Notice 1

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any error that may occur in this document. Furthermore, pictures shown are examples only.

Notice 2

This document complies with the program revision 1.3.2.

Notice 3

Additional information may be found in Release Notes.

Trademarks

All Microsoft products referenced in this document are either trademarks or registered trademarks of Microsoft Corporation.

Other brand or product names are trademarks or registered trademarks of their respective holders.
1 Introduction

2 General handling in the Relay Configuration Tool

3 Programming with the Relay Configuration Tool

4 The project tree editor

5 Graphic editor

6 Text editors

7 The physical part of the project

8 Compiling
Contents:

1. Introduction ...............................................................................7
   1.1. What is the Relay Configuration Tool? ..............................7
   1.2. Using this manual .............................................................7

2. General handling in the Relay Configuration Tool ...............9
   2.1. Starting and exiting the Relay Configuration Tool ...........9
   2.2. The help function in the Relay Configuration Tool ............9
   2.3. General operational items in the Relay Configuration Tool ....10
   2.4. The toolbar in the Relay Configuration Tool .....................13

3. Programming with the Relay Configuration Tool ...............15
   3.1. General characteristics of the IEC 1131-3 standard ..........15
   3.2. General project development with the Relay Configuration Tool 15

4. The project tree editor ............................................................17
   4.1. The project tree in the Relay Configuration Tool .............17
   4.2. Inserting items in the project tree .....................................18
   4.3. Deleting items in the project tree .....................................21

5. Graphic editor .............................................................................23
   5.1. The graphic editor - easy programming in SFC and FBD ....23
   5.2. Opening the graphic editor ...............................................23
   5.3. Editing SFC ......................................................................24
   5.4. Editing FBD ......................................................................27

6. Text editors ..............................................................................33
   6.1. The text editors - easy programming in IL .......................33
   6.2. Opening the text editor .....................................................33
   6.3. Dialog in the IL editor .......................................................34
   6.4. Variable declarations in the Relay Configuration Tool ........34

7. The physical part of the project .............................................37
   7.1. The physical part ...............................................................37
   7.2. Resources .........................................................................38
   7.3. Tasks .................................................................................38

8. Compiling .................................................................................39
   8.1. Taking programs into operation .......................................39
   8.2. Compiling .........................................................................39
   8.3. Compilation errors in the user error list .........................39
1. Introduction

1.1. What is the Relay Configuration Tool?

The Relay Configuration Tool (Recap) is an IEC 1131-3 compliant programming system for Programmable Logic Controllers running under Microsoft® Windows®. Relay Configuration Tool offers features for project management, programming, configuration, documentation and commissioning. The broad acceptance of Microsoft Windows ensures familiar user interface facilities known to most of today's engineers dealing with PLC projects.

IEC 1131-3 defines the programming languages of PLC programs and the set up of the physical target systems. In the Relay Configuration Tool, the project tree represents the complete structure of the current managed project. This concerns mainly the project library including the program organization units programmed by the user, further external included standard libraries with predefined sources, user-defined data types and the target structure.

For the definition of program organization units, the Relay Configuration Tool provides two main language editors. The graphic editor allows the definition of POUs in the languages SFC and FBD. These languages can also be mixed. The edit functions are easily accessible via toolbar. In addition, the Relay Configuration Tool editor allows dialog lead input of programs in IL and the variable declarations. The text editor handling is equivalent to any normal ASCII editor.

1.2. Using this manual

This manual offers a general overview of some important functions when starting to work with the Relay Configuration Tool. It is possible to have a quick introduction of the main philosophy and how to work with the tool in general. Detailed information of specific single items can be called via the context-sensitive help function of the Relay Configuration Tool.

It is recommended to first of all completely read this Quick Start Reference without working directly with the tool to get an overview of the possibilities of the Relay Configuration Tool system.

