Engineering to be activated and deactivated.

- Project Explorer, main window to the left of the workspace
- Device library, main window to the right of the workspace
- *Other Windows*
 - Output
 - Messages, page 105
 - Search results
 - Tools, page 106
 - Cross reference list, page 107
 - Element properties..., page 108
- Toolbars, (IndraLogic selection), page 109
 - Bookmarks, page 109
 - IndraLogic, page 110, important commands for menu items "Create" and "Debug"
- Object properties in the Project Explorer, page 238.

Menu Items

If required, the menu structure can be changed using the dialog in **IndraWorks ▶ Tools ▶ Customize**.

3.2.2 View - Other Windows

Messages

Menu: **View ▶ Other Windows ▶ Messages**

This command opens a window in which messages with regard to the project are displayed.

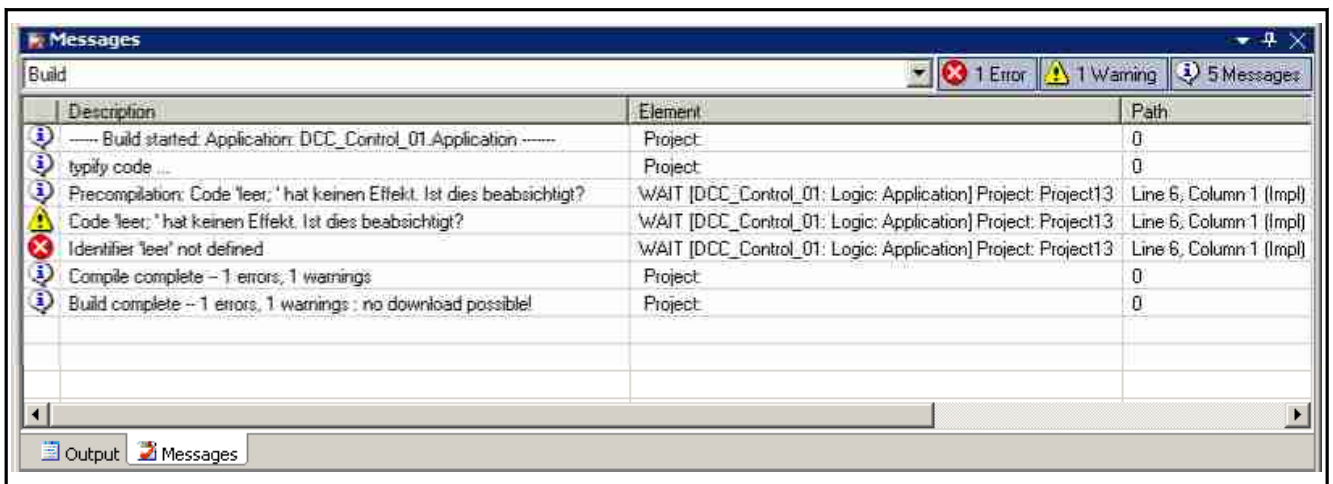


Fig.3-18: Window: Messages, example after "Generate code"

Messages can include

- Errors
- Warnings or
- Information

Messages are moreover categorized by component or functionality.

Messages regarding project syntax checks e.g. belong to the 'Precompile' category; messages for compiling the project to the 'Compile' category (e.g.



Menu Items

compilation errors, code size). Messages of the "Import", "Library Manager", etc. categories are possible as well.

A filter for the desired **message category** can be set in the selection list at the top of the message window (refer to the preceding figure) except for the **Pre-compile** messages which are always displayed in the area below the actual message table.

The messages in the selected category are output in table form and include the following information:

- Description (message text)
- Project (project name)
- Object (name of the associated project object)
- Position (position in the object that triggered the message, e.g. line number, network number, etc.).

If display of a certain message type is to be hidden or shown, use the buttons in the upper right-hand corner of the window. **Errors**, **Warnings**, **Information**, **Messages**. The buttons also show the currently available number of messages of the type.

Switch between the messages in the message window or jump from the currently selected message to the respective **position in the object**. For this purpose, the 'Next message', 'Previous message' and 'Go to source text position' **commands** are available.

Switch between the messages in the message window or jump from the currently selected message to the respective position in the object.

To do this, use these commands:

- "Next message" and
- "Previous message"

Double-click on a message entry in the table to move to the source text position.

The task list can also be sorted according to a variety of criteria using the "Sort by" command (context menu).

Choose from the following variants:

- "Time sequence"
- "Priority"
- "Description"
- "Origin" and
- "Position"

Double-click on a message entry in the table or click on the precompile message underlined in blue to move to the source text position.

Tools

Menu: **View ► Other Windows ► ToolBox**

This command opens a window with "tools" ("Toolbox") for the **currently active editor**.

By default, this ToolBox is available for graphical editors or for the visualization editor. It contains graphical programming elements that can be dragged into the editor window using "drag&drop".

Examples:

- ToolBox for [CFC editor](#), page 311
- ToolBox for [FBD/LD/IL editor](#), page 355
- Toolbox for the [Visualization editor](#), page 446.

Cross Reference List

This command opens a window in which the cross references for a specific project variable are listed. Cross references include the positions in the project where the variable is used.

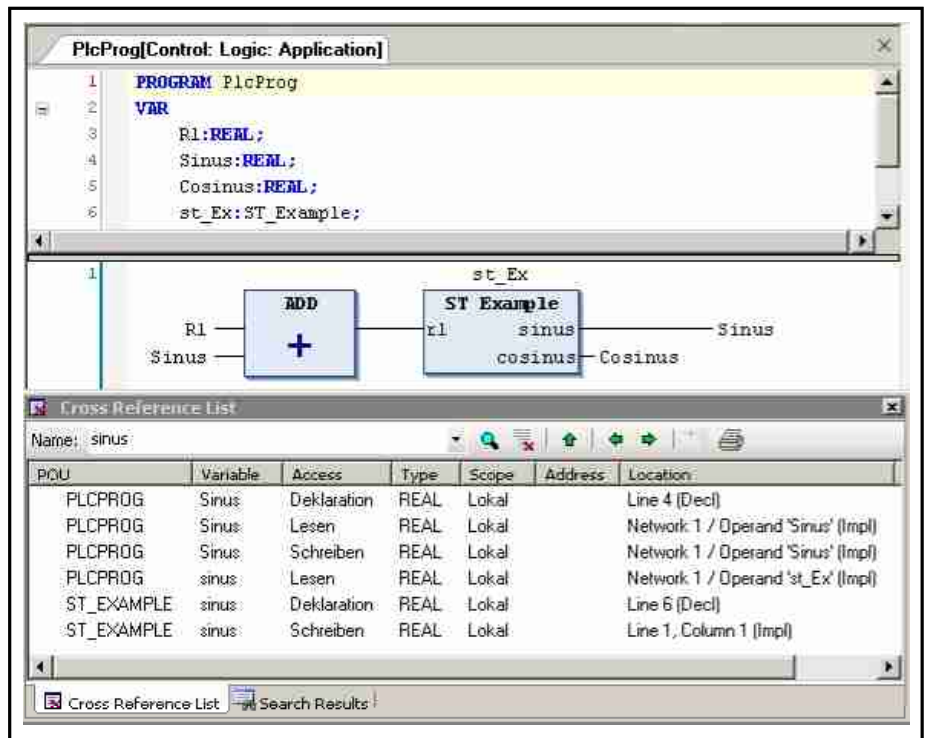


Fig.3-19: Example of a cross reference list

If **all cross references within the entire project** are to be listed: Jog the variable identifier manually or copy it from the editor window in to the **Name** field by means of copy&paste and confirm by means of the <Enter> key or click on the button

If **only the cross references within the same POU** are to be listed: Mark either the variable identifier in an editor window and drag it into the cross reference window using the mouse or activate the [Automatically update cross references when changing selection](#), page 205 option and place the mouse cursor in the identifier name in the editor window or select it. In this case, the identifier is automatically applied to the **Name** field and the cross references are listed below.

An **automatic update** of the cross reference list can be activated in **Tools** ► **Options** ► **IndraLogic 2G** ► **Smart coding** .

If a valid identifier is entered, the use positions found are displayed in table form with the following information about the variable:

| | |
|----------|---|
| POU | Name of the function block in which the variable is used. |
| Variable | Variable name |


Menu Items


| | |
|----------|--|
| Access | Use of the variable at the given position: Declaration/read/write/call |
| Type | Data type |
| Range | Validity range: global/local |
| Address | IEC address, if defined |
| Position | Position of the variable used within the editor (e.g. line number, network number) |
| Comment | Declaration comment for the variable |


Fig.3-20: Cross reference list, table header explanation


The list can be **sorted** alphabetically by column. By clicking on the column title, switch the sorting arrangement between ascending and descending order.


Double-clicking a line in the cross reference list **opens the respective POU** and selects the position identified. This corresponds with the use of the

 button (**Show position**) if an entry in the list is selected.

Use  (**Show previous position**, shortcut: <Shift> + <F4>) to jump to the previous entry in the list.

Use  (**Show next position**, shortcut: <F4>) to jump to the next entry in the list.


Use  (**Go to definition**, shortcut: <F2>) to jump to the position where the respective variable is declared. To do this, the corresponding declaration editor opens and the variable is highlighted there.

The  button corresponds to the command [Display cross references, page 98](#), which outputs a **display of the current cross reference list in the message window**.

This can be useful if the current list is to be kept available even though the automatic update is active and the list in the cross reference window can change.

View - Element Properties

View - Element Properties, General Information

Icon: 

Menu: **View** ▶ **Other Windows** ▶ **Element properties**


When this command is used, the main window "Properties" opens to the right of the workspace.

If an appropriate object is highlighted in the workspace, its properties appear in this window.

- [Element properties of sequential function chart objects, page 108](#)
- [Element properties of visualization elements, page 109](#)

Using the command again closes this window.

Element Properties - Sequential Function Chart Object (SFC)

Icon: 

Menu: **View** ▶ **Other Windows** ▶ **Element properties**

This command opens the Properties window for the currently selected SFC element.

Menu Items

Element properties such as name, comment, step times and associated actions are displayed in a subdivided table.

These can be edited in the 'Value' column:

Go to input mode by clicking the mouse and for steps place or remove a checkmark in the checkbox to enable or disable the "Initial step" property.

See also the detailed information for the individual

- [Element properties, page 404.](#)

Element Properties - Visualization Element

Icon: 

Menu: **View** ▶ **Other Windows** ▶ **Element properties**

This command opens the Properties window for the currently selected visualization element.

The element properties are displayed in a table, sorted in groups. Click on the plus or minus sign in front of the group name to display or hide the related parameters.

Which properties can be configured depends on the visualization element.

A description of the individual parameters is located in the [visualization editor, page 451](#),.

3.2.3 View - Toolbars

View - Toolbars, General Information

The IndraLogic PLC programming system provides the following toolbars that are activated if editors associated with the programming system support it and are active:

- [Bookmarks, page 96](#), bookmarking functionality in text-based editors
- [IndraLogic, page 110](#), frequently used functionality of the menu items **Create** and **Debug**.



Toolbars can be changed based on user-defined needs in **Tools** ▶ **Customize** .

The IndraLogic PLC programming system uses the following toolbars in connection with the comparison of PLC objects:

- [Toolbar in the Comparison dialog, page 110.](#)
- [Toolbar in the Comparison results dialog, page 111.](#)

Bookmark, Overview



The "bookmarks" toolbar provides bookmarking functionality for text editors. It can be activated using **View** ▶ **Toolbars....**

Bookmarks can be added to lines to facilitate navigation in long programs.



Bookmarks remain when closing the editor window.

If IndraWorks Engineering is closed, all bookmarks are removed.

The commands described below allow the setting of bookmarks and the navigation with them.

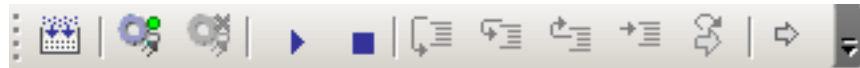
Menu Items

Commands:

-  Switching Bookmarks, page 96,
-  Next Bookmark, page 97,
-  Previous Bookmark, page 97
-  Deleting Bookmark, page 97

IndraLogic Overview

Icons:



The "IndraLogic" toolbar contains frequently used commands for the menu items **Create** and **Debug**.

-  **Create** ▶ [Compile, page 124](#); the command starts the compilation for the application that is currently active.
-  **Debug** ▶ [Login, page 127](#); this command connects the programming system, i.e. the active application with the control, thus establishing the **online mode**.
-  **Debug** ▶ [Logout, page 128](#); this command ends the connection between the development system and the control, causing a return to **offline mode**.
-  **Debug** ▶ [Start, page 133](#); this command starts the application on the control.
-  **Debug** ▶ [Stop, page 133](#); this command stops the application on the control.
-  **Debug** ▶ [Procedure Step, page 142](#); this command carries out a sequence of defined steps in the program in online mode.
-  **Debug** ▶ [Stop, page 133](#); this command carries out a single step in the program in online mode.
-  **Debug** ▶ [Execute to Return, page 143](#); this command starts the processing of a program loop up to the return in online mode.
-  **Debug** ▶ [Execute to Cursor, page 143](#); this command starts the processing of program lines up to the cursor in online mode.
-  **Debug** ▶ [Determine Next Instruction, page 143](#); this command specifies the instruction that is to be processed next.
-  **Debug** ▶ [Display Next Instruction, page 143](#); this command shows the instruction that is to be processed next.

Comparison, Toolbar

The toolbar in the upper part of the dialog serves the navigation and provides the following functions for comparison/merger:

Menu Items








| Symbol | Description |
|---|--|
|  | Show previous element: Clicking this symbol navigates to the previous element in the history of the already displayed elements. |
|  | Show next element: Clicking this symbol navigates to the next element in the history of the already displayed elements. |
|  | Select next difference: Clicking this symbol selects the next difference. |
|  | Select previous difference: Clicking this symbol selects the previous difference. |
|  | Preselect for the merger: Clicking this symbol preselects the selected line for the merger. Note: If no merger is possible for the selected line, no preselection can be made. |
|  | Reset preselection for merger: Clicking this symbol no longer preselects the selected line for the merger. |
|  | Start merger: Clicking this symbol carries out the merger for all preselected lines. |

Fig.3-21: Toolbar

Comparison Results, Toolbar

The toolbar in the upper part of the "Comparison results" dialog serves as filter and provides the following functions:




| Symbol | Description |
|---|--|
|  | Show all elements: Clicking this symbol shows all differing and complying elements. |
|  | Show only differing elements: Clicking this symbol shows only the differing elements. |
|  | Show only complying elements: Clicking this symbol shows only the complying elements. |

Fig.3-22: Toolbar

3.3 Project

3.3.1 Archiving and Restoring

Archiving IndraLogic Projects

A detailed description of how to archive projects is located in the IndraWorks help under Archiving Projects.

When archiving IndraLogic projects, a check determines which libraries and device descriptions are required for the project.

These are saved locally or on the FTP server in the archive.

Restoring IndraLogic Projects

A detailed description of how to restore projects is located in the IndraWorks help under Restoring Projects.

Menu Items

When restoring IndraLogic projects, the libraries and device descriptions required for the project to be restored are determined from the archive.

If these files are not in the [Library Repository...](#), page 185, or in the device database, the respective repository is supplemented from the archive when the project is opened for the first time after restoration.

3.3.2 Displaying PLC Objects of an External IndraWorks Project



Icon:

Menu: **Project ► Display PLC objects of an external IndraLogic project**

The command allows to open an external project and to copy PLC components from its content.

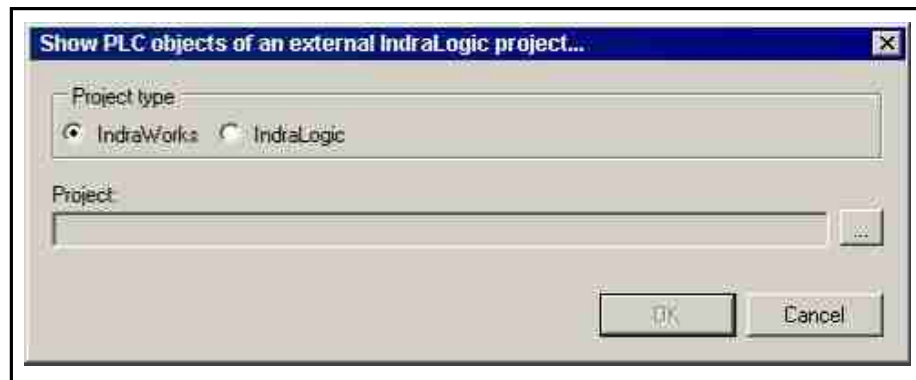


Fig.3-23: Dialog: Selecting the source

Only the "IndraWorks" and "IndraLogic" project types are permitted.

With the "IndraWorks" project type, the standard dialog for selection of an IndraWorks project opens after clicking on

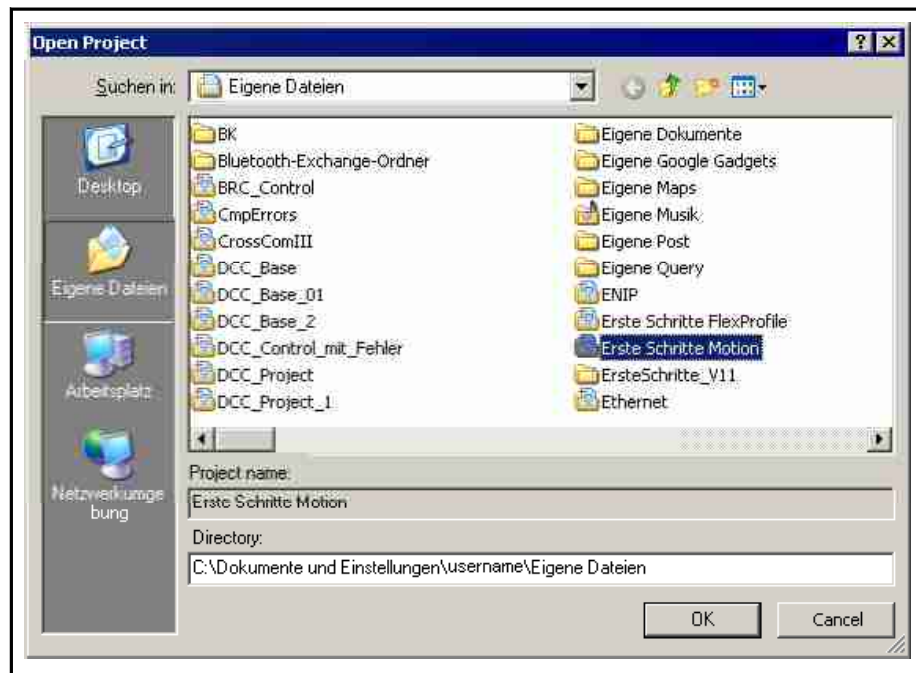



Fig.3-24: Dialog: Selecting the IndraWorks project

Menu Items

With the "IndraLogic" project type, you have to navigate to the desired IndraLogic project after operation of , e.g. **First steps Motion ► IndraLogic ► IndraLogic.project**.

