SIEMENS

SIMATIC HMI

**WinCC flexible 2005
Migration**

User's Manual

This user manual is part of the documentation package with the order number
6AV6691-1AB01-0AB0

Edition 06/2005
A5E00280194-02

Safety Guidelines

This manual contains notices you have to observe in order to ensure your personal safety, as well as to prevent damage to property. The notices referring to your personal safety are highlighted in the manual by a safety alert symbol, notices referring to property damage only have no safety alert symbol. These notices shown below are graded according to the degree of danger.

Danger

indicates that death or severe personal injury **will** result if proper precautions are not taken.

Warning

indicates that death or severe personal injury **may** result if proper precautions are not taken.

Caution

with a safety alert symbol, indicates that minor personal injury can result if proper precautions are not taken.

Caution

without a safety alert symbol, indicates that property damage can result if proper precautions are not taken.

Notice

indicates that an unintended result or situation can occur if the corresponding information is not taken into account.

If more than one degree of danger is present, the warning notice representing the highest degree of danger will be used. A notice warning of injury to persons with a safety alert symbol may also include a warning relating to property damage.

Qualified Personnel

The device/system may only be set up and used in conjunction with this documentation. Commissioning and operation of a device/system may only be performed by **qualified personnel**. Within the context of the safety notes in this documentation qualified persons are defined as persons who are authorized to commission, ground and label devices, systems and circuits in accordance with established safety practices and standards.

Prescribed Usage

Note the following:

Warning

This device may only be used for the applications described in the catalog or the technical description and only in connection with devices or components from other manufacturers which have been approved or recommended by Siemens. Correct, reliable operation of the product requires proper transport, storage, positioning and assembly as well as careful operation and maintenance.

Trademarks

All names identified by ® are registered trademarks of the Siemens AG. The remaining trademarks in this publication may be trademarks whose use by third parties for their own purposes could violate the rights of the owner.

Preface

WinCC flexible Migration

Purpose of this manual

This user manual is part of the WinCC flexible documentation..

Projects created with ProTool or WinCC can be re-used with WinCC flexible.

This manual contains the following information:

- Basic principles of migration
- HMI device for which the migration is performed without replacement of an HMI device.
- HMI device for which the migration is performed with replacement of an HMI device.
- Migration from OP3 to OP 73 or OP 73 micro
- Migration from OP7 to OP 77B or OP 77A
- Migration from OP 17 to OP 177B
- Replacement of ProTool objects with WinCC flexible objects

This manual is intended for project engineers, commissioning engineers and service technicians who are involved in migration to WinCC flexible.

The help integrated in WinCC flexible, the WinCC flexible Information System, contains detailed information. The information system contains instructions, examples and reference information in electronic form.

Basic Knowledge Requirements

General knowledge in the field of automation engineering is required to understand this manual.

You should also have experience of using PCs running under the Windows 2000 or Windows XP operating systems. A knowledge of VBA or VBS is required for advanced configuration by using scripts.

Scope of the manual

This manual is valid for the WinCC flexible 2005 software package.

Position in the information scheme

This manual is part of the SIMATIC HMI documentation. The information below presents an overview of the information landscape of SIMATIC HMI.

User manual

- WinCC flexible Micro

  – describes the engineering basics based on the WinCC flexible Micro engineering system (ES)

- WinCC flexible Compact/ Standard/ Advanced

  – describes the engineering basics based on the WinCC flexible Compact, WinCC flexible Standard and WinCC flexible Advanced engineering systems (ES)

- WinCC flexible Runtime:

  – Describes how to commission and operate your Runtime project on a PC.

- WinCC flexible Migration:

  – Describes how to convert an existing ProTool project to WinCC flexible.

  – Describes how to convert an existing WinCC project to WinCC flexible.

  – Describes how to migrate ProTool projects with an HMI migration from OP3 to OP 73 or OP 73 micro.

  – Describes how to migrate ProTool projects with an HMI migration from OP7 to OP 77B or OP 77A.

  – Describes how to migrate ProTool projects with an HMI migration from OP 17 to OP 177B.

  – describes how to migrate ProTool projects with HMI migration from RMOS graphic devices to Windows CE devices.