---

1. Microsoft is a registered trademark of Microsoft Corporation.
2. Windows is a registered trademark of Microsoft Corporation.
1. Introduction

Quick Start Reference
2. General handling in the Relay Configuration Tool

2.1. Starting and exiting the Relay Configuration Tool

During the start-up process the dialog 'Relay Configuration Tool log-in' appears on the screen. Enter your password and confirm the dialog and the project will be opened.

The IEC projects are password protected within the premises of ABB. The password for the IEC projects is ‘ABB’ granting the access to freely construct also the hardware tree. For example, when you launch the Relay Configuration Tool and the 'Relay Configuration Tool Log-In' dialog appears, type in ‘ABB’ and press the 'OK' button. At this stage the IEC project that is attached to the object, will be opened.

**Note!** You are free to change the passwords, if you prefer.

To exit the Relay Configuration Tool, double click on the top left corner of the Relay Configuration Tool's main window, or call the menu item 'Exit' of the submenu 'File'. You can also press the key combination <Alt><F4>.

2.2. The help function in the Relay Configuration Tool

The Relay Configuration Tool includes a context-sensitive help function on single subjects. The help function is accessible via the submenu 'Help' or the key <F1>. The help function provides complete reference information and possibilities of the selected item. The help function should be consulted in case you need exact information.

Help topics are provided for a complete window or a marked object. To display the help for a specific window (e.g. project tree window), the appropriate window must currently be active and the help function can be selected. To display the help of a single item in a window (e.g. a worksheet in the project tree), mark the appropriate object and select the help function.
2. General handling in the Relay Configuration Tool

Quick Start Reference

In addition, each dialog provides a help button, so that further information of the current dialog can be displayed, i.e. setting possibilities of the current dialog, etc.

2.3. General operational items in the Relay Configuration Tool

The Relay Configuration Tool supports mouse and keyboard operations. The complete functions (apart from where textual based input is required) can be accessed via mouse or via keyboard (and menu).

Some of the following technical terms may be unknown at the moment. The purpose is only to explain the general operation features with mouse and keyboard. For detailed information on the technical terms, please see Chapter 4, The project tree editor.

The main philosophy for the operation is the distinction of editing an object and changing the properties of an object. For example, to open a window containing a source definition of a program organization unit, click the item in the project tree with the left mouse button.
Fig. 2.3.-1 Example of source definition opened by double-clicking the item in the project tree with the left mouse button.

The properties (functional settings) of the objects are set via dialogs. To display the Properties dialog, double-click the appropriate object with the right mouse button. In this dialog you can, for example, change the name of an Action block or the processor type of a resource.
2. General handling in the Relay Configuration Quick Start Reference

Fig. 2.3.-2 Example of resource properties opened by double-clicking an object in the project tree with the right mouse button

This described method of editing definitions of objects goes for all objects throughout the Relay Configuration Tool, i.e. structure objects in the project tree, language objects in the graphic editor, etc.

In some cases, items must be marked before taking a certain action on them. For example, it is possible to select a single POU in the project tree for printing. Objects are marked by moving the mouse cursor onto the appropriate object and pressing the left mouse button once (left mouse click).

Finally, the appropriate action can be taken on it (i.e. drag a function in the graphic editor, display reference information by opening the help function, or selecting the print function if the object is a POU or worksheet in the project tree).

A mouse may not be available when you operate the Relay Configuration Tool in industrial environments. For this reason, the Relay Configuration Tool supports full keyboard operation. There are specific keyboard shortcuts for the single editors and
Relay Configuration Tool
Quick Start Reference

2. General handling in the Relay Configuration Tool

functional windows in the Relay Configuration Tool. Refer to the general Help for Relay Configuration Tool for the keyboard shortcuts.

2.4. The toolbar in the Relay Configuration Tool

The toolbar in Relay Configuration Tool is composed of two parts: the general part and the specific part.

The specific part only appears with the editors in which they can be used. It can differ from editor to editor (see Tables 2.4.-1 and 2.4.-2).

Toolbar icons, which are not available in the current state of the project or of the specific editor, are gray. For more detailed information about the toolbar functions, please refer to the context-sensitive Help.