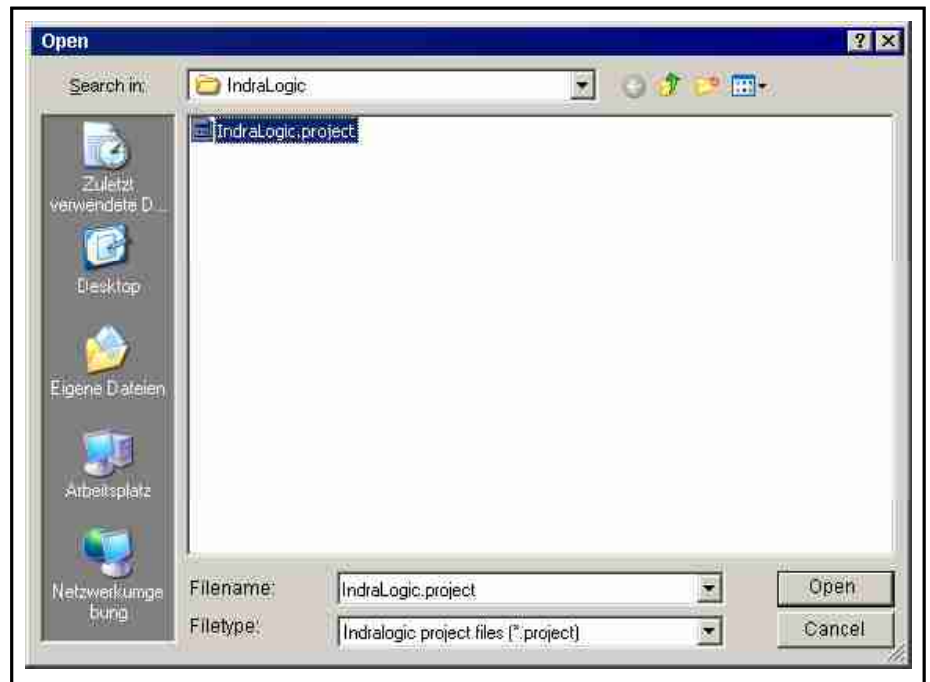


Fig.3-25: Dialog: Selecting the IndraLogic project

For both project types, a dialog appears to select the objects to be opened:

Menu Items

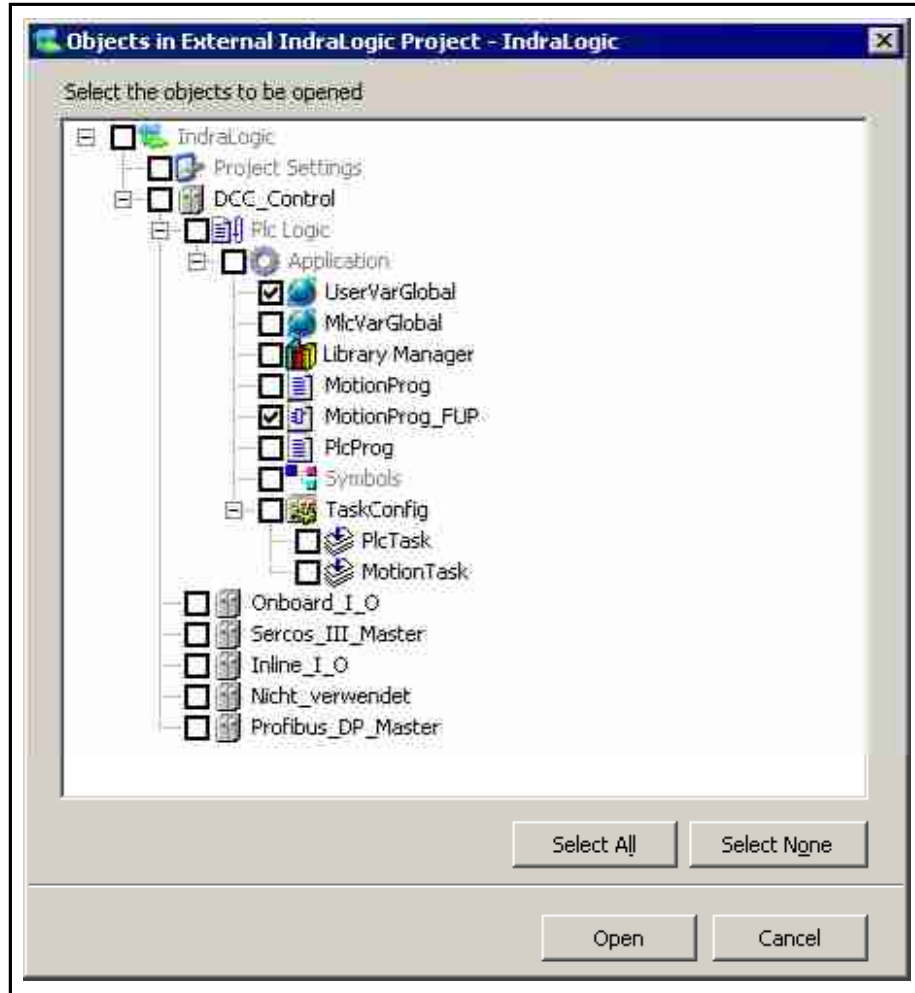


Fig.3-26: Dialog: PLC object selection

Objects that can be selected (i.e. displayed) have black font, all others are shown in gray font.

Using the "Select All" or "Select None" button, the user can mark all objects for opening or remove their marking.

Clicking the button in front of a selectable object with the mouse marks individual objects for opening.

If the user presses the "Open" button, the selected objects are opened for read access. The user can now e.g. mark text passages in these objects and copy them to objects of the open project.

The "External:" entry in the title of the document window shows the user that the opened object comes from an external project.

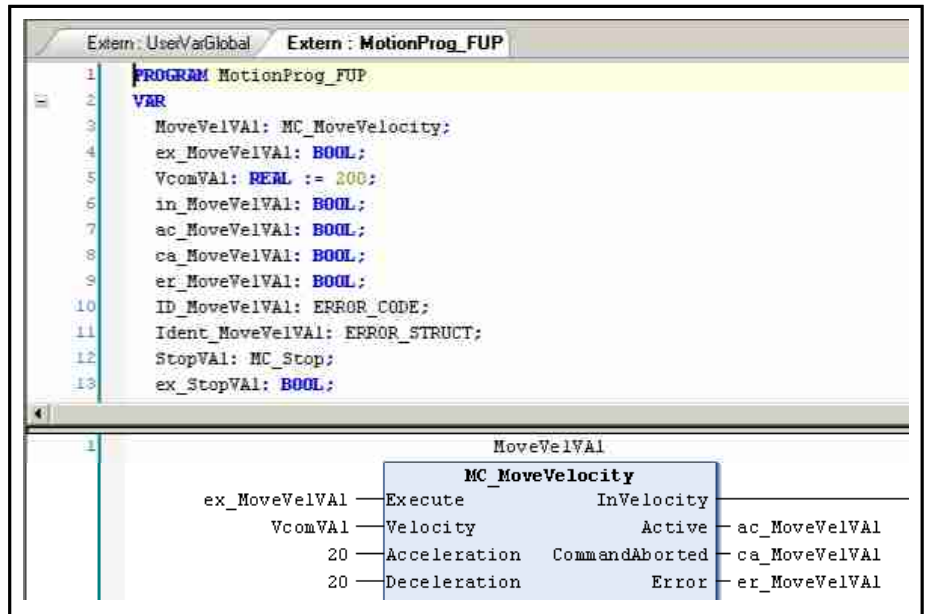


Fig.3-27: Opened external PLC object

3.3.3 Data Transfer

Data Transfer, Overview

IndraWorks provides a way to export **complete projects** and then to import them again.

These projects contain components associated with the related system.

An export is performed from the respective object where the menu items "Export" and "Import" can be called in its context menu.

See also Exporting and importing project data, Rexroth IndraWorks 12VRS, Engineering, DOK-IWORKS-ENGINEE*V12-APxx-EN-P.

IndraLogic itself provides a way to import **IndraLogic projects** created using different platforms. Blocks, data types, visualizations and resources (global variables, variable configuration, trace recording, control configuration, task configuration, etc.) can be imported.

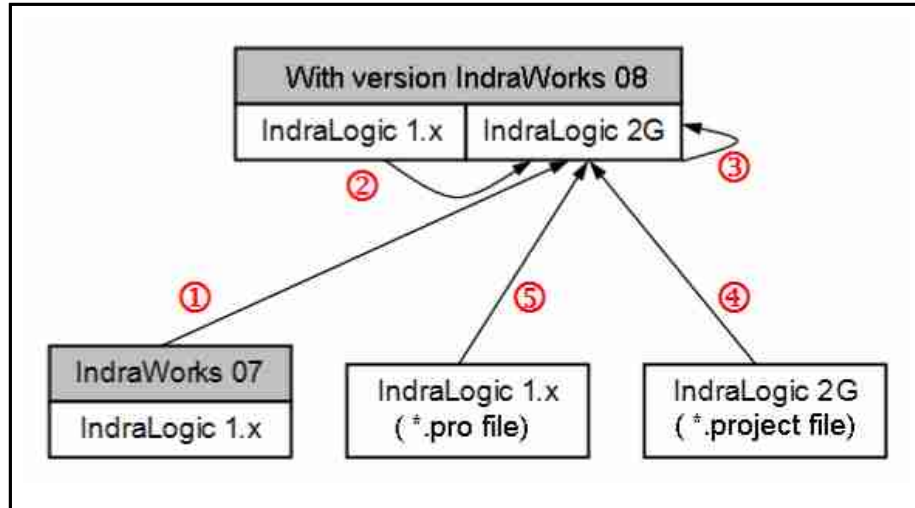
In order for projects to be imported, the following prerequisites have to be met:

1. IndraWorks Version 8 or later
2. Open IndraLogic 2G project
3. IndraLogic 2G libraries
4. Device description files

The "Application" node context menu can be used to import existing IndraLogic 1.x projects (both stand-alone projects and projects embedded in IndraWorks) and IndraLogic 2G projects into IndraLogic 2G projects.

The following figure clarifies the relationships.

Menu Items



- ① IndraLogic 1.x projects embedded in IndraWorks (Version 7) are automatically converted and then imported.
- ② IndraLogic 1.x projects embedded in IndraWorks (Version 8) are automatically converted and then imported.
- ③ IndraLogic 2G projects embedded in IndraWorks (Version 8) are imported. In this case the project is not converted, since both file types are identical.
- ④ Stand-alone IndraLogic 2G project files (*.project files) are imported.
- ⑤ Stand-alone IndraLogic 1.x project files (*.pro files) are automatically converted and then imported.

Fig.3-28: Schematic representation of various imports

Importing IndraLogic 1.x Projects

Importing projects To import an IndraLogic 1.x project, highlight the "Application node" in the Project Explorer and select "Data Import..." from the context menu. Alternatively, also import projects using the main menu **Application ▶ Data Import...**



Note that you can have several applications in one IndraWorks project. The data transfer applies only to the highlighted "Application" nodes.

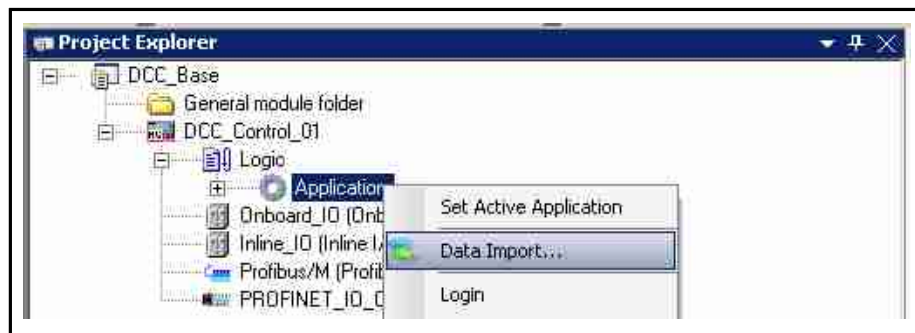


Fig.3-29: Importing IndraLogic projects

The "Data Import..." dialog appears.

Menu Items

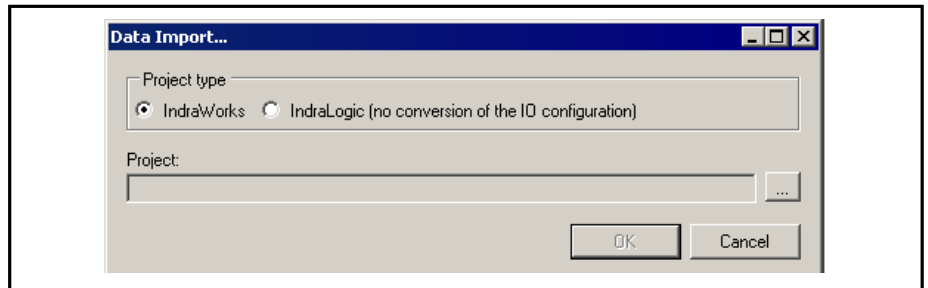


Fig.3-30: Selecting project type and project path

Under "Project type" select the type of your source project.

- **IndraWorks:**
In this case, an IndraLogic project (*.project) is embedded in an IndraWorks project.
- **IndraLogic (IO configuration is not converted)**
In this case, a stand-alone IndraLogic project file (*.pro or .project) is the source project.



Note that the control configuration is not imported for stand-alone IndraLogic 1.x projects (*.pro).

Use the  button to open the "Open Project" dialog and move to the project to be imported.

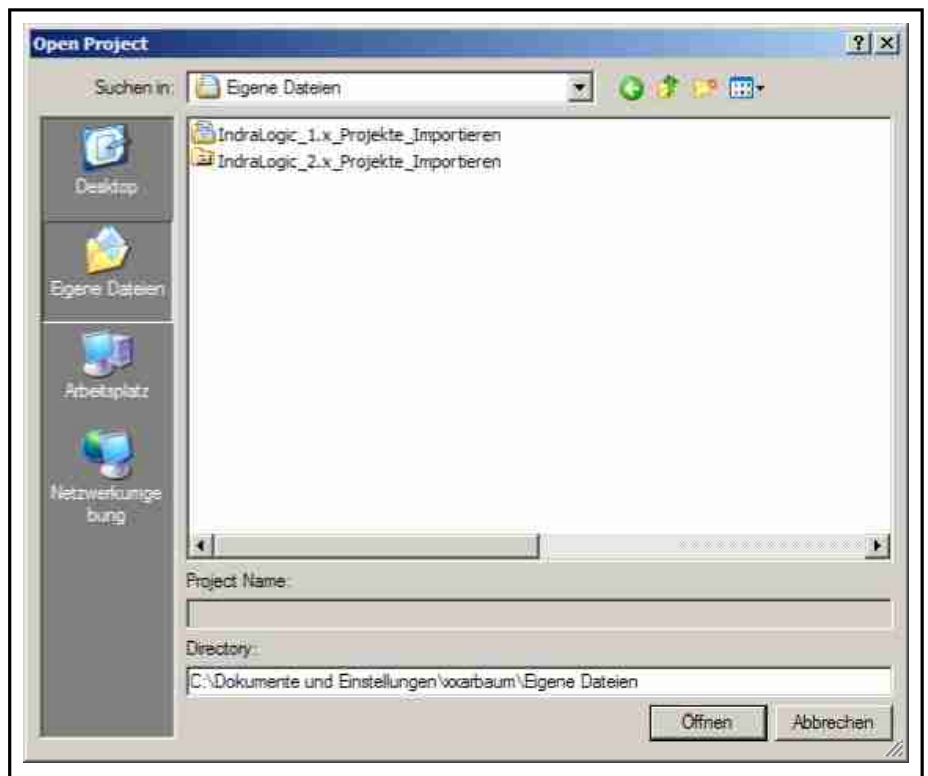


Fig.3-31: File selection

Confirm your selection with "Open".

Menu Items



Before the import takes place, the project is recompiled. Compilation errors might occur (e.g. if a library is missing), which can be ignored in terms of the project import.

The download for the project starts. In a later step, device descriptions can be installed to be able to ignore the following or similar error messages with "No".

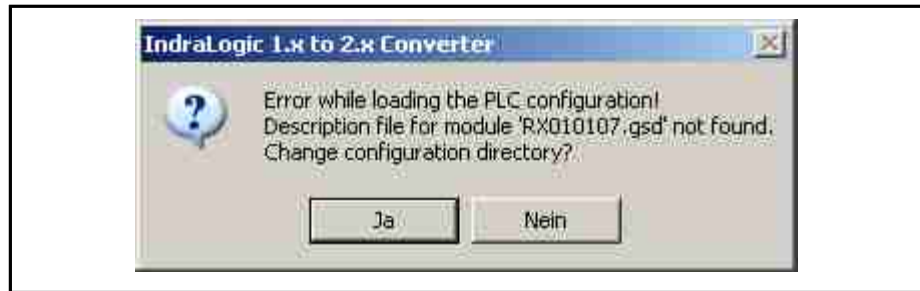


Fig.3-32: Error message that might appear while the PLC configuration is loading

The download includes a search for libraries required for the project. If the required libraries are not found, the following or a similar dialog appears.

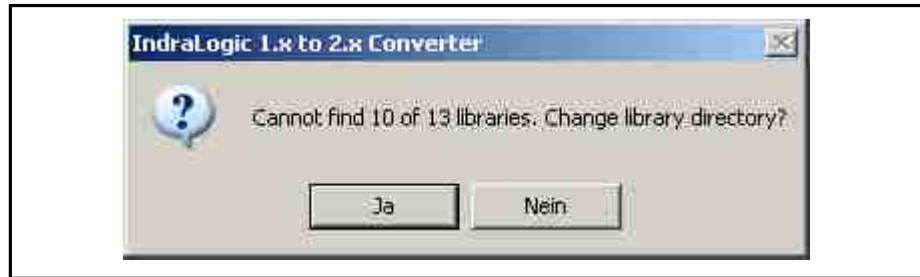


Fig.3-33: Dialog for libraries that are not found

To enter the library path for the respective library, click "Yes". The "Options" dialog opens automatically in which the paths for the requested libraries can be entered.

Library path

Enter the correct path in the "Options" dialog.

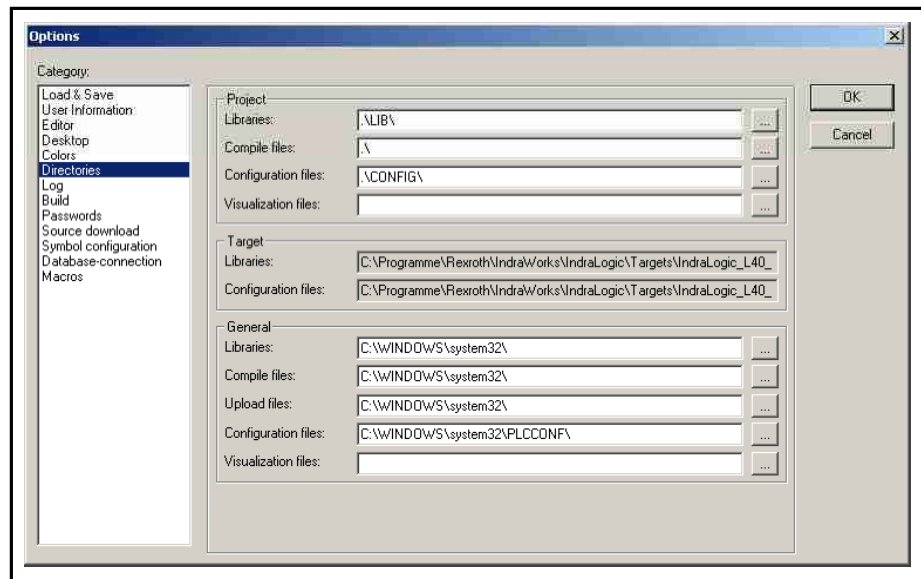


Fig.3-34: Specifying library paths

Menu Items

- In the "Project" and "General" areas, in the "Libraries" selection field, select the directories that IndraLogic should search and use. The entries in the "Project" area are saved with the project and those in the "General" area apply to all projects.
- The "Target" area includes a listing of the directories for libraries and configuration files that are set in the target system, e.g. by input into the target file. These fields cannot be edited.