- Communication:

  – Communication Part 1 describes the connection of the HMI device to SIMATIC PLCs.

  – Communication Part 2 describes the connection of the HMI device to third-party PLCs.

Operating Instructions

- Operating instructions for SIMATIC operating units:

  – OP 73, OP 77A, OP 77B

  – TP 170micro, TP 170A, TP 170B, OP 170B

  – OP 73micro, TP 177micro

  – TP 177A, TP 177B, OP 177B

  – TP 270, OP 270

  – MP 270B

  – MP 370

- Operating instructions for mobile SIMATIC operating units:

  – Mobile Panel 170

- Operating instructions (compact) for SIMATIC operating units:

  – OP 77B

  – Mobile Panel 170

Getting Started

- WinCC flexible for first time users:

    – Based on a sample project, this is a step-by-step introduction to the basics of configuring screens, alarms, and recipes, and screen navigation.

- WinCC flexible for advanced users:

    – Based on a sample project, this is a step-by-step introduction to the basics of configuring logs, project reports, scripts, user management, and multilingual projects, and integration into STEP 7.

- WinCC flexible options:

    – Based on a sample project, this is a step-by-step introduction to the basics of configuring the WinCC flexible Audit, Sm@rtServices, Sm@rtAccess and OPC Server options.

Online availability

The following links provide direct access to technical documentation on SIMATIC products and systems in English, German, French, Italian, and Spanish.

- SIMATIC Guide Technische Dokumentation in Deutsch:
  "http://www.ad.siemens.de/simatic/portal/html_00/techdoku.htm"

- SIMATIC Guide for Technical Documentation in English:
  "http://www.ad.siemens.de/simatic/portal/html_76/techdoku.htm"

Guide

Structure of this manual:

- Basic - Chapter 1

- Migration without replacement of HMI devices - Chapter 2

- Migration with replacement of HMI devices - Chapter 3

- Migration from OP3 to OP 73 - Chapter 4

- Migration of OP7 to OP 77B - Chapter 5

- Migration of OP7 to OP 177B - Chapter 6

Conventions

A distinction is made in the naming conventions for the configuration and runtime software:

- "WinCC flexible 2005" refers to the configuration software.

- "Runtime" designates the runtime software running on the HMI devices.

- "WinCC flexible Runtime" designates the visualization product for use on standard PCs or panel PCs.

The term "WinCC flexible" is used in the general context. A version name such as "WinCC flexible 2005" is used whenever it is necessary to distinguish it from other versions.

The following formatting is used to facilitate reading of the manual:

Notation	Scope
"Add screen"	• Terminology that occurs in the user interface, e.g., dialog names, tabs, buttons, menu commands. • Inputs required, e.g., limit values, tag values • Path information
"File > Edit"	Operational sequences, e.g., menu commands/shortcut menu commands.
<F1>, <Alt>+<P>	Keyboard inputs

Please pay particular attention to such notices:

Note

Notes containing important information about the product and its use or a specific section of the documentation to which you should pay particular attention.

Trademarks

HMI®
SIMATIC®
SIMATIC HMI®
SIMATIC ProTool®
SIMATIC WinCC®
SIMATIC WinCC flexible®

Third parties using for their own purposes any other names in this documentation which refer to trademarks might infringe upon the rights of the trademark owners.

Further Support

If you have any technical questions, please get in touch with your Siemens representative or agent responsible.

You will find your contact person at:

"http://www.siemens.com/automation/partner"

You will find a guide to the technical documentation offered for the individual SIMATIC Products and Systems here at:

"http://www.siemens.com/simatic-tech-doku-portal"

The online catalog and order system is found under:

"http://mall.automation.siemens.com/"

Training Centers

Siemens offers a number of training courses to familiarize you with the SIMATIC S7 automation system. Please contact your regional training center or our central training center in D 90327 Nuremberg, Germany for details:

Telephone: +49 (911) 895-3200.

Internet: "http://www.sitrain.com"

Technical Support

You can reach the Technical Support for all A&D products

- Via the Web formula for the Support Request
  "http://www.siemens.com/automation/support-request"

- Phone: + 49 180 5050 222

- Fax: + 49 180 5050 223

Additional information about our Technical Support can be found on the Internet pages "http://www.siemens.com/automation/service"

Service & Support on the Internet

In addition to our documentation, we offer our Know-how online on the internet at:

"http://www.siemens.com/automation/service&support"

where you will find the following:

- The newsletter, which constantly provides you with up-to-date information on your products.