![Relay Configuration Tool](image)

**Fig. 2.4.-1** The toolbar in the Relay Configuration Tool

The toolbar functions in the Relay Configuration Tool:

**Table 2.4.-1**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>![Mark mode]</td>
<td>Mark mode</td>
</tr>
<tr>
<td>![Connect objects]</td>
<td>Connect objects</td>
</tr>
<tr>
<td>![Create text]</td>
<td>Create text</td>
</tr>
<tr>
<td>![Create function or FB]</td>
<td>Create function or FB</td>
</tr>
</tbody>
</table>
2. General handling in the Relay Configuration

Quick Start Reference

The toolbar functions only available in SFC:

**Table 2.4.-1**

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Create variable" /></td>
</tr>
<tr>
<td><img src="image" alt="Insert Connector /Jump or Label" /></td>
</tr>
</tbody>
</table>

Table 2.4.-2

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Insert SFC branch" /></td>
</tr>
<tr>
<td><img src="image" alt="Create step/function sequence" /></td>
</tr>
<tr>
<td><img src="image" alt="Create action" /></td>
</tr>
<tr>
<td><img src="image" alt="Create divergence" /></td>
</tr>
</tbody>
</table>
3. Programming with the Relay Configuration Tool

3.1. General characteristics of the IEC 1131-3 standard

This chapter describes the main characteristics of the IEC 1131-3 standard in general. For making full use of the advantages of IEC definitions, it is recommended to read the standard or take part in a training course for the Relay Configuration Tool.

The international standard IEC 1131-3 describes the programming languages for industrial controller systems. The programming languages are the graphic languages SFC - Sequential Function Chart, FBD - Function Block Diagram and LD - Ladder Diagram; and the textual languages IL - Instruction List and ST - Structured Text. (Languages FBD, SFC and IL are supported in the Relay Configuration Tool).

The program source items are described as program organization units (POU), being functions (FU), function blocks (FB) and programs. These are defined in the programming languages listed above.

One of the most important features of the IEC 1131-3 standard is the method of user defined reusable code declaration.

Furthermore, the set up of target systems is defined. The range of industrial controller systems varies from highly sophisticated, multitasking and multiprocessor systems down to single processor and single task systems. The IEC 1131-3 standard defines the structure of configurations (PLC), resources (CPU) and tasks. The programs described above are assigned to the tasks of a CPU. Data exchange within the different POUs is also defined.

3.2. General project development with the Relay Configuration Tool

The general project development with the Relay Configuration Tool can be divided into several distinct steps. The steps are:

- Define project structure in project tree (software and target structure).
- Define the single program organization units.
- Assign programs to target.
- Compile, download program.

The project structure is basically set with the project tree. The possibilities of handling the project tree are described in Chapter 4. The project tree editor.

The single program organization units consist of a variable declaration and a code body definition. The code body is written with the IEC 1131-3 programming languages IL, FBD and SFC. The editors for the definition of code bodies and variable declarations are described in Chapters 6.Text editors and 7. The physical part of the project.

Associating the programs to the target tasks is done with the project tree.

Compilation of the program is described in Chapter 8. Compiling.
3. Programming with the Relay Configuration

Quick Start Reference

Fig. 3.2.-1 General project development procedure

The single steps of the project development illustrated in Fig. 3.2.-1, does not have to be strictly followed by the user. It depends on the preferred work flow (bottom up, top down).
4. The project tree editor

4.1. The project tree in the Relay Configuration Tool

The main project management tool in the Relay Configuration Tool is the project tree. The project tree is a structuring item with reference to the IEC structure of the project. All POUUs to be defined are included in the main POU file of the project tree. The target specific structure is structured in the main Target file. A further main file contains libraries connected to the project.

The project tree in the Relay Configuration Tool represents the complete IEC structure of the project. The aim of the project tree is to have a central tool from where each single definition can be accessed. All editors and project specific tools are started from the project tree.

Main files are located on the far left of the project tree. Each file consists of an icon representing the type of file followed by the description of the file. A file is opened and closed by moving the mouse cursor onto the file and double-clicking the left mouse button. After opening a file, the content of the file is displayed.