Use "OK" to confirm your entries.

Converting library references

If there are libraries included in the project that do not yet have conversion assignment saved in the "Library options", page 222., the "Library Reference Conversion" dialog appears for defining how these references are to be converted:

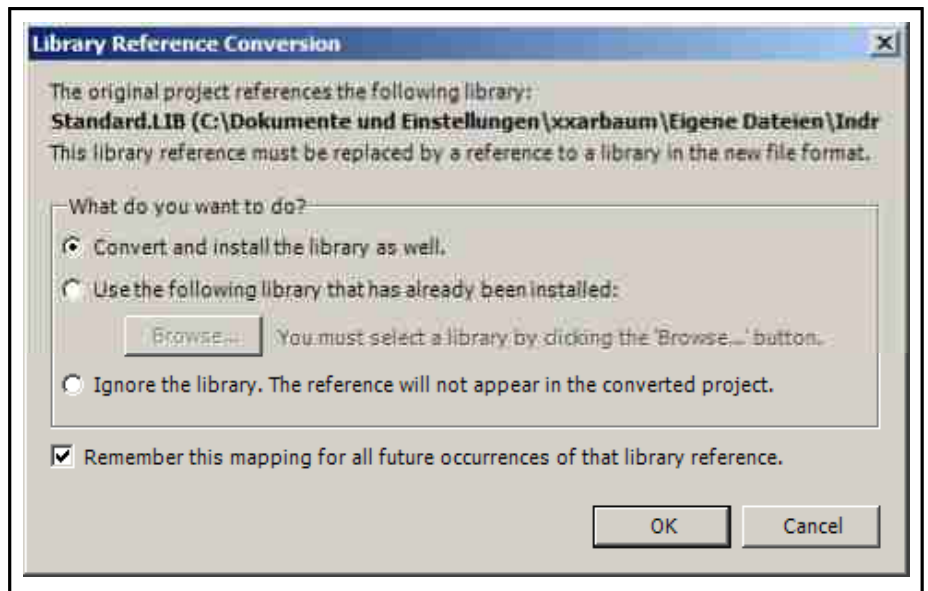


Fig.3-35: "Library Reference Conversion" dialog

The following options are possible:

- **Convert and install the library.**
If this option is selected, the integrated library is converted to the new format and remains referenced within the project. It is automatically installed in the library repository in the "Other" category and remains used. If the project information required for an installation (title, version) does not come with the library, you are requested to enter this information in the "Enter project information" dialog.
- **Use the following library that has already been installed.**
Select this option if a library that has already been installed is to be used for this reference. Use the "Browse" button to open the "Select Library" dialog. Select the desired version of one of the installed libraries here.
This corresponds with the configuration of the version handling in the "Library Properties" dialog (see "Library manager properties", page 230).
"*" means that the latest version of the library available in the system is always used in the project. The list of available libraries is structured as described for the "Library repository dialog", page 185.,
It can be sorted according to company and category.

Menu Items

- Ignore the library. The reference does not appear in the converted project.**

If this option is selected, the library reference is removed and the library is no longer included in the converted project.
- Remember this mapping for all future occurrences of that library reference.**

If this option is selected, the settings made in this dialog are also used for future project conversions as soon as the respective library is referenced.

For standard libraries, an assignment to the new libraries is already specified.

Select the desired option and confirm with "OK".

Enter project information (product information)

In the "Enter project information" dialog, the imported library is uniquely identified. Fill in the "Title", "Version" and "Company" fields here.

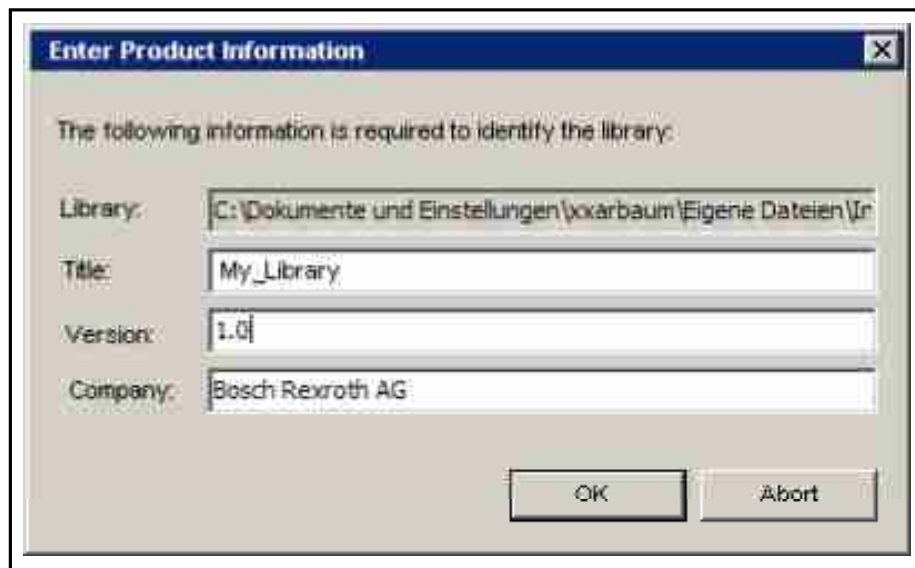


Fig.3-36: Enter project information (product information)

- Title Specify the desired name for the library.
- Version Input a unique version of the library.
- Company Input the company name.

Use "OK" to confirm your entries.

Import objects

In the "Current project object as import target:" window, please select the objects to be imported.

Menu Items

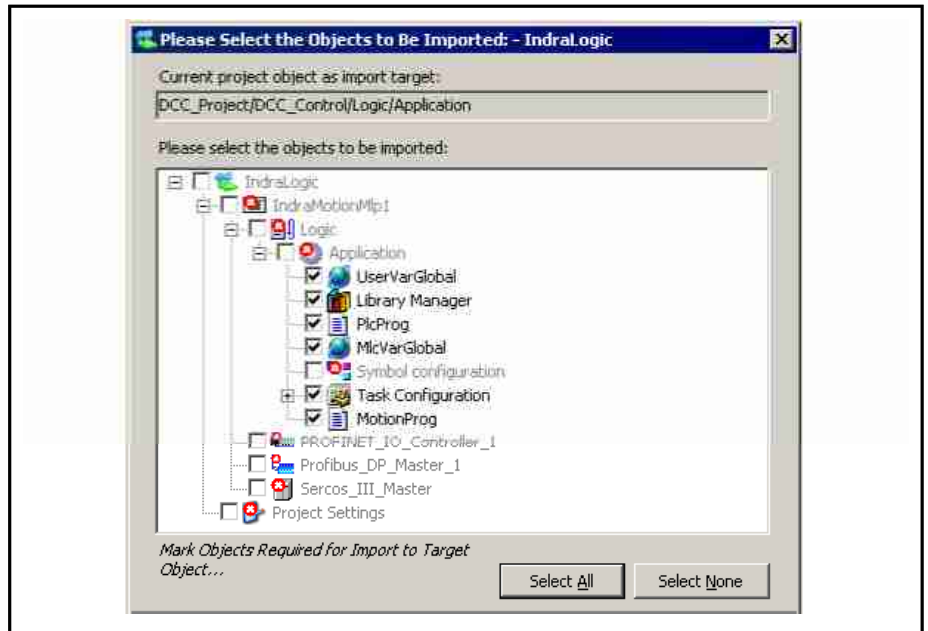


Fig.3-37: Selecting the objects to be imported

- Use the "Select All" button to highlight and import all of the objects.
- Use the "Select None" button to deselect objects that have been highlighted.



For stand-alone IndraLogic 1.x projects (*.pro), the I/O configuration is not imported into the target system.



If you select objects that already exist in the target project before the import, a selection dialog prompts whether to overwrite these objects; see "Overwriting objects", page 121.

Overwriting objects

Objects that already exist in the target project can be overwritten by placing a checkmark in the following selection dialog.

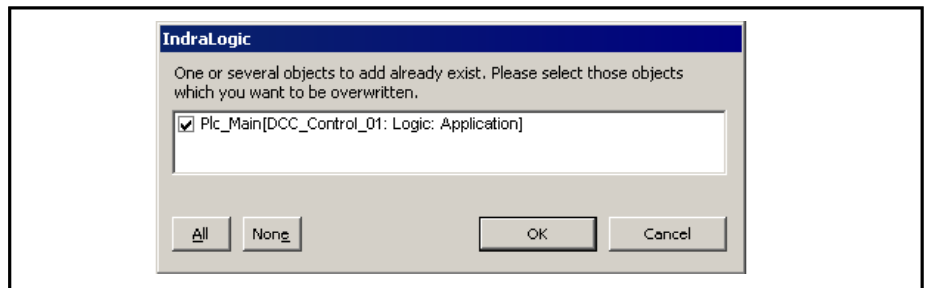


Fig.3-38: Overwriting objects

- Use the "All" button to highlight all of the objects available in the dialog. These objects are overwritten in the target project.
- Use the "None" button if you do not wish to overwrite any of the objects in this selection dialog.

Click on "OK" to complete the object import. The imported objects are added to the project. Finally, a message appears that indicates that the objects have been imported.

Menu Items

Importing IndraLogic 2G Projects

Importing projects To import an IndraLogic 2G project, highlight the "Application" node in the Project Explorer and select "Data Import..." from the context menu.

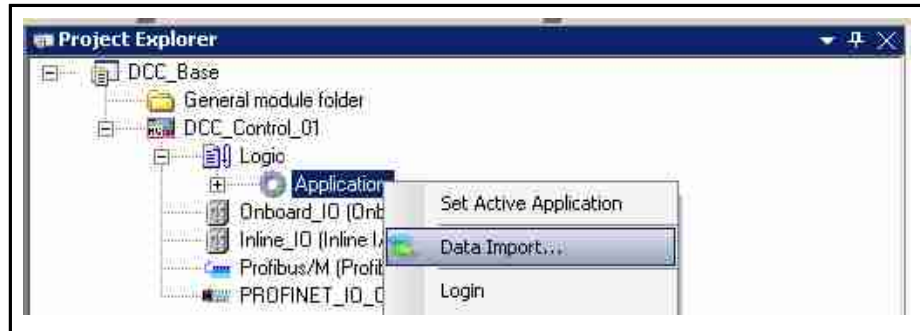


Fig.3-39: Importing IndraLogic projects

The "Data Import..." dialog appears.

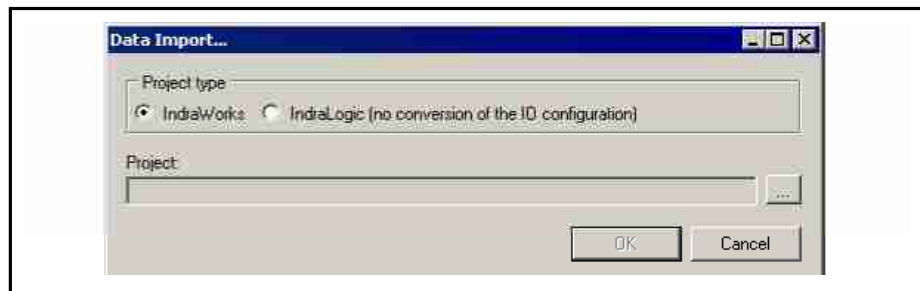


Fig.3-40: Selection dialog: Project import...

Under "Project type" select the type of your source project.

- **IndraWorks:**
In this case, an IndraLogic project is embedded in an IndraWorks project (*.project).
- **IndraLogic (IO configuration is not converted)**
In this case, a stand-alone IndraLogic project file (*.pro or .project) is the source project.

Use the button to open the "Open Project" dialog and move to the project to be imported.

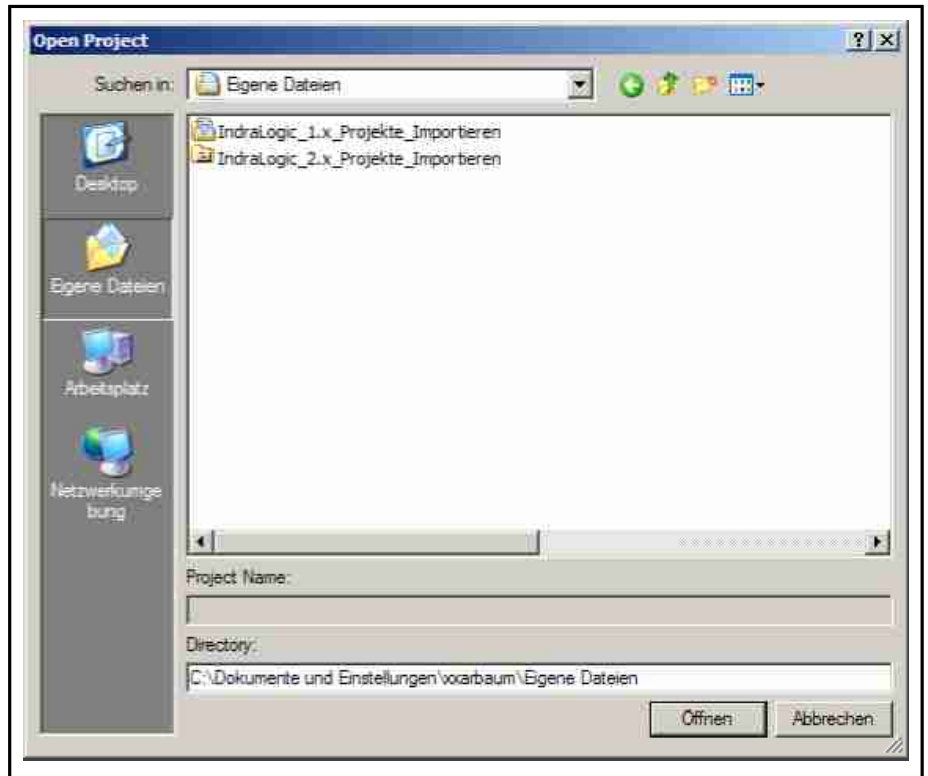


Fig.3-41: Project selection

Select objects

In the "Current project object as import target:" window, please select the objects to be imported.

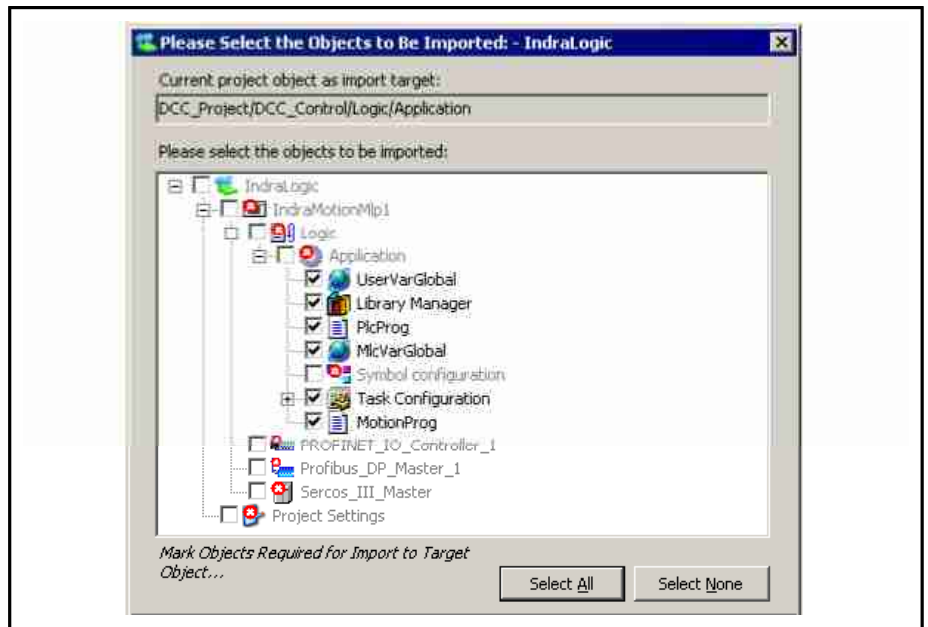


Fig.3-42: Selecting the objects to be imported

- Use the "Select All" button to highlight and import all of the objects.
- Use the "Select None" button to deselect objects that have been highlighted.

To start the import, click "OK".

Menu Items



If you select objects that already exist in the target project before the import, a selection dialog prompts whether to overwrite these objects; see ["Overwriting objects"](#), page 124.

Overwriting objects

Objects that already exist in the target project can be overwritten by placing a checkmark in the following selection dialog.



Fig.3-43: *Overwriting objects*

- Use the "All" button to highlight all of the objects available in the dialog. These objects are overwritten in the target project.
- Use the "None" button if you do not wish to overwrite any of the objects in this selection dialog.

To complete the import, click "OK". The imported objects are added to the project, which is confirmed by a dialog.

3.4 Create

3.4.1 Create, Overview

The "Create" menu item provides functionalities to build a program, i.e. for compiling an application program.

The "Create" commands are used for code generation and for syntactic checks in all objects or in modified objects in the active application. Offline code generation is possible in order to be able to check for compilation errors before the code is loaded to the device.

The results are output in the message box, message category: Compilation

The "Clean" commands are used to delete the compilation information from the target system that was saved during the most recent download and during code generation. This affects [Online Change](#), page 80, for example.

If required, the menu structure can be reconfigured via the **IndraWorks ▶ Tools ▶ Customize ▶ Commands** dialog.

Commands:

- [Compile](#), page 124,
- [Recompile](#), page 125,
- [Generate code](#), page 125
- [Clean](#), page 125
- [Clean all](#), page 125

3.4.2 Compile

Icon:

Default shortcut: <Shift>+<F11>

Menu: **Create ▶ Compilation**

Menu Items

This command starts the compilation for the currently active application. This means that all objects for this application are subject to a syntactic check.



Note that **no** code is generated as it is the case when logging into the target system or downloading the application!

The compilation is carried out automatically before every login with changed program.

After the syntactic check is complete, error messages or warnings might be displayed within the message window in the "Compile" category.

If the number of errors/warnings exceeds 500, all other messages are skipped and a special message provides information about the situation.

Commands of the [Message window, page 105](#), category are available in order to navigate between the messages and between messages and source code.

If the program has not been modified since the most recent error-free compilation, it is not recompiled and the message "The application is current" is output in the message window.

However, if the syntactic check should nevertheless be repeated, use the [Re-compile, page 125](#) command.