- The right documents via our Search function in Service & Support.

- A forum, where users and experts from all over the world exchange their experiences.

- Your local representative for Automation & Drives.

- Information on field service, repairs, spare parts and more under "Services".

Table of contents

Basics 1

1.1 Basic principles of migration

Introduction

Projects created in ProTool or WinCC can be re-used with WinCC flexible. "Migration" must be installed on the engineering computer to enable the migration of a ProTool or WinCC project to WinCC flexible. "Migration" is part of the basic WinCC flexible package.

What happens during migration

To convert a ProTool or WinCC project, open the project with WinCC flexible. WinCC flexible identifies the project file by its file name extension. Next, WinCC flexible starts migration, and creates a WinCC flexible project that contains all objects required to map the project to WinCC flexible.

Note

The migration includes all screens, tags and alarms in the ProTool project. Of other object groups, such as the user scripts, only the objects actually used in the ProTool project will be included.

During migration, WinCC flexible creates a log file which returns information on the type and number of converted objects, and on any problems that may have occurred during conversion.

If the HMI for which a ProTool project was configured is not supported by WinCC flexible, change to an HMI that is supported by WinCC flexible for migration.

Adaptation test after migration

During migration, some ProTool functions may not be converted 1:1 to WinCC flexible functionality, due to various different functions in WinCC flexible. Depending on the content of the ProTool project, it may be necessary to reconfigure the converted project.

For example, WinCC flexible supports the setup of user groups with different authorizations. This concept replaces the control of user authorizations at password level used in ProTool.

During migration of WinCC projects, not all components of the project are imported but the migrated project can be used as the starting basis for a new WinCC flexible project. Complete migration of WinCC projects will be possible with WinCC flexible Version 2, because this version provides full support of WinCC functionality.

1.2 Convertible projects

Migration depending on the HMI

You can basically migrate all ProTool projects created for graphics devices or Windows-based systems to WinCC flexible.

Line devices only support the migration of OP7 projects. During migration of projects for those HMIs, an OP 77B HMI device is automatically selected. OP3, OP17, TD 17 line devices and C7 devices do not support the migration of projects.

The project must be available as a ProTool project file (*.pdb, *.pdz) to allow its migration.

In the following cases, you require a ProTool engineering station to restore the project prior to migration:

- You only have a backup of the project file (*.pdz) created with ProTool V5.2 or an earlier version located on an HMI. Transfer the project to a ProTool engineering station. Do this using the ProTool function "File" > "Upload."

- You have a compressed project (*.arh). Select the compressed project on a ProTool engineering station using the "File" > "Project Manager" command. Click "Restore" to unpack the project data and save those data to a project file (*.pdb).

Scope of migration

The scope of migration depends on the HMI you will use to run the project:

- HMIs supported in ProTool and WinCC flexible:

  To run a project on an HMI that is supported in WinCC flexible and ProTool, the system automatically migrates the ProTool project when you open it in WinCC flexible.

- HMIs supported only in ProTool and not in WinCC flexible:

  If the project was configured for an HMI that is only supported in ProTool and not in WinCC flexible, the HMI must be changed for migration.

  At the start of migration, you will be requested to select one of the HMIs supported in WinCC flexible.

- OP7:

  When you open a ProTool project for OP7 in WinCC flexible, the project will be automatically converted into a WinCC flexible project for an OP 77B.

  If the OP7 project has more than one editing language, select one of the Runtime languages as the reference language before you start the conversion. The reference language influences the conversion of screens which consist of static texts and IO fields positioned between the texts. The layout of text elements and IO fields will be optimized for the selected reference language during migration. When using the other Runtime languages, you may have to edit the relevant screens.