A file can contain further files or bottom elements. Bottom elements, (i.e. worksheets of POU declarations, the instance tree of a resource or the control dialog), are directly opened by moving the mouse cursor onto the description of the element and double-clicking the left mouse button.
4. The project tree editor

Quick Start Reference

4. The project tree editor

Fig. 4.1.-1  Project tree

4.2. Inserting items in the project tree

The structure of the project tree can be edited. This mainly concerns the subtree 'Logical POUs' and the subtree 'Physical Hardware'. To define a new POU, you must insert it into the project tree by marking a POU and then pressing <INS>. The dialog 'Insert' is displayed (see Fig. 4.2.-1):
Fig. 4.2.-1  Insert dialog of new POU

After you have set the properties of the POU, all items concerning the new POU (Variables, code and eventually further POU dependent items like textual description sheets) are then listed into the sublevel of the POU file. You can open the single definition editors by double-clicking the left mouse button on each of these items in the project tree. The handling of the editors is described in Chapters 5. Graphic editor and 6. Text editors.
The same method of editing the project tree (adding further items) goes for the physical part of the project. Several configurations can be added into the main physical hardware file. Inside of a configuration level, further resources can be inserted. Inside a resource, further tasks can be added. Each time during insertion, a properties dialog is displayed, in which you can set the specific characteristics of the items that will be inserted.

After you have added tasks to the task file of a resource, program instances of the program types defined in the logical POU part can be set to the single tasks. To assign a program to a task, open the properties dialog of the task by right double-clicking on the task icon. The exact handling of the physical part of a project is described in Chapter 7. The physical part of the project.
4.3. Deleting items in the project tree

If you want to delete items from the project structure, mark the item to be deleted and click the delete function (e.g. by pressing <DEL>). A safety dialog is displayed. When you have confirmed the safety dialog the item and all sublevels of the appropriate item are deleted from the project. There is no possibility of restoring deleted items!
4. The project tree editor Quick Start Reference
5. Graphic editor

5.1. The graphic editor - easy programming in SFC and FBD

The definitions of POUs language types SFC and FBD are programmed with the graphic editor. The handling of the graphic editor is the same, whichever language is used. Mixed representation of the graphic programming languages is supported. A specific toolbar for the graphic editor contains the essential edit functions and modes (see Section 2.4. The toolbar in the Relay Configuration Tool).

The main characteristics of the graphic editor are:

- The toolbar contains the most important editor functions for simple operations (see Section 2.4. The toolbar in the Relay Configuration Tool).
- The zoom function ensures a clear overview of complex graphic representations.
- Mark ranges and Drag & Drop while connections remain.
- Mixed language programming (mixture of SFC and FBD in a single worksheet).
- Dialog level for modifying the characteristics of the single language elements by right double-click.

5.2. Opening the graphic editor

You can start the graphic editor by double-clicking with the left mouse button on a POU code body file in the project tree. The POU code body file is of the type SFC or FBD.
5. Graphic editor

Fig. 5.2.-1 Opening the graphic editor

5.3. Editing SFC

To initially edit a SFC network, select the 'Mark mode' icon in the toolbar. Set an insertion mark by clicking with the left mouse button on the editing area and click the icon 'Create step/transition sequence'. A minimal legal SFC network is inserted. The properties of single objects are accessible at any time by right double-clicking on the appropriate object.
The minimal SFC network structure consists of a closed loop of initial Step connected to an Action block and a subsequent Transition. The SFC structure is enhanced by marking the Step or Transition and clicking the icon 'Create step/transition sequence' in the toolbar.

By pressing the icon 'Create step/transition sequence', further Steps and Transitions can be added to the present SFC loop structure. The last inserted Step is marked. The size of the loop is automatically adjusted.