3.4.3 Recompile

Menu: **Create ▶ Recompile**.

The command starts the compilation for the currently active application again, even if it was recently compiled without errors.

3.4.4 Generate Code

Menu: **Create ▶ Generate Code**

Generate code for test purposes with this command.

As with the login, compilation code is generated with the application in the control, but the code is not loaded into the control. In this way, the code can still be checked for compilation errors and corrected before it is used in online operation.

3.4.5 Clean

Menu: **Create ▶ Clean**

The command deletes the compilation information for the currently active application. This information was created and saved during the most recent [download, page 132](#), of the application to the target system.

After the cleaning is complete, an [online change, page 80](#), is no longer possible for the respective application.

First, the program has to be completely reloaded to the control.

3.4.6 Clean All

Menu: **Create ▶ Clean All**

The command deletes the compilation information for all applications,

After cleaning is complete, an [online change, page 80](#), is no longer possible for the respective applications.

First, the program has to be completely reloaded to the control.

See also [Clean, page 125](#).



Menu Items

3.5 Debug

3.5.1 Debugging, Overview

These commands influence the application program on the control in online operation.

By default they are available in the main menu "Debug".

If required, the menu structure can be reconfigured via the **IndraWorks ▶ Tools ▶ Customize ▶ Commands** dialog.

For security reasons, when calling the commands that are marked with an asterisk * below, users are always requested to confirm the selection.

An additional request for confirmation for the commands marked with two asterisks ** appears if the option [Secure online operation, page 239](#), is enabled in the communication settings for the respective device.

CAUTION

The extraordinary modification of variable values in an application running on the control might lead to unwanted behavior of the controlled system.

Depending on the controlled system, damage at the system and workpieces might result or the health and life of persons might be at risk.

Evaluate possible risks before writing or forcing variable values and take corresponding precautions.

CAUTION

The online change modifies the running application program and causes a restart.

Depending on the controlled system, damage at the system and workpieces might result or the health and life of persons might be at risk.

Ensure that the new application code still results in the desired behavior of the controlled system.

The commands:

- [Login, page 127](#)
- [Logout, page 128](#)
- [Generate boot application, page 129](#)
- [Logout current online user, page 129,,](#)
- [Edit object \(offline\), page 130,](#)
- [Load, page 132, *](#)
- [Online Change, page 133, *](#)
- [Start, page 133, **,](#)
- [Stop, page 133, **](#)
- [Single cycle, page 134, **](#)
- [Multiple download..., page 134**](#)
- [Reset warm, page 135*,](#)
- [Reset cold, page 135*,](#)
- [Reset origin, page 136*,](#)

- [Breakpoints, page 136,](#)
- [New breakpoint..., page 138](#)
- [Toggle breakpoint, page 140](#)
- [Breakpoint positions, page 138](#)
- [Call stack, page 141](#)
- [Monitoring, page 141](#)
- [Procedure step, page 142](#)
- [Single step, page 142](#)
- [Execute to return, page 143](#)
- [Execute to cursor, page 143](#)
- [Determine next instruction, page 143](#)
- [Display next instruction, page 143](#)
- [Write values for all online applications, page 144, **](#)
- [Force values on all online applications, prepare value, page 145, **,](#)
- [Cancel forcing for all value on all online applications, page 147**](#)
- [Add all forces to monitoring window, page 148,](#)
- [Sequence control, page 148,](#)
- [Display mode, page 151, **](#)

3.5.2 Login



Default shortcut: <Alt> + <F8>

This command connects the programming system, i.e. the "active application", [page 66](#), with the target system (control or simulated device), creating the **online mode**. To achieve this, the [communication settings, page 67](#), for the device have to be configured correctly.

If the "Login" command is called from the online mode, the currently active application is concerned.

If the command is called from the context menu if an application is selected in the Project Explorer, login is carried out with this selected application even if it is not set as "active application".

The following situations are possible in case of login with the currently active application (without errors, communication settings correct):

- **The application is not available on the control yet:** Confirmation of the download ([Load, page 132](#)) is requested. A dialog with the following text appears: "<Application name> application does not exist on the control. Do you want to create and load the application?" The "Details" button in this dialog leads to information on applications already available on the control.
- **The application is already available on the control** and has not been changed since the last loading. Login is carried out without further interaction with the user.
- **The application is already available on the control**, has, however, been **changed** since the last loading. You are asked whether you want to complete an [Online change, page 133](#), or load the entire application or login without changing the currently running application.

A dialog with the following text appears:

Menu Items

"The code has changed since the last download. What do you want to do?"

- Login with online change.
- Login with download.
- Login without any change."

The "Details" button in this dialog leads to information on the application changed in the programming system as compared to its previous version which is currently available on the control.

- **Another version of the application is already available on the control, is** however, currently **not** running. You are asked whether it is to be replaced. A dialog with the following text appears: "On the control, there is an unknown version of the < Application name> application. Do you want to replace it by the application in the project?"

The "Details" button in this dialog leads to information on the application in the IDE (integrated development environment = programming system) as compared to the one on the control.

- **A version of the application is already available on the control and is running.** You are asked whether you want to login and overwrite the currently running application nevertheless. A dialog with the following text appears:

"Warning: An unknown version of the <Application name> application is currently running on the control. Do you still want to load the current code and replace the existing application?"

The "Details" button in this dialog leads to information on the application in the IDE (integrated development environment = programming system) as compared to the one on the control.

Compilation before login

Before login and if the current application project has not yet been compiled since opening or since its last change, it is compiled.

This means that it is compiled in the logged off condition as in a [compilation run, page 124](#), and a compilation code is generated for the control, as well.

If errors occur during compilation, a message box appears with the following text: "Errors occurred during compilation. Would you like to log in without loading the program?" This way, choose to correct the errors first or to log in despite the errors in the most recent version of the application that might be on the control. The errors are output in the message box in the "Compile" category.

See also:

- [Compile, page 124](#).

Information on loading

If the project is loaded to the control completely at login or partially at online change, information appears in the message window on the generated code size, on the size of the global data, on the resulting memory requirement on the control and in case of online change also on a list of the respective function blocks.

3.5.3 Logout



Default shortcut: <Ctrl>+<F8>

This command ends the connection between the development system and the target system (control or simulated device) causing a return to **offline mode**.

3.5.4 Generating a Boot Application

This command is available in online mode for generating a boot application. A boot application (boot project) is started automatically when the control is started. It is named <ApplicationName>.app

Use in online mode:

The boot application is saved on the target device with the name **<ApplicationName>.app**.

Use in offline mode:

The standard dialog for saving a file opens.

Select a directory for saving the current application as a boot application in file format to load it to a target system at a later time. The file is a "boot application" type file and the extension ".app" is automatically added to its name. After confirming with "Save", another dialog appears, in which you are asked if a "compilation protocol", page 80 that might be present in the project directory should be overwritten:

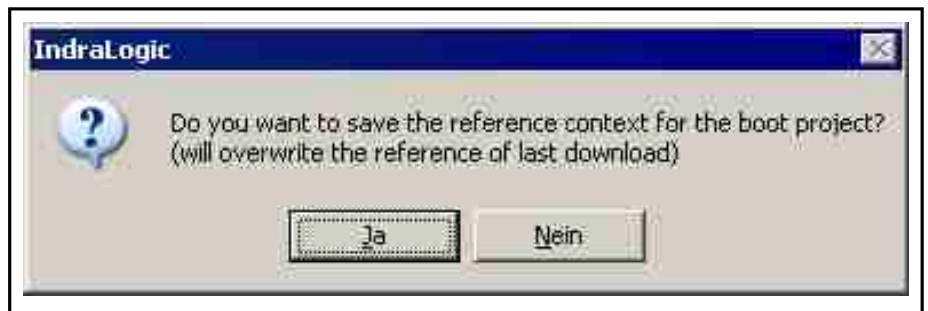


Fig.3-44: Dialog for saving the compilation information for the boot application

Select "Yes" if you plan to transfer the new boot project to the control with an external tool and you still want to log into the application without being forced to download it again.

A compilation protocol from a previous download would not match the newly created project, since it contains code and references from the previous application.

3.5.5 Logging out the Current Online User

Icon:

This command (User management) logs of the user currently logged in to the project or the library. A corresponding message is output only if currently no user is logged in. Otherwise, no dialog and no message appears.

If the user is currently logged in to several projects or integrated libraries (not necessarily using the same user account), the following logout dialog is opened.

Menu Items



Fig.3-45: Logout dialog

Via the selection list in the **Project/Library** field, you can now select the project or the library from which the **Current user** is to be logged off.

The toolbar always shows the user currently logged into the project.

3.5.6 Editing Objects Offline in the Online Mode (Hijacking)

This command is only available in online mode with running or stopped application.

If the user is logged in with an application, all opened PLC objects belonging to this application are displayed in online mode. If another object is opened, it is also displayed in the online mode.

The user can now in the context menu select the **Edit object (offline)** menu item in order to also open the selected object in the editable mode.

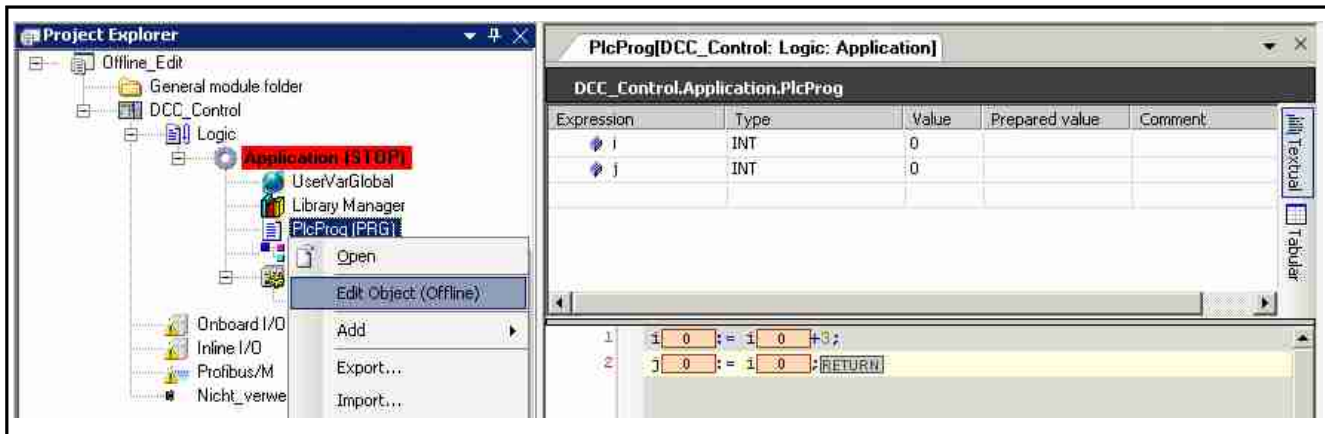


Fig.3-46: Opening the PlcProg object for offline editing

Alternatively, this command is also available in the **Debug ► Edit object (offline)** menu. It affects the POU selected in the Project Explorer.

As a result, the user has opened the object online and in the editable mode.

Menu Items

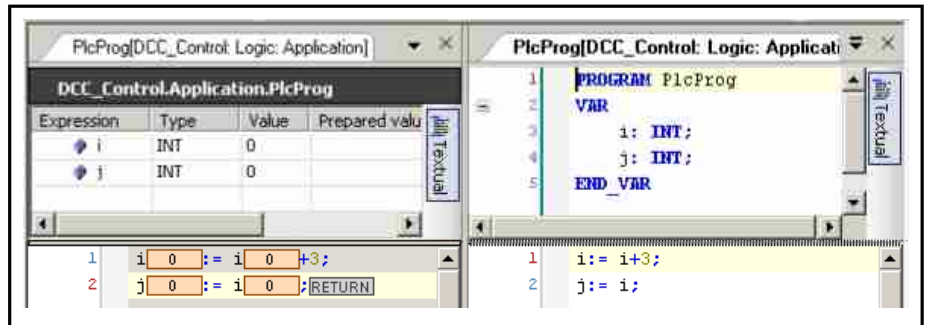


Fig.3-47: POU PlcProg online (left) and offline in the edit mode (right)

Any change in the right window is immediately displayed in the left window as well **without** becoming effective in the program code or in the control.

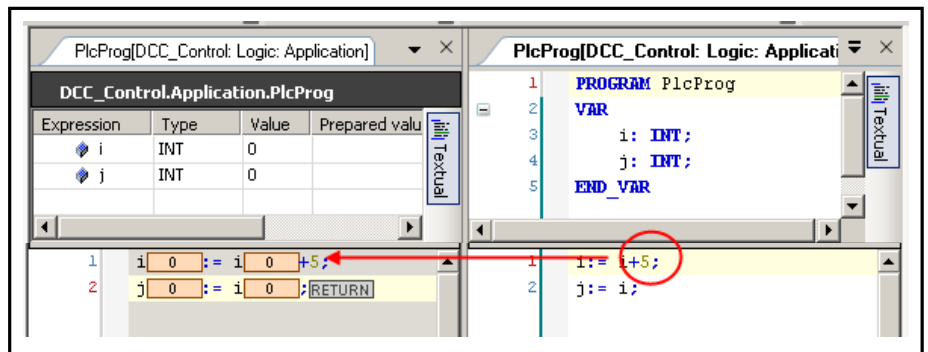


Fig.3-48: Offline change completed

The changes made in the offline editor can be transferred to the control directly via the

- **Debug ▶ Online change** or
- **Debug ▶ Loading**

menu commands, i.e. without log off and log in.

In the example, only the step width changes at the running application after the **online change**.

After **loading**, the variables are re-initialized and after application (re)start, counting is started at zero with the new step width.



Note the following points before performing an online change:

- Is the modified code free of errors?
- Application-specific initializations (reference motion, etc.) are not executed, since the machine retains its status.
- Can the new program code really work without re-initialization?
- Pointer variables retain their value from the last cycle. If it is pointed to a variable that changed in size, the value is no longer correct. For this reason, ensure that pointer variables are re-assigned in every cycle.
- If the active step in an SFC chart is removed, the chart remains inactive.

At saving, the changes made are also retained at the interface in the project without "Online change" or "Loading".

Menu Items

If the project is under **version control** and the object opened "offline" is checked in, it is brought into the "hijacked" state analogously to other objects in the logged in state.



Fig.3-49: Display of the changed object in the Project Explorer

3.5.7 Load

This command is available in online mode. It triggers the [compilation, page 124](#), and compilation of the [active application, page 66](#). This means that in addition to a syntactic check, application code is generated and loaded to the control.



All of the variables, with the exception of persistent variables, are re-initialized.

The following situations are possible:

If there is no application on the control yet, the user is asked if the application is to be loaded to the control.

After the user confirms with "Yes", the "download" is performed.

A matching dialog appears during [Login, page 127](#).



Fig.3-50: Message box, load application?

If other applications are on the control, the user is first asked if they are to be deleted. Then the application is loaded. See the related description for the same situation with the "Login" command.

If a different version of the same application is already on the control, the active application is loaded, overwriting the previous version. If the versions do not differ from each other, a dialog appears that indicates this situation and the application is not reloaded.



Fig.3-51: Message dialog, no download

3.5.8 Online Change

⚠ CAUTION

The online change modifies the running application program and causes a restart.

Depending on the system controlled, damages at the system and workpieces can result or the health and life of people can be put at risk.

Ensure that the new application code still results in the desired behavior of the controlled system.

This command causes an explicit "online change" to be carried out for the "active application", page 66,.

Online change means that only the modified parts of an application already running on the control are loaded to the control. This cannot be done after "Clean all" or "Clean"!

The cleaning deletes the compilation information that is automatically saved during each compilation (build, code generation) and is the basis for an on-line change.

An online change is automatically provided if login occurs with an application that is already running on the control but has been modified since the most recent download in the programming system.

See further information on "code generation and online change", page 80.

3.5.9 Start

Icon: ▶

This command starts the application on the control.

If the command is called from the online mode, the currently **active application** is concerned.

If the command is called from the context menu if an application is selected in the Project Explorer, login is carried out with this **selected application** even if it is not set as "active application".

3.5.10 Stop

Icon: ■

This command stops the application on the control.

Menu Items

3.5.11 Single Cycle

This command causes the active application to be executed for a single cycle.

3.5.12 Multiple Download...

This command is available in online or offline mode. It triggers the [compilation, page 124](#), of and [code generation, page 80](#), for all applications included in the project intended by the user. This way, in addition to a syntactical check of these applications, the related code is also generated and loaded to the respective control. For each application selected for download, a corresponding [compilation protocol, page 80](#) named <projectname>.<device-name>.<application ID>.compileinfo is generated in the project directory.

After the "Multiple download..." command is enabled, a dialog window opens that displays a list of all of the applications included in the project:

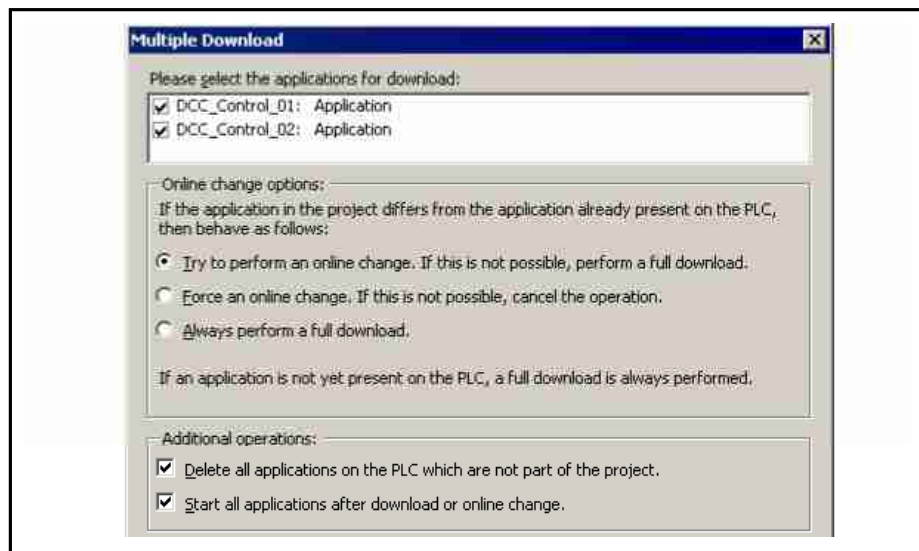


Fig.3-52: "Multiple Download" dialog

An application can be selected for download by placing a check in the respective checkbox. Several applications may be selected, even if they are to be loaded to different controls. By default, either all of the checkboxes are selected or only those for the applications selected by the user for the most recent download.

If a previous version of the selected application already exists on the control and it differs from the current version, the following options appear for selection:

- **Try to perform an online change. If this is not possible, perform a download.** This option is selected by default. If it is selected, the modified parts of the selected applications on the control are changed and only the newly created parts of the respective applications are loaded to the control.
- **Force an online change. If this is not possible, cancel the operation.** If an online change for (at least) one of the selected applications cannot be completed (e. g. if the command 'Clean all' or 'Clean application' was previously executed), no download is performed.
- **Always perform a download.** All of the parts of the selected applications are loaded to the control without taking existing versions into account.

Selected applications that do not yet exist on the control are automatically downloaded to the related control.