Migration without changing the HMI

Projects for the HMIs listed below can be migrated directly and without changing to another HMI:

- Standard PC
- Panels:

  TP 170A, TP 170B, TP 170B Color, OP 170B, TP 270, OP 270
- Mobile Panels:

  Mobile Panel 170
- Multi Panels

  MP 270B, MP 270B Touch, MP 370, MP 370 Touch
- Panel PCs

  PC 670, PC 670 Touch, PC 870, PC 870 Touch, PC IL 70 Touch
- SINUMERIK Panel PCs

  OP010, OP012, OP015
- SIMOTION Panel PCs
- C7 Devices

  C7-635, C7-635 Touch

Migration of projects for those HMI is starts automatically when you open the ProTool project in WinCC flexible.

Migration with change of the HMI

Graphic devices (OP25, TP25, OP27, TP27, MP 270, OP35, TP35, OP37, TP37) and certain Windows-based systems (MP 270, OP37Pro, FI25, FI45) are not supported in WinCC flexible. Hence, you need to change the HMI when migrating ProTool projects for those devices.

Line devices only support the migration of OP7 projects. During migration of projects for these HMIs, an OP 77B HMI will be selected automatically.

Migration procedure with change of the HMI:

- Select the HMI device to use at the start of migration.
- Next, the system automatically migrates the ProTool project from the old to the new HMI.
- Finally, it converts the ProTool project for the new HMI to a WinCC flexible project.

Optimizing the HMI change

The conversion quality depends on the degree of difference between the HMI of the ProTool project and the HMI of the WinCC flexible HMI project. Certain differences between the devices, such as the number of keys, dimensions or performance classes, can be compensated by editing the converted project. Other differences may restrict the functionality of the converted project, such as scope of functions, data volume or performance.

The table below shows:

- The preferred HMI for optimum results.

- For which changes you may expect restrictions.

Find the ProTool HMIs in the columns, and the WinCC flexible HMIs in the rows.

HMIs with keyboard

	ProTool - Windows-based systems			
WinCC flexible HMIs	**FI25**	**FI45**	**OP37Pro**	**MP 270**
OP 270 6"	---	---	---	---
OP 270 10"	---	---	---	0
MP 270B 10" Keys	---	---	0	++
MP 370	0	0	0	+
PC	+	+	+	+
Panel PC 670 10"	+	---	+	+
Panel PC 670 12"	++	+	+	+
Panel PC 670 15"	+	+	---	---
Panel PC 870 12"	++	+	+	+
Panel PC 870 15"	+	++	---	---

	ProTool - Graphic devices			
WinCC flexible HMIs	**OP25**	**OP27**	**OP35**	**OP37**
OP 170B	0	0	---	---
OP 270 6"	++	++	---	---
OP 270 10"	+	+	++	+
MP 270B 10" Keys	+	+	++	+
MP 370	---	---	+	++
PC	---	---	+	+

++   Preferred type: Conversion has optimum result

+   Conversion possible, but HMIs differ, for example, in the number of keys, dimensions, or in the performance class of the available IO devices.

0   Conversion possible, but there may be restrictions with respect to functionality, data volume or performance

---   Conversion is either not possible, or a highly complex task

HMIs with touch screen

	ProTool - Graphic devices		
WinCC flexible HMIs	**TP27-6"**	**TP27-10"**	**TP37**
TP 170A	0	---	---
TP 170B	0	---	---
TP 270 6"	++	+	---
TP 270 10"	+	++	0
MP 270B 6" Touch	++	+	---
MP 270B 10" Touch	+	++	0
MP 370 Touch	+	+	++

++   Preferred type: Conversion has optimum result

+   Conversion possible, but HMIs differ, for example, in the number of keys, dimensions, or in the performance class of the available IO devices.

0   Conversion possible, but there may be restrictions with respect to functionality, data volume or performance

---   Conversion is either not possible, or a highly complex task

1.3   Migrating WinCC projects

Introduction

A WinCC project can be completely migrated to WinCC flexible Version 2.

To give you some idea about WinCC flexible V.2, you can now migrate elements of WinCC.

Scope of migration

Components included in the migration of WinCC project to WinCC flexible :

- Screens

  Migration includes the basic screen objects, such as circle, rectangle, rounded rectangle, ellipse, line, polyline, polygon, static text, IO field, bars and buttons, and returns those objects in the corresponding WinCC flexible objects.