Insertion of simultaneous / alternative branches is performed by following the same principle. To insert an alternative branch mark a Step and click the icon 'Insert SFC branch'. The alternative branch is inserted underneath the marked Step. To insert a simultaneous branch, select the Transition under which you want to insert the Transition and click the icon 'Insert SFC branch'.
The properties of the single objects can be set in the properties dialogs of the objects. The properties dialog is displayed by double-clicking the right mouse button on the appropriate object. Objects in SFC language are all language items: Steps, Transitions, Action blocks and Variables.

Further Action blocks can be added to a Step by marking the present Action block and pressing the icon 'Create action'.

If the Action block should consist of a Detail (i.e. code which is run while the Step is active according to the qualifiers of the Action block), the action code body worksheet must be inserted below the icon ‘Actions’ in the project tree.

Apart from variable worksheets and code definitions worksheets, SFC POUs have directories for Action and Transition details in the project tree. They can be inserted or deleted in the project tree as described in Chapter 4. The project tree editor.
The code bodies of Actions and Transitions, are accessible by double-clicking the left mouse button on the appropriate item, either in the project tree or in the graphic editor. You can define Action and Transition details in the languages FBD and IL. If the Action is a Variable, the name of the Action directly corresponds with the variable name. The Variable must be defined in the variables declaration of the SFC POU as a Boolean data type. This can be done in two ways:

- declare the Variable before editing your SFC network by using the variable editor or the dialog 'Variable Dialog'.
- declare the Variable while editing in SFC. In this case, just enter the name for the new Variable in the dialog 'Action' and confirm the dialog. Automatically the dialog 'Automatic variable declaration' appears. In this dialog the variable can be declared. The Relay Configuration Tool inserts the Variable automatically in the variable worksheet of the POU.

The properties of a Transition contain the information whether the Transition is a Detail or a Direct connection. If the type is set to Direct connection, the Transition can be connected to other graphic language items, that is functions, function blocks and Variables.

5.4. Editing FBD

The philosophy of programming in FBD is slightly different than editing SFC. Language items are functions and function blocks, that can be set at any location and finally be connected to other items.

To insert a function or a function block, place the marker where you want the upper left corner of the function block and click the left mouse button. Click 'Create function or FB' button in the toolbar (see Fig. 2.4.-1). A selection dialog is displayed, in which you can specify the function or function block that you want to insert.
The properties of FBD objects are also available by double-clicking the right mouse button. The FBD objects consist of functions and function blocks, variables, formal parameters and connection lines.

To connect items, select the 'Connect objects' mode in the toolbar. The mouse cursor shape indicates the active connection mode. You can connect items by moving the mouse cursor onto the first connection, press the left mouse button, move the mouse cursor onto the second connection and press the left mouse button again.

In Fig. 5.4.-3 below, the type of the Transition has been set to Direct connection via the properties dialog of the Transition.
Fig. 5.4.-3 An example of mixed graphic languages: The output of the AND function is connected to the Transition.

The principle of performing connections is to navigate the connection line from source to destination. The direction of the connection is changed by mouse movement. To set a corner (i.e. change direction), click the left mouse button while navigating.

Connections can be set between inputs and outputs of functions and function blocks, Variables and already existing connection lines.

To connect function or function blocks inputs and outputs you can also use the drag & drop function. Move the appropriate function or function block so that the connections overlap.

Fig. 5.4.-4 Two items before connection
Fig. 5.4.-5  The AND function has been dragged & dropped, so that the connections overlap.

Fig. 5.4.-6  The items have been connected. The AND function can be dragged to any other location, the connection remains.

The easiest method of connecting variables to the inputs or outputs of functions and function blocks is to mark the appropriate connection, and click the icon 'Create variable' in the toolbar. The dialog 'Variable' appears.
Fig. 5.4.-7 Connection of an input to a Variable, an automatically generated name is suggested.

If there are already declared Variables in the variable editor, these Variables can be chosen in the list box of the variable dialog. You can add new Variables by entering a name for the Variable and confirming the dialog. The dialog 'Automatic variable declaration' appears. Via this dialog you can define the Variable. The Relay Configuration Tool inserts the variable declaration automatically in the variable worksheet of the POU.