Menu Items

In addition, by selecting the corresponding checkbox, users can specify if

- Old applications that are no longer part of the project are to be deleted from the control
- The selected applications are to be started after the download/online change.

Note that PERSISTENT type variables are not initialized in general. However, if the data layout was changed, the persistent variables are automatically re-initialized.

After having confirmed the settings in the dialog with "OK", a syntactical check of all of the selected applications is carried out. Then, for each application, the communication with the related control is verified before the download is carried out.

After the download is complete, a list of the selected applications appears, which contains detailed information about the operations performed for each application.

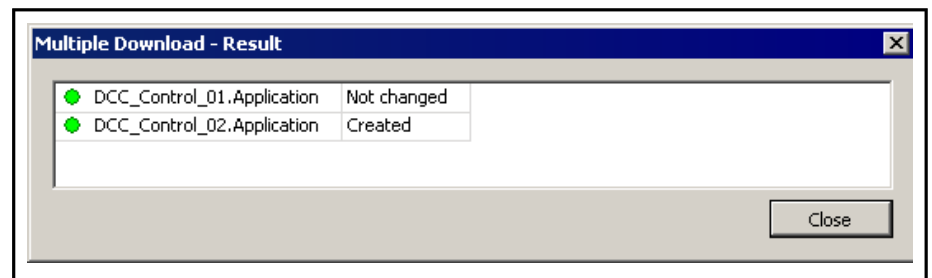


Fig.3-53: "Multiple download - Result" dialog

3.5.13 Reset Warm

This command is available in online mode. With the exception of the [remnant variables \(retain, persistent\), page 519](#), it resets all variables to their initialization value.

If variables were initialized with specific values, they are reset to these exact values. All other variables are reset to the default initialization values (for example, integer variables are reset to 0).

As a safety precaution, IndraLogic requires another user confirmation before all variables are overwritten. The situation is comparable to a loss of power or switching the control off and back on while the application is running ("warm start").

A reset disables the [breakpoints](#) currently set in the project. If the "Warm" reset command is called just when the program sequence is stopped at a breakpoint, the user is asked if the current cycle should be completed before executing the reset or if the task should be stopped and the reset executed immediately. However, not all runtime systems are able to execute a reset without completing the current cycle first.

To start the application again, use [Start, page 133](#) after the reset.

Also refer to

- [Reset origin, page 136](#),
- [Reset cold, page 135](#).

3.5.14 Reset Cold

This command is available in online mode. It corresponds to the "Reset warm" command, but in addition to the "normal" variables, [retain variables, page 519](#), (!) are also reset to their initialization values. The situation is com-

Menu Items

parable to starting an application program that was just loaded to the control ("cold start").

A reset disables the **breakpoints** currently set in the project.

If the "Cold reset" command is called just when the program sequence is stopped at a breakpoint, the user is asked if the current cycle should be completed before executing the reset or if the task should be stopped and the reset executed immediately.

However, not all runtime systems are able to execute a reset without completing the current cycle first.

Also refer to

- [Reset warm, page 135,](#)
- [Reset origin, page 136.](#)

3.5.15 Reset Origin

This command is available in online mode. It resets the values of all inclusive **remanent variables, page 519,** back to their initialization values and **deletes the application program from the control.**

A reset disables the **breakpoints** currently set in the project.

If the "Origin" reset command is called just when the program sequence is stopped at a breakpoint, the user is asked if the current cycle should be completed before executing the reset or if the task should be stopped and the reset executed immediately. However, not all runtime systems are able to execute a reset without completing the current cycle first.

Also refer to

- [Reset warm, page 135,](#)
- [Reset cold, page 135.](#)

3.5.16 Breakpoints



Icon:

This command opens the **Breakpoints** dialog providing an overview of all breakpoints currently set in the project. The breakpoint parameters are displayed and can be modified. In addition, breakpoints can be added and removed or enabled and disabled.

The first breakpoint parameters shown are those that were set using the commands [Toggle breakpoint, page 140,](#) (here default values are set for the breakpoint "conditions") or [New breakpoint, page 138,](#) in the "Debug" menu (here the conditions can be defined directly).



If you have decided to use Multitasking, remember that by using the breakpoint you only stop the **one** task in which the breakpoint is located.

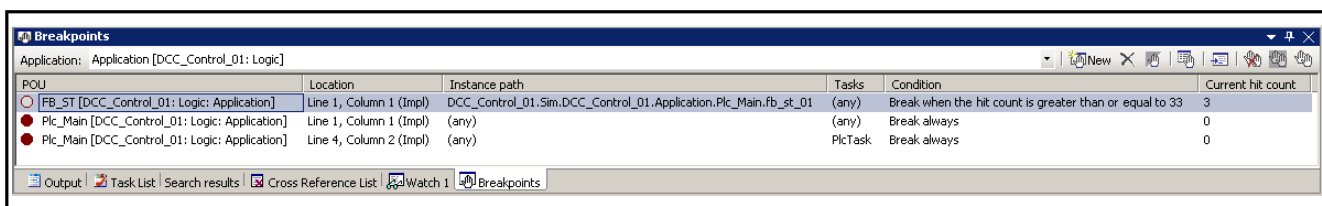


Fig.3-54: "Breakpoints" dialog

| | Description | Example |
|------------------------|--|---|
| Application | Name of the active application | Application [DCC_Control_01: Logic] |
| POU | Name of the function block that contains the breakpoint | Plc_Main |
| Position | Breakpoint position within the POU: line and column number (text editor) or network or element numbers (graphical editor); In the case of function blocks, "(Impl)" indicates that the breakpoint is located in the implementation of the function block, not in an instance. | Line 4, column 2 (Impl) |
| Instance path | Complete object path of the breakpoint position | DCC_Control_01.Sim.DCC_Control_01.Application.Plc_Main.fb_st_01 |
| Tasks | Names of the tasks in the execution of which the breakpoint becomes effective. If there is a limitation (default), "(all)" is listed here. | PlcTask |
| Condition | Indicates when the breakpoint becomes effective (dependency on number of hits); for possible values see New breakpoint, page 138 , (Default: 1). | Break if the number of hits is greater than or equal to 33 |
| Current Number of hits | Indicates how often the breakpoint has been ("hit") during the execution up to this point. | 3 |

Fig.3-55: "Breakpoints" dialog, description

The following functions are available by using the **buttons** in the upper right part of the dialog. They are used to edit the breakpoint parameters of the breakpoints selected in the list or to add new breakpoints or delete them:

| | | |
|--|----------------------------|--|
| | New breakpoint | Opens the New breakpoint dialog See also the description of the corresponding New breakpoint, page 138 , command. |
| | Delete breakpoint | Removes the breakpoint; Attention: do not confuse this function with "Disable"! |
| | Enable/disable breakpoint | Switches back and forth between the "enabled" and "disabled" status. In case of disabling, the breakpoint is not removed from the list and can be enabled again. |
| | Properties | Opens the "Breakpoint properties" dialog where the breakpoint parameters can be edited. The dialog is comparable to the "New breakpoint" dialog; see New breakpoint, page 138 , for a description. |
| | Go to source code position | Opens the "Select online status" dialog; from where you can move to the position of the breakpoint in the source code. |
| | Delete all breakpoints | Deletes all breakpoints in the application. The list is emptied. Attention: do not confuse this function with Disable! |
| | Enable all breakpoints | Enables all breakpoints that are currently disabled |
| | Disable all breakpoints | Disables all breakpoints that are currently enabled. The breakpoints remain in the list and can be disabled again. |

Fig.3-56: Symbols

Menu Items

3.5.17 Breakpoint Positions

Possible "breakpoint" positions, page 82, depend on the type of editor.

In principle, these include any position in a POU where the values of variables can change or where the program flow branches or another function block is called.



A breakpoint is always automatically set in all methods that can be called

that means that if a method managed by an interface is called, the breakpoints are also set in all of the methods of the function blocks that implement the interface and likewise in all of the function blocks derived that use the method.

If a method is called using a pointer to a function block, breakpoints are set in the method of the function block and in all of the function blocks derived that require the method.

Breakpoint symbols:

- Breakpoint in online mode:

```
1 | ldl();
2 | ● erg_0 :=fbinst
3 | TF hvarFALSE THEN
```

- Disabled breakpoint:

```
1 | ldl();
2 | ○ erg_0 :=fbinst
3 | TF hvarFALSE THEN
```

- Program stop at breakpoint:

```
1 | ldl();
2 | ● erg_0 :=fbinst
3 | TF hvarFALSE THEN
```

Also refer to

- [Breakpoint positions in the IL/LD/FBD editor, page 365](#)
- [Breakpoint positions in the CFC editor, page 317](#)
- [Breakpoint positions in the ST editor, page 388](#)

3.5.18 New Breakpoint

Icon:

Menu: **Debug** ► **New breakpoint...**

This command is used to define a new [breakpoint, page 82](#), in the project.

It is not relevant where the cursor is currently positioned, the 'New breakpoint' dialog opens where in the Position sub-dialog, one of the possible breakpoint positions in the entire project can be selected and in the "Condition" sub-dialog, certain conditions can be defined for the breakpoint becoming effective.



To set a breakpoint at the current position, also use the [Toggle breakpoint, page 140](#), command or the corresponding function in the [Breakpoint dialog, page 136](#).

Position

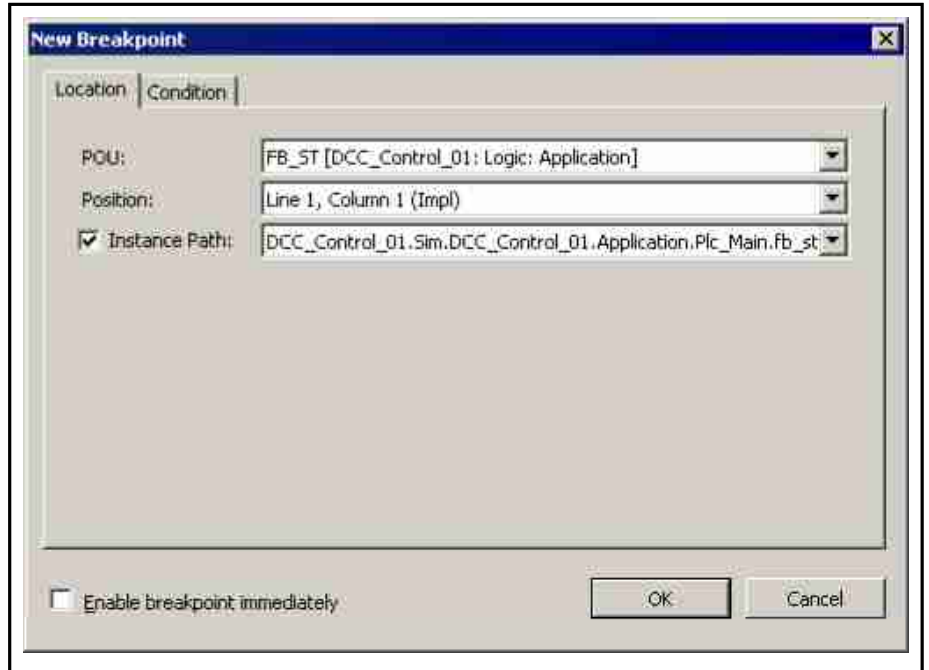


Fig.3-57: Dialog - "New breakpoint", position

- **POU:** The selection list provides all of the POU objects in the project. Select the POU object in which the breakpoint is to be set.
- **Position:** The selection list provides all of the possible breakpoint positions at the selected POU in the "POU" selection list; see [Breakpoint positions, page 138](#). Depending on the type of editor, these positions are listed as line and column numbers (text editor) or as network or element numbers.

For function blocks, "(Impl)" is also displayed.

The breakpoint can be set in the **implementation** or in an **instance**. If it is to be set in the implementation version, leave the "Instance path" option disabled. If it is to be set in an instance, enable the "Instance path" option to select the instance.

- **Instance path:** If the POU selected above is a function block and this option is disabled, the new breakpoint is set in the implementation version of the function block. If the breakpoint is to be set in an instance, enable this option and select the desired instance.

Menu Items

Condition

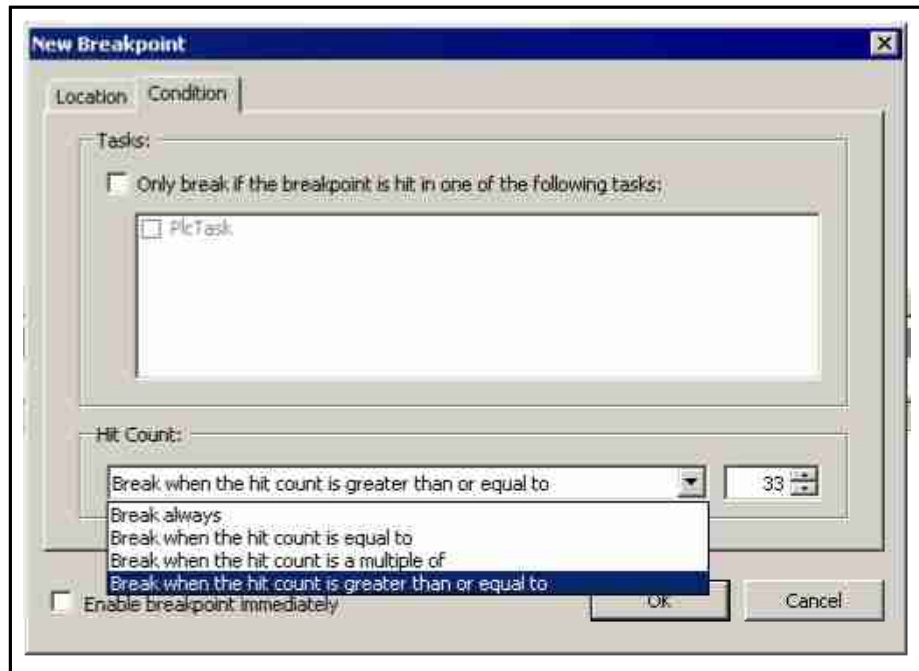


Fig.3-58: Dialog - "New Breakpoint", condition

- **Tasks:** If you enable the option "Only break when the breakpoint is hit in one of the following tasks" (by default: disabled), the breakpoint only becomes effective if the POU in which it is set is executed by one of the tasks selected here. All of the tasks defined in the project are listed and the desired tasks can be selected by placing a checkmark in the checkbox.
- *Number of hits:*
 - **Always break:** The program always stops at this breakpoint.

Alternatively, a number can be specified that indicates how often the breakpoint has to be hit before the program stops in accordance with the following conditions at execution.

 - **Break when the hit count is equal to**
 - **Break when the hit count is a multiple of**
 - **Break when the hit count is greater than or equal to** the "number".

3.5.19 Toggle Breakpoint

Default shortcut: <Alt>+<F9>

Menu: **Debug** ► **Toggle breakpoint**

In principle, this command is used to switch back and forth between active and inactive status for a [breakpoint](#), [page 82](#).

However, it also can set a new breakpoint if none has already been set at the current [breakpoint position](#), [page 138](#).

See [New breakpoint](#), [page 138](#).

If an active breakpoint is already present, it is disabled.

If an inactive breakpoint is already present, it is enabled.



Every active breakpoint becomes inactive when a user exits the online mode and then logs in again.

3.5.20 Call Stack

An active breakpoint is identified by a symbol; an inactive breakpoint is identified by a symbol.

Icon:

This command opens the window with the **call stack**. When a program is executed step by step, the current position is always displayed here with its complete call path.

Below the title line, the window always displays the name of the active **application** and the name of the **task** which is controlled by the POU that was just reached.

The call stack consists of a list of positions, each described by the **POU** name, the **position** and - in the case of instances - the **instance path**. Depending on the editor, the position is indicated by line and column number (text editor) or as network or element numbers (graphical editor).

The first line in the list, indicated by a yellow arrow, describes the current execution position. If this position is located in a function block that is called by another function block, the position of the call is described in the second line. If the caller is then called by another function block, this call position is described in the third line, etc.

The call stack is also available in offline mode or in normal online operation (without having to use debugging functions). In this case it contains the most recent position indicated during a step by step execution, though it is displayed in a "grayed out" font.

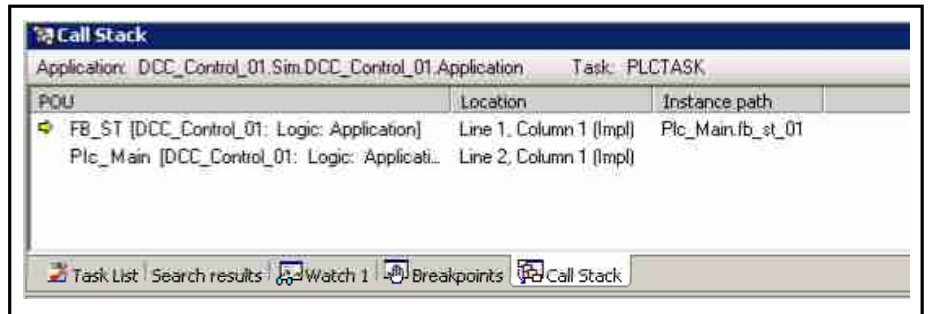


Fig.3-59: Call stack, current position of FB_ST, called by Plc_Main

3.5.21 Monitoring

This command opens a submenu with the five commands described in the following.

Watch 1, Watch 2, Watch 3 and Watch 4.

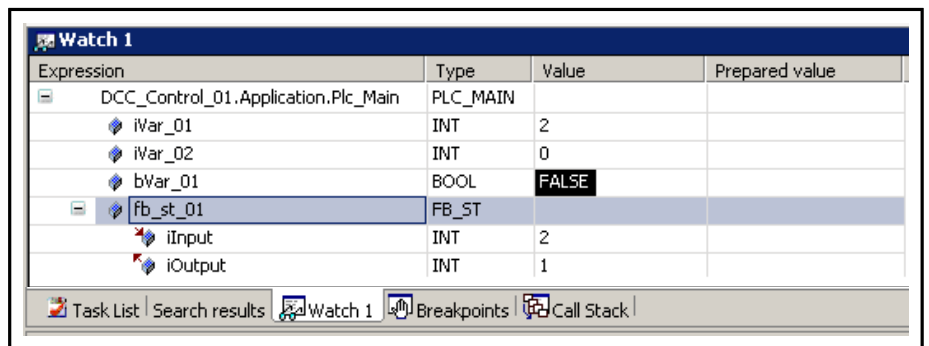


Fig.3-60: Example of a watch list in online mode

Menu Items

In this way, the respective monitoring list can be opened in a window and [edited, page 499](#).

In addition, **Watch all forces** can be used to get an overview of all of the permanently affected variables.

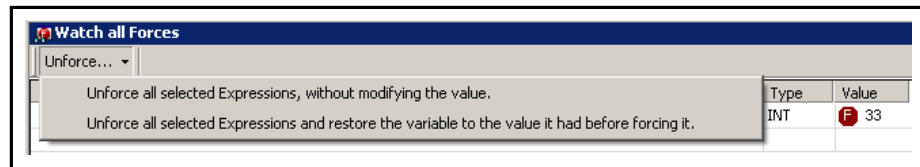


Fig.3-61: Example, "Watch all forces"

3.5.22 Procedure Step

Icon:

Default shortcut: <F12>

Menu: **Debug** ► **Procedure Step**.

This command can be used to execute a program in defined steps in online mode ([Stepping, page 82](#)), e.g. to make troubleshooting easier.

A single step is executed. For single step instructions, this is comparable to "stepping" with "single step"; see [Single step, page 142](#).

If a function block call is reached, "Procedure step" causes the function block called to be completely processed within the current step.

A complete action is processed in one sequence diagram.

If you only want to execute one step to the first instruction of a called function block, you have to use the [Single step command, page 142](#).

Also refer to

- [Execute to return, page 143](#).

3.5.23 Single Step

Icon:

Default shortcut <Alt>+<F12>

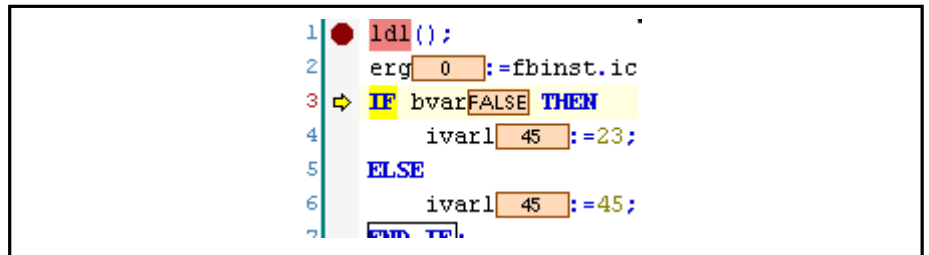
Menu: **Debug** ► **Single Step**.

This command can be used to execute a program in single steps in online mode ([Stepping, page 82](#)), e.g. to make troubleshooting easier.

A single step is executed. The program stops before the next instruction. If required, it is switched to another POU. If the current position is a function or a function block call, it is stopped before the first instruction of the function block called.

In all other situations the command has the same effect as a procedure step; see "[Procedure step](#)", [page 142](#).

Possible stop positions during single step processing depends on the editor type. The current position is displayed with yellow shadowing.




```

1 ldl();
2 erg 0 :=fbinst.ic;
3 IF bvarFALSE THEN
4   ivarl 45 :=23;
5 ELSE
6   ivarl 45 :=45;
7 END_IF

```

Fig.3-62: Example of a single step

3.5.24 Execute to Return

Icon: 

Default shortcut: <Shift>+<F10>

Menu: **Debug** ► **Execute to Return**.

This command can be used with a step by step execution of an application program (e.g. for debugging purposes).


If the program does not include any calls, the command 'Execute to return' causes a jump back to the start of the program. If, however, a jump was made previously into a called function block, it triggers a jump back to the calling instruction (note: This is new compared to IndraLogic 1.x).

This way, in the case of nested calls, 'Execute to return' takes effect step by step backwards in the hierarchy of the caller. This allows to jump back one step in order to jump into another called POU from there, for example.

Also refer to

- [Single step, page 142,](#)
- [Procedure step, page 142.](#)

3.5.25 Execute to Cursor

Icon: 

Default shortcut: <F7>

Menu: **Debug** ► **Execute to Cursor**.

This command can be used to execute the program up to a temporarily defined position (e.g. for debugging purposes).

The effect is the same as when the next breakpoint is defined at this position; see "New breakpoint", page 138.

The instructions located between the current and the newly defined position are executed.

3.5.26 Determine Next Instruction

Icon: 

Menu: **Debug** ► **Determine Next Instruction**.

This command can be used with a step by step execution of an application program (e.g. for debugging purposes) in order to define the instruction that is to be carried out next.

To do this, position the cursor in the instruction that is to be executed next and trigger the command.

3.5.27 Display Next Instruction

Icon: 

Menu Items

Menu: **Debug** ► **Display Next Instruction**.

This command can be used in online mode to move back to the current execution position if the cursor was placed somewhere else in the user interface during the step by step processing of a program. The window of the corresponding function block is brought back into the foreground and the cursor is placed in front of the next instruction to be executed.

3.5.28 Write Values for All Online Applications

Default shortcut: <Ctrl>+<F7>

Menu: **Debug** ► **Write values for all online applications**.**⚠ CAUTION**

The extraordinary modification of values in an application running on the control might lead to unwanted behavior of the controlled system.

Depending on the controlled system, damage at the system and workpieces might result or the safety of persons might be endangered.

Evaluate possible risks before writing or forcing variable values and take corresponding precautions.

"Writing" a value with this command means that at the beginning of the next cycle, the corresponding variable in the control is set to the value defined for it in the programming system.



For more information, see also the "[Force values](#)" command used to set the variable values permanently in the control.

To prepare variables for writing, the desired value has to be defined in online mode at one of the following positions used for [monitoring, page 82](#),:

- In a monitoring window that was defined in the project and that contains a list of the variables to be monitored ([monitoring list, watch list, page 499](#)).
- In the online view of an object in the [declarations, page 330](#), of the associated editor

Example:

Write values

Open an object in online mode, e.g. a program written in ST.

The expressions (variables) that can be monitored are displayed in a table in the declarations.

Click on the corresponding field in the **Prepared value** column and enter the desired value.

Then execute the **Write values** command that is in the **Online** menu by default.

The prepared value is then entered in the same line in the **Value** column, which indicates that it was written to the control.

The "Prepared value" field is now empty again.

The same action can be performed in a [monitoring list, watch list, page 499](#), which contains the corresponding expression or variable.

In this context, also note the [Prepare value, page 145](#) dialog, which can be opened for the variables that were just "forced" and in which a new value can also be defined for "writing".

3.5.29 Force Values on All Online Applications

Menu: **Debug** ▶ **Force values on all online applications.**

CAUTION

The extraordinary modification of variable values in an application running on the control might lead to unwanted behavior of the controlled system.

Depending on the controlled system, damage at the system and workpieces might result or the health and life of persons might be at risk.

Evaluate possible risks before writing or forcing variable values and take corresponding precautions.

This command is available in online mode. It causes one or more expressions (variables) in the application on the control to be permanently set to defined values.

This setting is then carried out once at the beginning and once at the end of a processing cycle.

Sequence of the process in a cycle:

1. Read inputs
2. Force values
3. Process code
4. Force values
5. Write outputs



See also the [Write values, page 144](#) command used to make the setting with a defined value only once at the beginning of the cycle.

The forcing is performed until the user cancels it explicitly for a [single](#) variable or for [all](#) variables or until the user logs out of the application.

To prepare variables for forcing, the desired value has to be defined in online mode at one of the following positions used for [monitoring, page 82](#),:

- In a monitoring window that was defined in the project and that contains a list of the variables to be monitored ([monitoring list, watch list, page 499](#)).
- In the online view of an object in the [declarations, page 330](#), of the associated editor
- In the online view of an object within the implementation section of the [FBD/LD/IL editor, page 362](#).

A "forced" value is indicated by an  symbol.

Menu Items

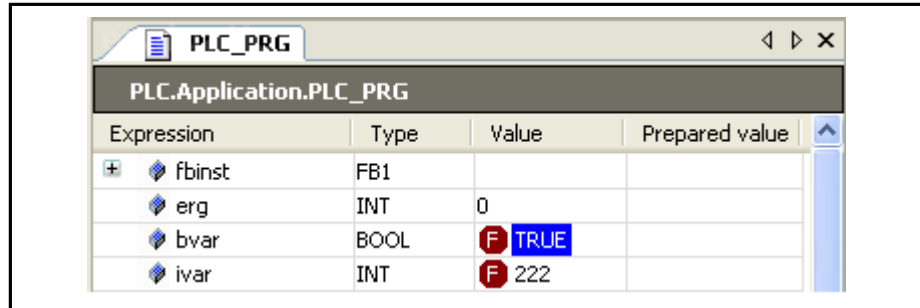


Fig.3-63: Example of forced variables in the declaration editor of a function block (online view)

"Prepare value" dialog

This dialog is used to define a new value for a variable. To do this, either a new value is entered or forcing is canceled and the value returns to the current or previous value.

The dialog opens when clicking the "Prepared value" field of a variable that has just been forced

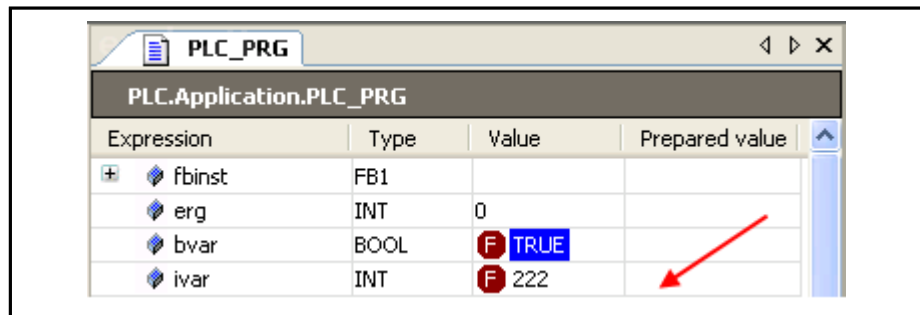


Fig.3-64: Example of forced variables in the declaration editor, click "Prepared value"

or in the Inline monitoring field of the variable in the implementation section of the FBD/LD/IL/ST editor.

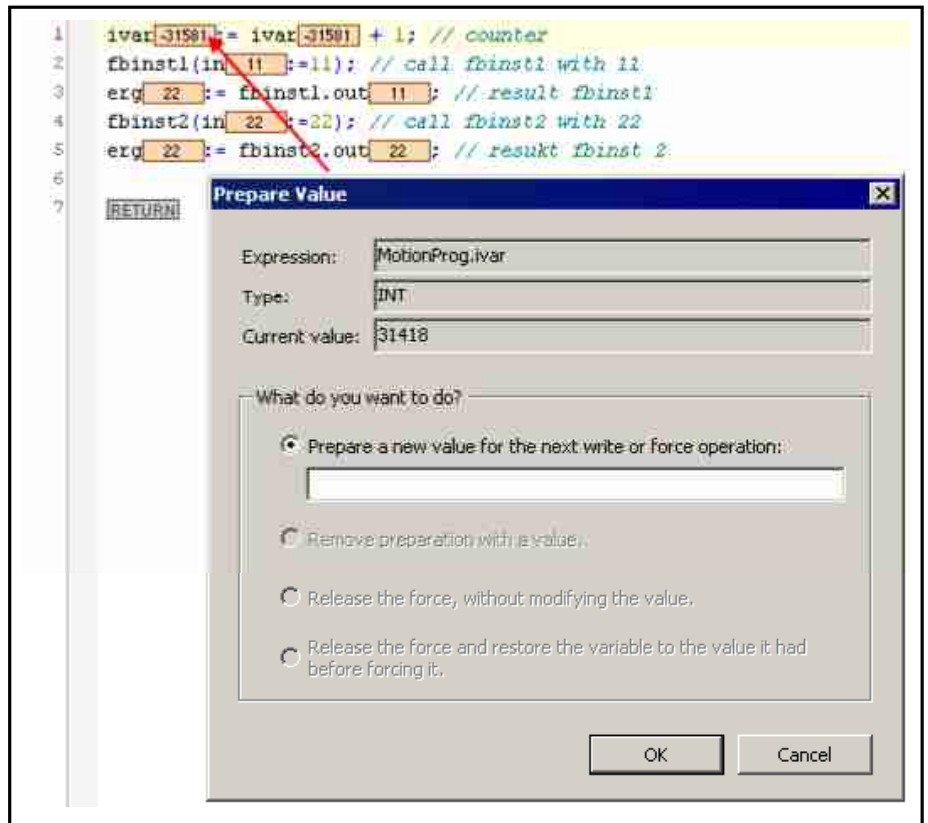


Fig.3-65: "Prepare value" dialog

The following information on the respective variables is displayed:

- **Expression:** Path: e.g. "PLC.Application.PLC_PRG.ivar"
- **Data type:** e.g. "DWORD"
- **Current value:** e.g. "TRUE" or "23"

Select one of the following options to determine **what should happen with the variable**:

- **Prepare a new value for the next write or force operation:** Depending on the data type of the variable, a new numerical value or string value can be entered to which the variable is to be set.
- **Remove preparation with a value:** The prepared value is deleted.
- **Release the force without modifying the value:** The variable is marked with <Unforce> and thus prepared to receive the current value from the control.
- **Release the force and restore the variable to the value it had before forcing it:** The variable is marked with <Unforce and restore> and thus now prepared to receive the value that it had before forcing.

After closing the dialog with "OK", the variable in the "Prepared value" field in the monitoring view(s) now indicates a newly defined value or "<Unforce>" or "<Unforce and restore>". Then next time the "Force values" or "Write values" command is carried out (for the first option), all of these prepared values are set.

3.5.30 Cancel Forcing for All Values on All Online Applications

Menu: **Debug** ► **Cancel forcing for all values on all online applications.**



Menu Items

CAUTION

The extraordinary modification of variable values in an application running on the control might lead to unwanted behavior of the controlled system.

Depending on the controlled system, damage at the system and workpieces might result or the health and life of persons might be at risk.

Evaluate possible risks before writing or forcing variable values and take corresponding precautions.

This command can be used in online mode to cancel forcing for all of the variables currently being forced for the applications included in the project.

The variables then receive their current value(s) from the control.

This corresponds to the option "Release the force without modifying the value." that can be selected in the [Prepare values](#) dialog for a forced value.

3.5.31 Add All Forces to Monitoring Window

This command is available in online mode as soon as one of the four [monitoring views](#), [page 141](#), "Watch 1", "Watch 2", "Watch 3" or "Watch 4" is active.

It adds all of the variables in the current application that are now prepared to be forced or have already been forced to the watch list.

Please note, however, that this only works for docked monitoring views.

Also note the "Watch all forces" view, which automatically and continuously displays all variables that have been prepared for forcing and also provides commands for canceling the forcing.

3.5.32 Sequence Control

Menu: **Debug** ▶ **Sequence control**.

This command is used to switch the sequence control on and off which is supported for the ST, FBD, LD and IL language editors.

An activated sequence control allows to track the processing of the application program.

1. The current values of the variables and thus also the results of function calls and operations are displayed in the editor windows, at the relevant processing position and at the relevant processing time. For comparison: The standard monitoring only provides the value of a variable between two processing cycles.
2. Exactly the code lines or networks are marked in color that have been run through in the current cycle.

The sequence control can only be activated in online mode and only works in the currently visible part of the currently active editor window. 'Sequence control activated' is displayed in the status line as long as the function is active and sequence control positions (parts of the code run through) are visible in the editor window.



An activated sequence control extends the application runtime!
If the "Secure online operation" option is enabled in the "Communication settings" of the control, a message box appears upon sequence control switch-on. You are asked whether you really want to activate the function and are once again provided the possibility to cancel the process.

Illustration of the sequence control in the different language editors:

By default, green is configured as color for marking the sequence control positions. (The color can be changed in the [Text editor options](#), page 218.)

In all editors, the current values of variables and inputs and outputs concerned are displayed in small rectangular fields as they are also used for the standard monitoring. In code parts run through, these fields are displayed in the color defined for the sequence control, in code parts not run through, they are shown in white with grey font.

Please note that the displayed value at a code position not run through is a "normal" monitoring value, i.e. the value between two task cycles.

```

1  i 1619 := i 1619 + 1;
2  b 0 := NOT b 0;
3  IF str 'abcdefghij' = str1 " " THEN
4  f12 1.5 := f1 1.23;
5  ELSE
6  f12 1.5 := 1.5;
7  D 6.5E+04 := B 255 * B 255;
8  END_IF;
9  IF D 6.5E+04 < 0.0 THEN

```

Fig.3-66: Example: Sequence control in the ST editor

In **network editors**, the networks run through are marked at the left edge by means of bars in the 'sequence control color'.

In the **LD**, the connection lines currently run through are generally shown in **green** (or the 'sequence control color' currently set), the other ones in **gray**.

The current value of the connection is shown, as well:

- TRUE by means of **thick blue**,
- FALSE by means of **thick black** lines,
- unknown or analog values by means of **thin black** lines.

Due to the combination of the relevant information, this might lead to dashed lines.

Menu Items

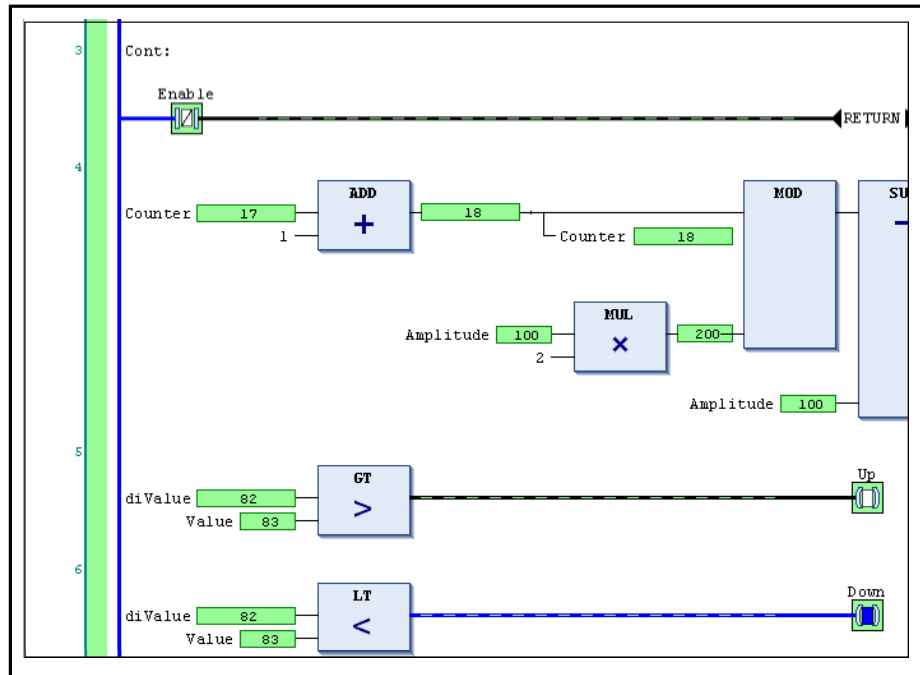


Fig.3-67: Example: Sequence control in the LD editor

In IL, two fields are used for each instruction to display the current values.

The field on the left side of the operator contains the current accumulator value, the field on the right side of the operand the operand value.