After confirming the dialog, the Variable is connected with the input.
Variables can also be inserted somewhere in the editing area by setting an insertion mark and clicking the icon ‘Create variable’ in the toolbar.

In the following, some general editing possibilities of the graphic editor are described.

Single objects can be moved with the drag & drop function. First of all, mark the appropriate object, hold down the left mouse button and move the mouse. Drag the object to a certain position according to the mouse movement. After you have released the left mouse button, the object remains at the new location, if no collision conflict occurs.

To mark several distinct objects at the same time, press <SHIFT> and hold it down, while you click the wanted objects. The drag & drop function also works for several objects marked this way.

Several directly connected objects can be marked by pulling a rectangle over the appropriate objects: Press the left mouse button, hold it down and finally move the mouse. A rectangle is displayed, indicating the marked area. After releasing the mouse button, the objects covered by the rectangle are marked. Further distinct objects can be marked by following the principle described above.

You can delete objects by marking the appropriate object and selecting the deletion function. An undo function allows restoring the network before the last action was taken on it (i.e. delete a network and undo, the original display is restored).
6. **Text editors**

6.1. **The text editors - easy programming in IL**

The handling of the text editors in the Relay Configuration Tool is identical to working with a normal ASCII editor. Furthermore, for IL and variable declarations a dialog level eases IEC 1131-3 programming, as the elements (operators, keywords, etc.) are listed in single fields in the dialog. The type of dialog depends on the type of the declaration, i.e. the dialog in a variables declaration contains only the elements for variable declarations.

The main characteristics of the text editor are:

- The toolbar contains the most important editor functions for simple operations (see Section 2.4. The toolbar in the Relay Configuration Tool. The specific part in the toolbar is not in use in IL).
- The zoom function enables a clear overview of the text.
- Dialog level for modifying the characteristics of the single language elements by double-clicking the right mouse button.
- Find, find & replace functions.
- Clipboard support.

6.2. **Opening the text editor**

The text editor is used for the declaration of any textual definitions, i.e. code definitions in IL, variables declarations, data type definitions and description texts. The editor is opened by double-clicking the left mouse button on one of the items in the project tree.

![Example of a worksheet in IL](Image)

*Fig. 6.2.1  Example of a worksheet in IL*
6. Text editors

6.3. Dialog in the IL editor

For editing IL code bodies, a dialog is available. The IL dialog of the text editor is opened by double-clicking the right mouse button on a line. The dialog contains the definition of a single line in the text editor. Each single expression of the line is listed in distinct fields. List boxes contain available IEC 1131 operators, data types, etc.

![IL Dialog](image)

Fig. 6.3.-1  IL Dialog for editing in IL

6.4. Variable declarations in the Relay Configuration Tool

In the Relay Configuration Tool, you have three possibilities for declaring Variables:

- Declaring Variables while editing a code body
- Declaring Variables using the variable editor
- Declaring Variables using the variable dialog

**Declaring Variables while editing a code body** means inserting a Variable in a code body worksheet which has not been declared before. In this case, the dialog 'Automatic variable declaration' appears on the screen, where you can define the Variable. By confirming this dialog, the variable declaration is automatically inserted in the variable worksheet.

**Declaring Variables using the variable editor** means declaring Variables by typing the declarations of Variables or function block instances in the variable editor. In this case, the menu item 'Variable Dialog' in the submenu 'Edit' has to be unchecked. Double-click the left mouse button on the icon 'Variable declaration' in the project tree, and the variable editor with the corresponding variable worksheet is opened.

**Declaring Variables using the variable dialog** means using a dialog instead of the variable editor. In this case the menu item 'Variable Dialog' in the submenu 'Edit' has to be checked. Double-click the left mouse button on the icon 'Variable declaration'
in the project tree and the dialog 'Declaration of variables and FB instances' appears. In this dialog you can define the variables or function block instances.
6. Text editors

Quick Start Reference
7. The physical part of the project

7.1. The physical part

In addition to the definition of the software of a project, also the physical part of the project has to be defined. The physical part of the project is structured in the subtree 'Physical Hardware' of the project tree. The physical part of the project consists of:

- Configurations (PLCs)
- Resources (CPUs)
- Task
- Program instances

In the subtree 'Physical Hardware', also the necessary operational items for downloading and online are available.