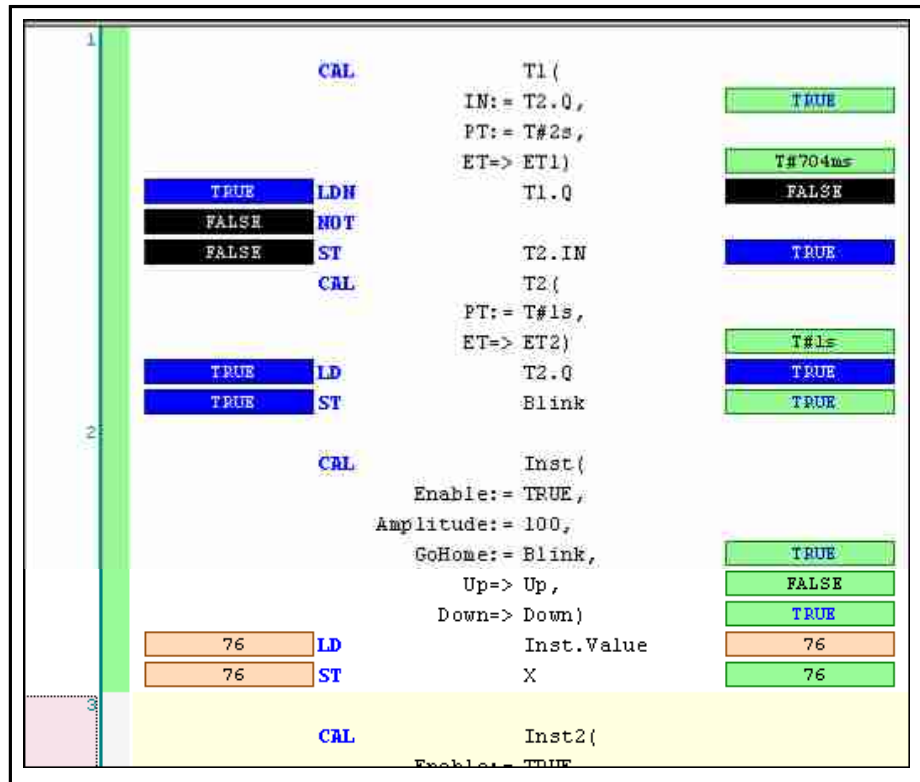


Fig.3-68: Example: Sequence control in the IL editor



Note that **writing values** is also possible in the sequence control mode.

Forcing is not possible!

Double-clicking the value display field opens the [Prepare value, page 145](#), dialog to enter the desired value.

3.5.33 Display Mode

This command can be used to set the format for displaying the values ([Monitoring, page 82](#)) in online mode.

To do this, click on the desired option, which is then displayed with a checkmark.

- Binary
- Decimal
- Hexadecimal

3.6 Tools

3.6.1 Comparison of PLC Objects

Comparison of PLC Objects, General

The Compare... menu item provides the option

1. to compare PLC objects of the **current** project with each other,
2. to compare PLC objects of the **current** project with those of an export file (*.iwx),
3. to compare PLC objects of two export files with each other.

In the first case, the desired objects are to be highlighted in the Project Explorer.

In the second case, the desired object is to be highlighted in the Project Explorer and "Compare..." is to be selected in the context menu of the object. In the appearing window, the desired export file has to be selected.

To compare two export files, the menu item can be accessed in **Tools ▶ Compare...**

The description is made based on the combination of project/export file. This applies similarly to export file/export file.

If the comparison cannot be made, since the object types differ, an error message appears.

Basics of the comparison in IndraWorks Engineering:

- Rexroth IndraWorks 12VRS Engineering, DOK-IWORKS-ENGINEE*V12-APxx-EN-P, R911334388.
- – [Toolbar in the "Compare" dialog, page 110.](#)
- – [Toolbar in the "Comparison results" dialog, page 111.](#)

Comparison of PLC objects

- [General module, page 152](#)
- [Logic, page 153](#)
- [Application, page 154](#)
- [Task configuration, page 157](#)
- [Task, page 152](#)

Menu Items

- Folder (Difference from IndraWorks folder), page 159
- Data types, page 159
- Global variable list, persistent variables, network variables, page 161
- Network variable lists, receiver end, page 162
- Programs, function blocks, functions..., page 163
- Interfaces, interface methods, interface properties, page 169
- Library manager, page 170
- Library info, page 172
- Data server, page 173,
- Data source (data server), page 174,
- Symbol configuration, page 175,
- Visualization manager, page 175,
- Visualization, page 177
- Text list, page 177
- Image pool, page 177
- Recipe manager, page 177
- Recipe definition, page 179
- Devices (Onboard I/O, Inline I/O, field busses, ...), page 179

Comparison: General Module

Here only the name is displayed in the window. "Merging" is not possible, since the element is fixed.

Differences can result based on the name or language.

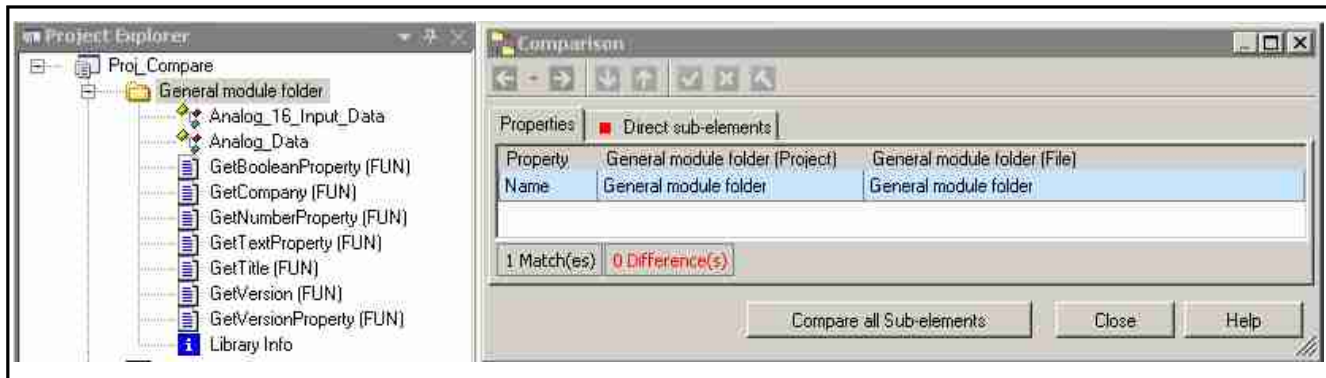


Fig.3-69: Comparison: General module folder

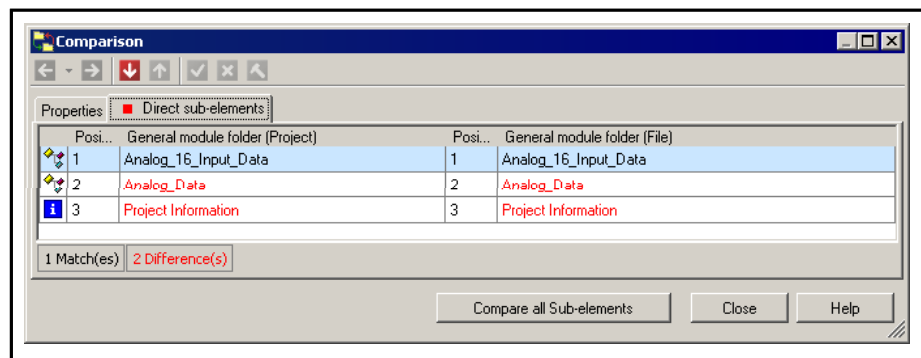


Fig.3-70: Comparison: Listing of the "Direct subelements", two elements with differences (red)

Menu Items

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer to merge.

Select the cross to prevent (undo) merging.



The terms "Project Information" and "Library Info" designate the same element.

Comparison: All subelements

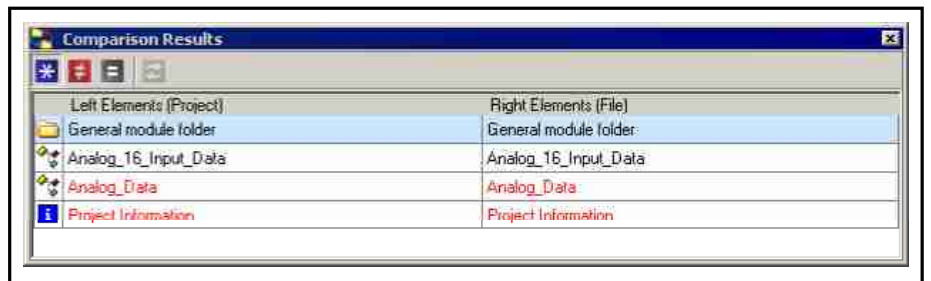


Fig.3-71: Comparison: "All subelements", two elements with differences (red with filter option "All elements", "Different elements", "Identical elements")

Comparison: Logic

Here only the name is displayed in the window. "Merging" is not possible, since the element is fixed.

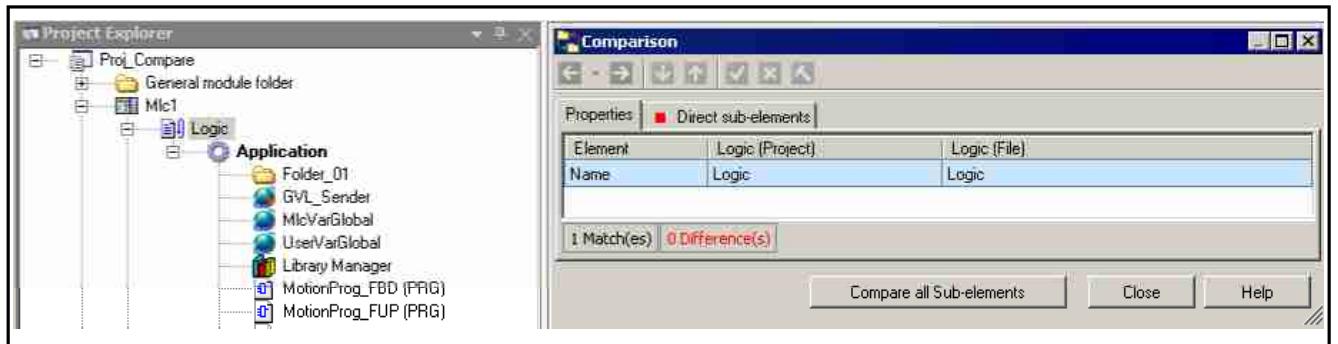


Fig.3-72: Comparison: Logic

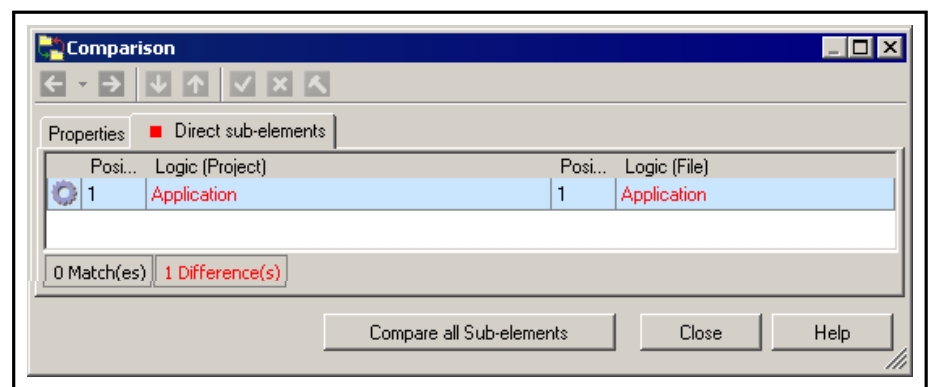


Fig.3-73: Comparison: Logic with one single subelement "Application"

Menu Items

Comparison: All subelements

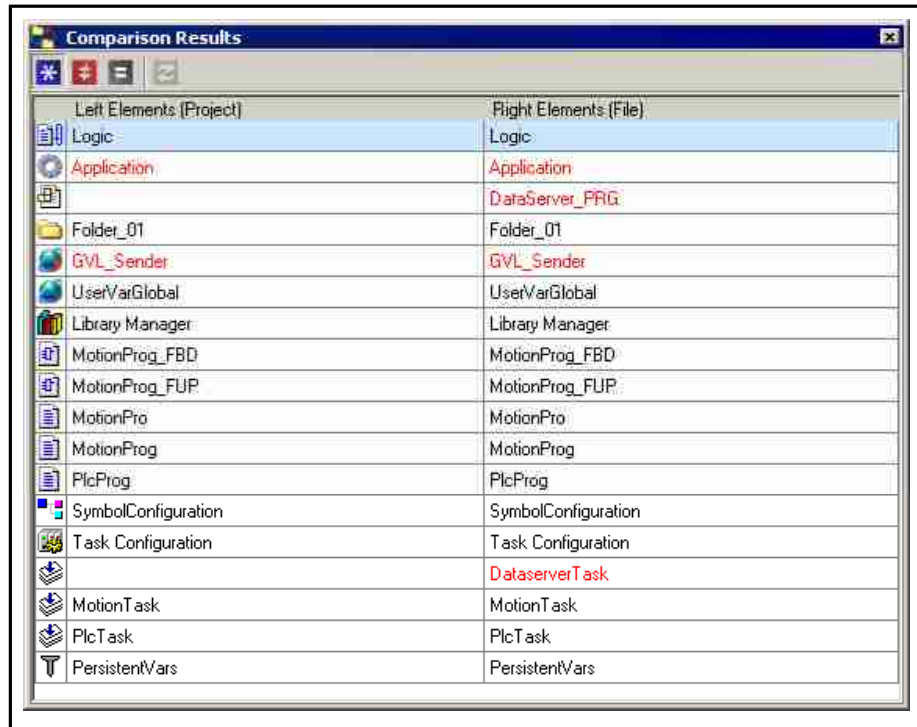


Fig.3-74: Comparison: Logic with all subelements with filter option "All elements", "Different elements", "Identical elements"

Comparison: Application

Here, the name is displayed, the options for the boot project handling and the other application properties. These are located under Properties in the "Application" node context menu.

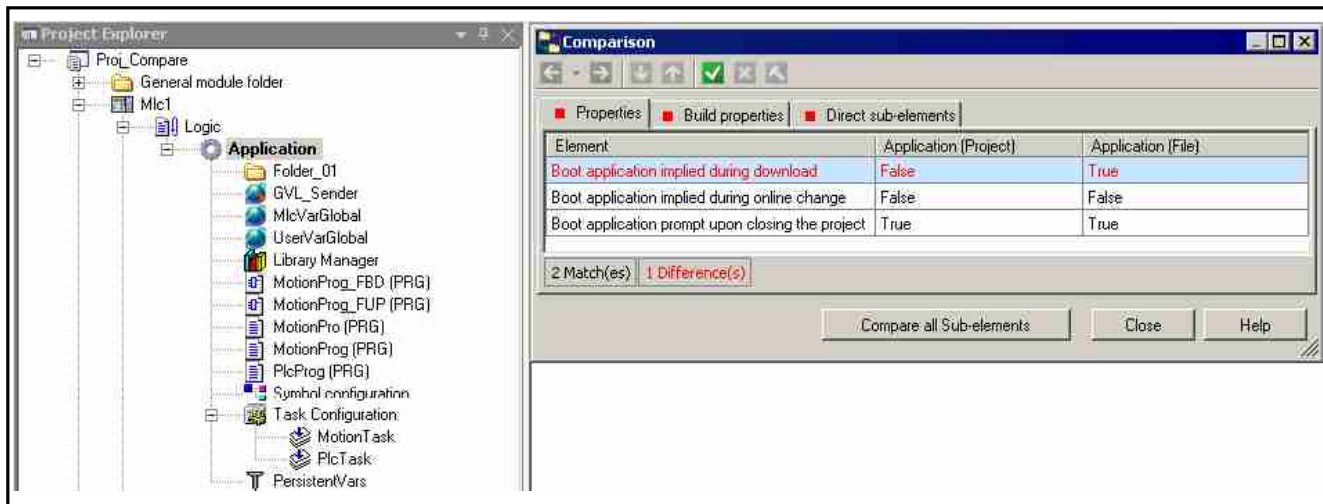



Fig.3-75: Comparison: Application

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer  to merge.

Select the cross  to prevent (undo) merging.

The next tab shows the build properties.

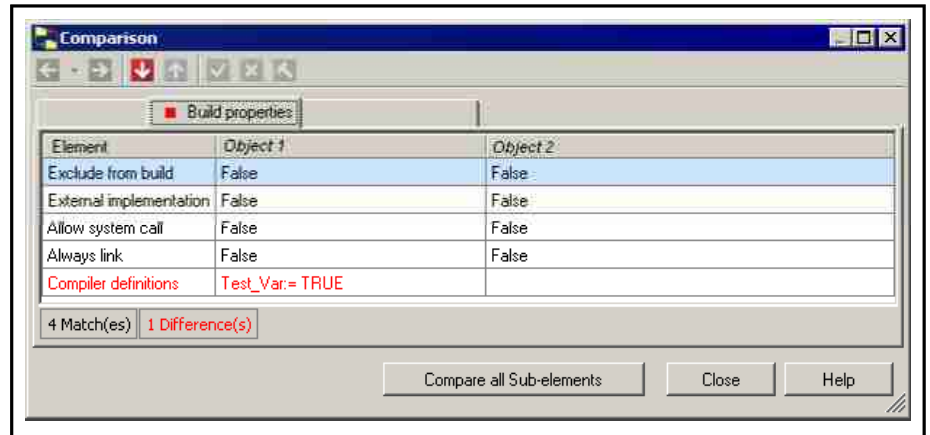






Fig.3-76: Comparison: Build properties object 1/object 2, one difference identified and marked in red

If the assignment in the columns of the two objects differs,  can be used to select the next difference ( for the previous difference).

If the checkmark  appears, it can be used to prepare the acceptance (merging):

Select the hammer  to merge.

Select the cross  to prevent (undo) merging.

The "DataServer_PRG" program (right position 7) is missing in the project and is to be taken from the file.

"GVL_Sender" differs with regard to the content and the order of the objects is different from position 7.

Menu Items

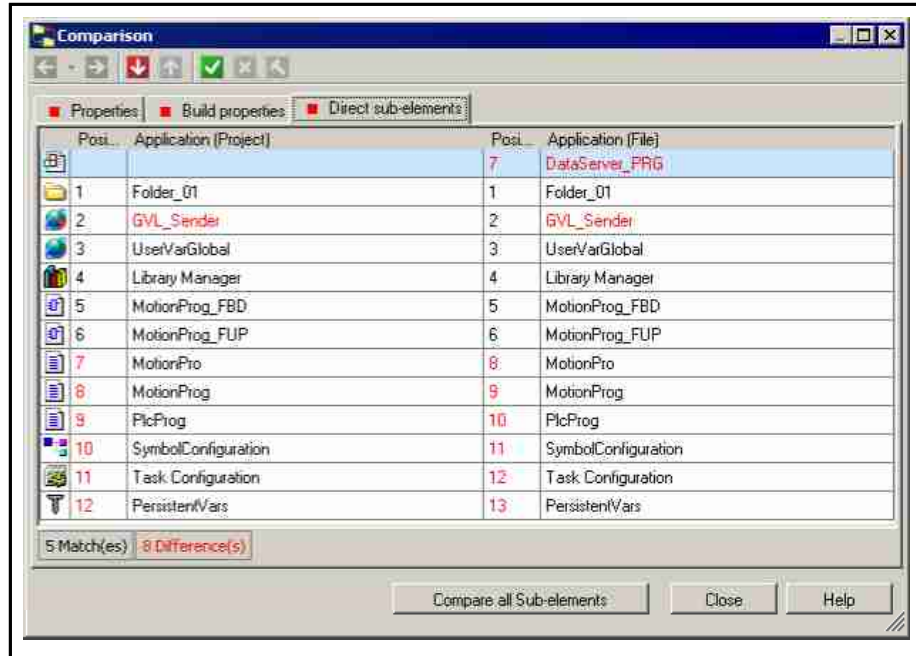


Fig.3-77: Comparison: "Direct subelements"

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer to merge.

Select the cross to prevent (undo) merging.

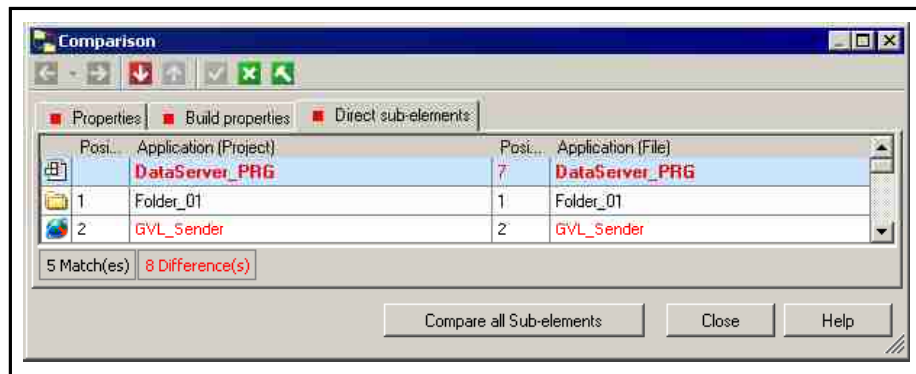


Fig.3-78: Comparison: File is accepted into the project with "Start merge operation"

Comparison: All subelements



Fig.3-79: Comparison: Application with all subelements with filter option "All elements", "Different elements", "Identical elements"

In addition to the direct subelements, their subelements are displayed, as well. Here, the "sub-subelements" of the task configuration (DataServerTask, only right, MotionTask and PlcTask).

Comparison: Task Configuration

Here only the name Task configuration is displayed in the window. "Merging" is not possible, since the element is fixed.

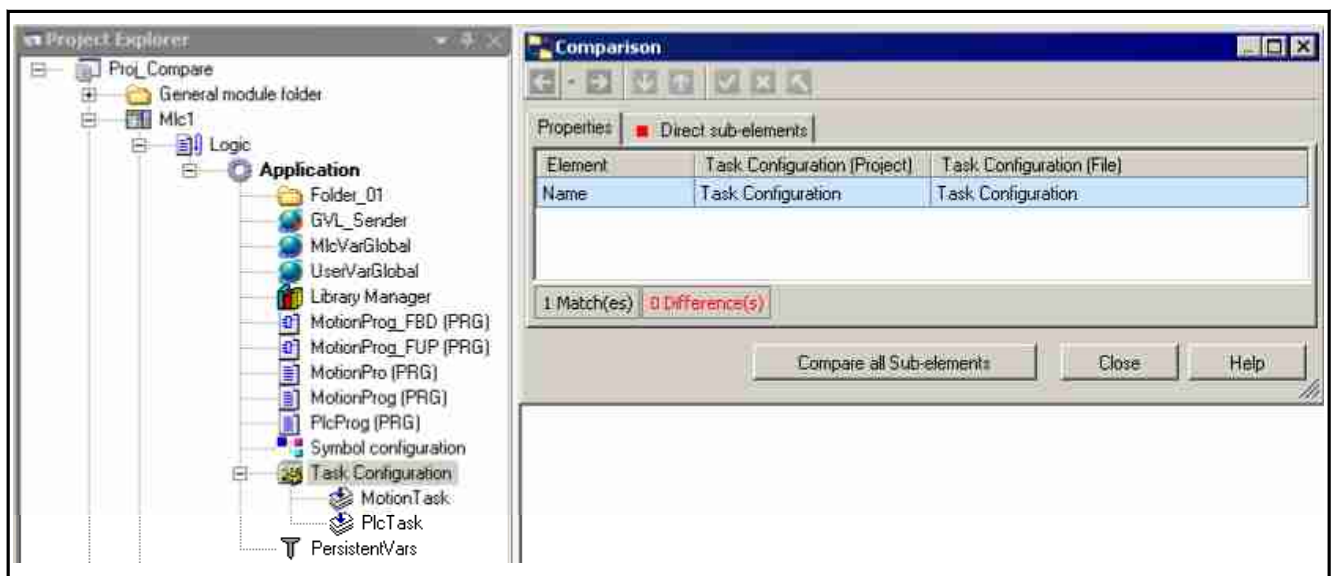


Fig.3-80: Comparison: Task configuration

Menu Items

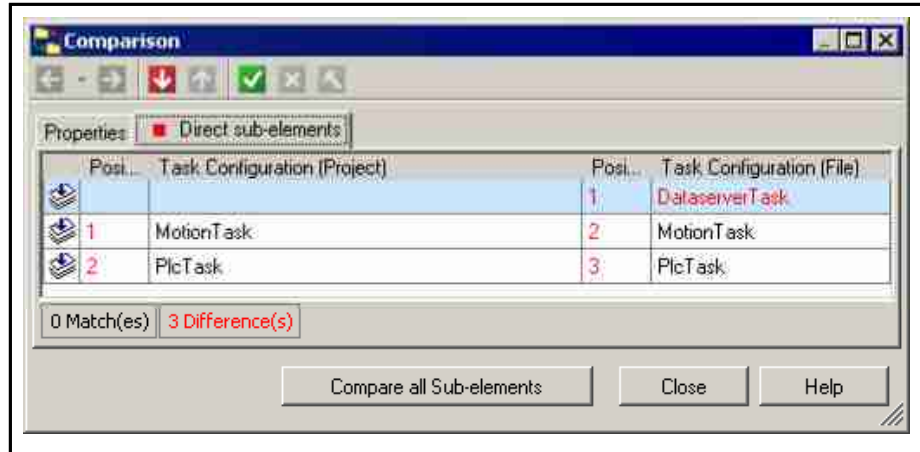


Fig.3-81: Comparison: Task configuration, tasks as "Direct subelements"

The "DataserverTask" task is only contained in the file storage and can be accepted into the project.

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer to merge.

Select the cross to prevent (undo) merging.

Comparison: All subelements

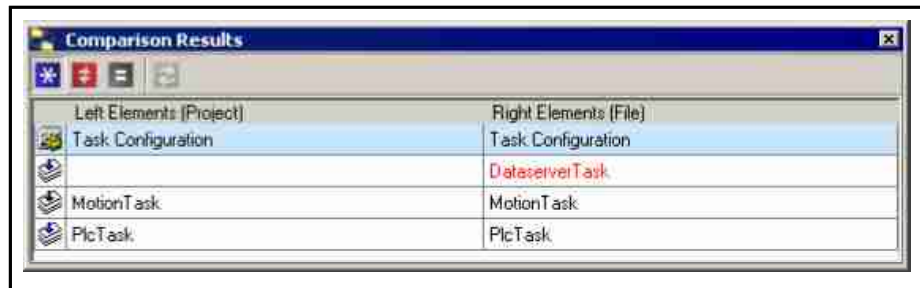


Fig.3-82: Comparison: Application with all subelements with filter option "All elements", "Different elements", "Identical elements"

Comparison: Task

Task objects are compared in tabular form. Here, comparison between the "PlcTask" and "MotionTask" tasks.



Merging is not recommended...

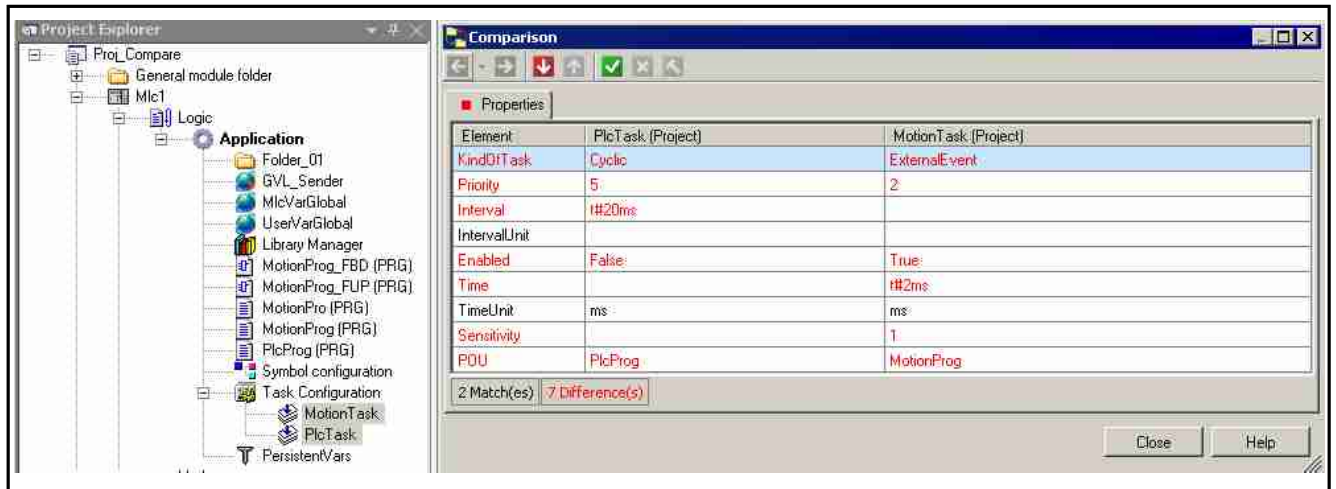


Fig.3-83: Comparison: Task, here comparison between the "PlcTask" and "MotionTask" tasks

Comparison: Folder (Differences with Respect to the IndraWorks Folder)

Only the name is displayed. It can be renamed if the folder in the file (right) has another name.

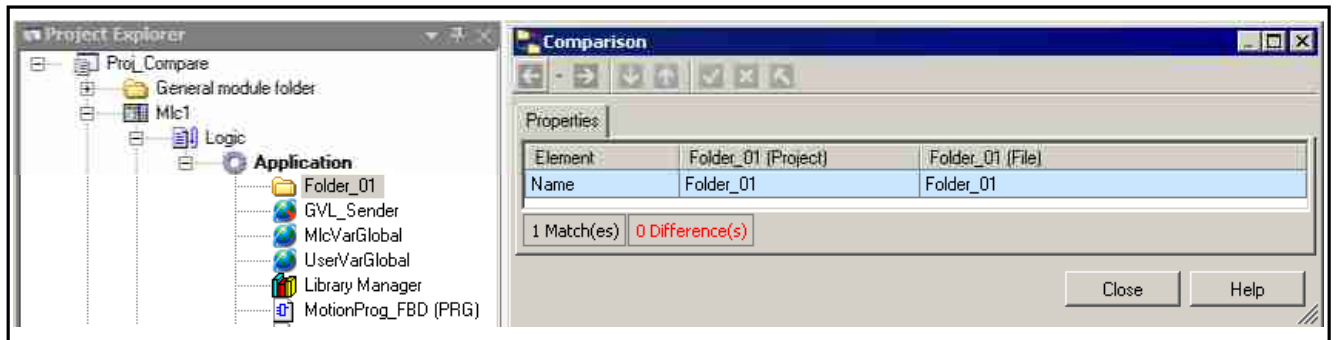


Fig.3-84: Comparison: Folder

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer to merge.

Select the cross to prevent (undo) merging.

Comparison: Data Types

The comparison applies to all data types that can be created with the DUT editor as independent objects.

Among others, the following are supported: structures (STRUCT), arrays (ARRAY), unions (UNION), enumeration data types (ENUM) and partial range data types (SUBRANGE).

When data types are compared, the content is displayed as text and can be modified.

Menu Items

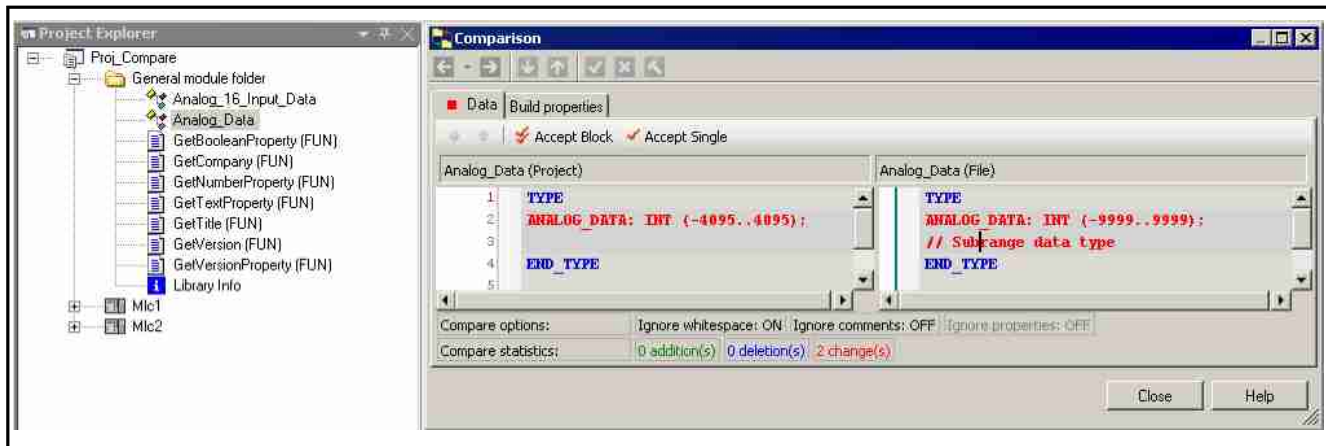


Fig.3-85: Comparison: Data Types

The next/previous difference is selected with or .

Accept Block can be used to accept the complete highlighted text block.

Accept Single can be used to accept text line by line. In this case, only the possible result of the merge is displayed first. Clicking the button again resets the display.

Select the hammer to merge.

Select the cross to prevent (undo) merging.

The next tab shows the build properties.

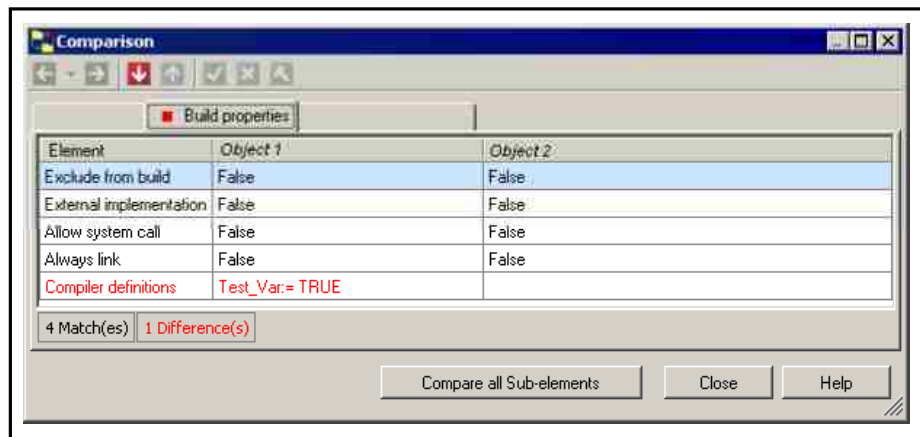


Fig.3-86: Comparison: Build properties object 1/object 2, one difference identified and marked in red

If the assignment in the columns of the two objects differs, can be used to select the next difference (for the previous difference).

If the checkmark appears, it can be used to prepare the acceptance (merging):

Select the hammer to merge.

Select the cross to prevent (undo) merging.

Comparison: Global Variable List, Persistent Variables, Network Variables

The comparison applies to all global variable lists ①, global variable lists used as network variables on the sender end ② and persistent variable lists ③ that can be added as independent objects of an application (or in the general module folder).

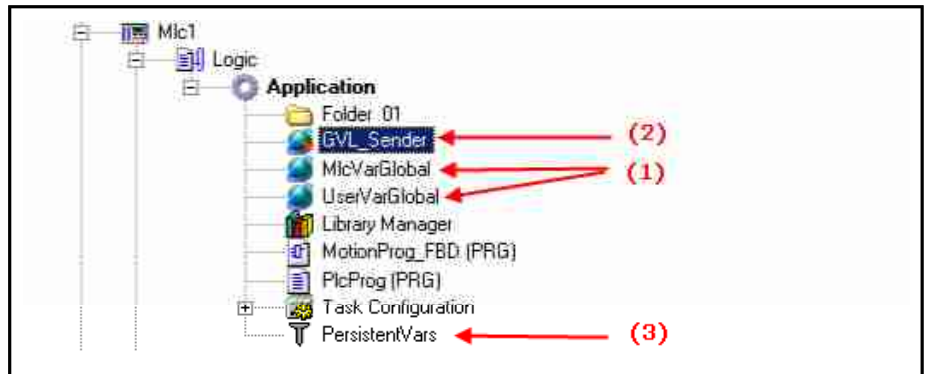


Fig.3-87: Comparison: Possible global variable lists

When data types are compared, the content is displayed as text and can be modified.

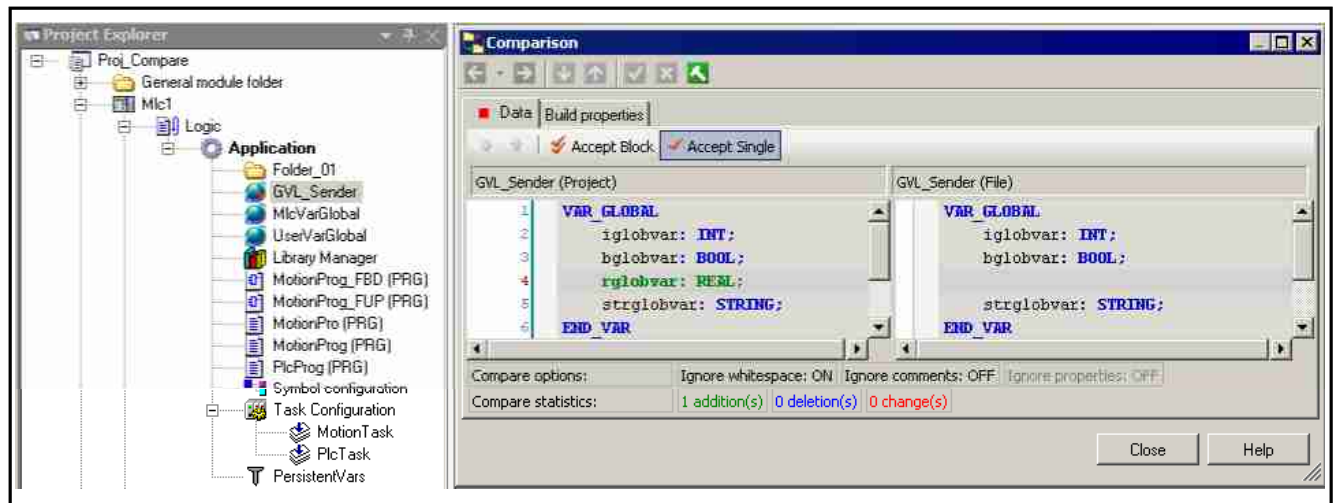


Fig.3-88: Comparison: Global variable lists

The next/previous difference is selected with or .

Accept Block can be used to accept the complete highlighted text block.

Accept Single can be used to accept text line by line. In this case, only the possible result of the merge is displayed first. Clicking the button again resets the display.

Select the hammer to merge.

Select the cross to prevent (undo) merging.

The next tab shows the build properties.