![Fig. 7.1.1 The physical part of the project tree](image)

The terms configuration and resource are IEC 1131 terms for the target system. The common interpretation of a configuration is an overall PLC system. A resource is interpreted as a CPU. On configuration level, the configuration is inserted by clicking the icon 'Physical Hardware' in the subtree and selecting the insert function, or pressing insert.
7. The physical part of the project

7.2. Resources

The resource (CPU) forms the main level for commissioning. You can set the properties of a resource when you insert a resource. The properties are accessible by right double-clicking on the resource in the project tree. You can insert a resource by marking the configuration file and selecting the insert function via the menu, keyboard or toolbar (refer to general handling of project tree).

7.3. Tasks

The task level allows implementation of several tasks on the resource level. Tasks are inserted via the project tree insertion function. On the same level, program instances can be set.

![Insertion dialog on task level](https://example.com/insertion_dialog.png)

**Fig. 7.3.-1 Insertion dialog on task level**

Task specific settings can be set via the button 'Settings...'. When you select a program that will be inserted, the list box contains the available program types of the logical POU section of the project tree.
8. Compiling

8.1. Taking programs into operation

In the following, the steps of taking programs into operation are described. First of all, after creating a project and setting the target specific information in the project tree, it must be compiled to the target specific code.

8.2. Compiling

Generally, after creating a new project, the program must be compiled and downloaded to the target. After finding program errors, it is possible to easily modify the program part and compile only the changed worksheets.

There are several possibilities of compiling created programs in the Relay Configuration Tool. Often used alternative is the Make function (to be selected in the menu or in the toolbar). The Make function compiles all changed or new created worksheets. These changed worksheets are displayed with an asterisk in the project tree. After compiling, the asterisks disappear.

The compilation modes are described in detail in the context-sensitive help for the Relay Configuration Tool.

8.3. Compilation errors in the user error list

Should compilation errors occur, they are displayed in a specific window. The error window makes it possible to access error locations by double-clicking the left mouse button on the error message so that the errors can be corrected as fast as possible. The corresponding worksheet is opened and the object or line with the error is marked. When you have corrected the first error, you can press <F7> to go directly to locate of the next error.

![Fig. 8.3.-1 Error window - an undefined Variable has been used in the source](image_url)

In the user error list, also warnings are displayed.
8. Compiling

Quick Start Reference
Index

C

Clipboard
  In the text editors ................................................................. 33
Compilation errors
  User error list ........................................................................ 39
Compiling
  Compilation errors .................................................................. 39
  Make ..................................................................................... 39
  Taking programs into operation ............................................. 39
CPU
  A physical part of a project .................................................. 37
  Main level for commissioning ............................................... 38
CPUs
  A physical part of a project .................................................. 37

D

Delete
  Items of a project ............................................................... 21
Drag & Drop
  In the Graphic editor .......................................................... 23
  Marking several objects using drag & drop ......................... 32
  Used for connecting functions ............................................. 29

E

Editor
  Graphic editor ....................................................................... 23
  Language editors .................................................................. 7
Errors
  User error list ....................................................................... 39

F

FBD
  Connecting items .................................................................... 28
  Connection lines ..................................................................... 28
  Direct connection ................................................................... 28
  Editing ................................................................................... 27
  Formal parameters .................................................................. 28
  Function blocks ..................................................................... 28
  Functions ............................................................................... 28
  Insertion of function blocks ............................................... 27
Function blocks
  Inserting in FBD ................................................................... 27

G

Graphic editor
  Deleting objects ...................................................................... 32
  Drag & Drop .......................................................................... 23
  easy programming in SFC and FBD ...................................... 23
  Editing FBD .......................................................................... 23
  Editing SFC ........................................................................... 24
  General editing possibilities ............................................... 32
  Main characteristics ............................................................ 23
Index

Relay Configuration Tool

Quick Start Reference

Marking several objects .............................................................. 32
Mixed language programming .................................................. 23
Opening .................................................................................... 23
Tasks ....................................................................................... 37
Zoom ....................................................................................... 23

H

Help
Context-sensitive help .............................................................. 9

I

IEC 1131-3 standard
FB ................................................................. 15
FU ................................................................. 15
Function blocks ........................... 15
Functions ................................................................. 15
General characteristics ................................................................. 15
POU ................................................................. 15
Program organization units ............................................................... 15
Insert
Configuration to the physical part of a project ............................................ 37
Function blocks ................................................................. 27
Items to project ................................................................. 18

L

Languages
FBD ................................................................. 7
IL ................................................................. 7
SFC ................................................................. 7

M

Mode
Connect object mode .............................................................. 28

O

Objects
Deleting objects ................................................................. 32
Marking several objects .............................................................. 32

P

Password protection
In projects ................................................................. 9
PLC
A physical part of a project ............................................................. 37
PLCs
A physical part of a project ............................................................. 37
POU
Contents of ................................................................. 15
Program instance
A physical part of a project ............................................................. 37
Programs
Assigning a program to task ............................................................. 20
Project tree
Deleting items ................................................................. 21
Inserting items ................................................................. 18
### Projects
- Deleting items ................................................................. 21
- General development steps ............................................ 15
- Inserting items .............................................................. 18
- Libraries ...................................................................... 17
- Password protection ...................................................... 9
- Program instances .......................................................... 37
- Project tree .................................................................. 17
- The physical part ............................................................. 37

### Relay Configuration Tool
- Declaration of variables ................................................ 34
- Development procedure ............................................... 15
- Exiting ........................................................................ 9
- General features ............................................................. 7
- General project development steps ................................... 15
- Introduction ................................................................ 7
- Libraries .................................................................. 17
- POUs ........................................................................ 15
- Programming ................................................................. 15
- Project tree .................................................................. 17
- Starting ....................................................................... 9
- Supported IEC 1131-3 programming languages ...................... 15
- Taking programs into operation ....................................... 39
- Text editors .................................................................. 33
- The physical part of a project ........................................... 37
- Toolbar ....................................................................... 13

### SFC
- Action blocks ................................................................. 26
- Editing ....................................................................... 24
- Insertion of simultaneous / alternative branches ................. 25
- Steps ........................................................................ 26
- Transition details .......................................................... 26
- Transition properties ....................................................... 27
- Transitions ................................................................ 26
- Variables .................................................................. 26

### Task
- A physical part of a project ............................................... 37
- Assigning a program to task ............................................ 20

### Text editors
- Clipboard ...................................................................... 33
- Easy programming in IL ................................................ 33
- Find/Find and Replace functions ...................................... 33
- Main characteristics ........................................................ 33
- Opening ...................................................................... 33

### The graphic editor
- Easy programming in SFC and FBD ................................... 23

### Toolbar
- Connect objects ............................................................. 13
- Create action ................................................................ 14
Index

Quick Start Reference

Create divergence ................................................................. 14
Create function or FB .......................................................... 13
Create step/function sequence ............................................. 14
Create text ............................................................................ 13
Create variable .................................................................... 14
Insert Connector/Jump or Label ............................................. 14
Insert SFC branch ............................................................... 14
Mark mode ........................................................................... 13
Transition
  Properties ........................................................................... 27

V

Variables
  Auto-insertion of variable in POU worksheet ..................... 31
  Connecting to input/output .................................................. 30
  Declaration in Relay Configuration Tool ......................... 34
  Declaration using variable dialog ..................................... 34
  Declaration using variable editor ..................................... 34
  Declaration while editing code body ................................. 34

W

Worksheets
  Compilation errors ............................................................. 39
  Compiling with Make ........................................................ 39

Z

Zoom
  In the graphic editor ......................................................... 23
  In the text editor ............................................................... 33