- Tags

  All tags which are interconnected with an S7 controller will be applied. The following tags, however, will not be accepted:

  – Raw data tags

  – Text reference tags

  – Floating point numbers 64-bit IEEE 754

  – Text tags 16-bit

  Structures are also excluded from migration.

- Channels

  Migration only supports S7 channels with MPI or TCP/IP protocol.

- Dynamic

  Logical links between IO fields and tags will be included. All other dynamic data are excluded from migration.

When migration is completed, you can save save the log file to any folder, and then view it in Internet Explorer, for example.

1.4 Installing Migration

Introduction

"Migration" is supplied with WinCC flexible. WinCC flexible setup automatically installs it on the engineering station if you selected the "standard" setup.

You can install the program either immediately or at any other time if you select the "user-defined" setup of WinCC flexible.

Requirements

Windows runs on the engineering station.

WinCC flexible is installed without "Migration".

You are authorized to install programs.

Installing Migration later

1. Run WinCC flexible setup.

   The "InstallShield Wizard" dialog box opens.

2. Select "Modify program", then click "Next."

   The "Select features" dialog box opens.

3. Enable the check box next to "Migration" in the feature list.

   The check boxes next the subordinate entries "ProTool Migration" and "WinCC Migration" will be set automatically.

4. If you only want to migrate ProTool projects, disable the check box next to "WinCC Migration." This accelerates Setup.

5. Click "Next."

Result

Setup installs the program. When you then open a ProTool project in WinCC flexible, Migration starts automatically.

1.5 Converting a ProTool project

Introduction

Any project created in ProTool may be used as the basis for a WinCC flexible project. Procedure:

- Convert the ProTool project into a WinCC flexible project. Depending on the HMI of the ProTool project, select an HMI that is supported in WinCC flexible for the migration.

- When the conversion is completed, check that the new project meets your requirements with respect to functionality and layout of the screens. WinCC flexible offers support by generating a log file during the automatic conversion. This log file also helps to identify elements of the old project which could not be converted automatically.

- After you debugged the new project in WinCC flexible, and edited it where necessary, you are ready to run the project.

Requirements

WinCC flexible and the "Migration" option are installed on the engineering station.

WinCC flexible is running.

A ProTool project file with the extension "*.pdb" is available for conversion.

Step 1: Conversion

1. Select "Open" from the "Project" menu. You can also use the "Open ProTool Project" command on the start page of the Project Wizard. The "Open an existing project" dialog box opens.

2. Select as "File type:" "ProTool project (*.pdb)". If you only have a backup of the project file (*.pdz) created with ProTool-Version V5.2 or an earlier version on an HMI, the project must be restored a ProTool engineering station using the ProSave "Restore" function before you run the migration. If only a compressed project (*.arh) is available, open the project on a ProTool engineering station before you run the migration, and select the "Restore" function to save it as an uncompressed project file (*.pdb).

3. Set the folder of the ProTool project.

   All ProTool projects in this folder will be displayed.

4. Enable "Integrated" in the file selection dialog if the ProTool project is integrated in STEP 7.

5. Double-click the required project.

6. If the ProTool project was configured for an HMI that is not supported in WinCC flexible, the "Change HMI device" dialog box opens.

   Select the HMI on which you are going to run the converted project.

7. If the ProTool project was configured for an OP7 and includes more than one Runtime language, the "Change reference language" dialog box opens.

   Select the language for which the position of the IO fields in the screens must be optimized.

   With an OP7, the HMI is automatically changed to an OP 77B.

Migration of the ProTool project starts. The Migration Wizard runs the migration.

When the conversion has been successfully completed, the system outputs a corresponding alarm. The converted project will be opened automatically in WinCC flexible.

Step 2: Checking and reworking

You are supported by a log file that contains information about the scope, course and result of the migration.

1. Open the log file with the link on the last page of the Migration Wizard. If you want to view the log file at a later time, it can be found in the output window.

2. Get an overview of which ProTool objects could not be mapped 1:1 to WinCC flexible objects. Check whether it is necessary to make further manual adaptations of the WinCC flexible project.

3. Open all screens successively, and check the layout of the control elements and texts.

Only for migration with change of HMI device:

1. Check the global settings. Certain settings are no longer defined in the WinCC flexible project and must be made directly in the Control Panel of the HMI.

2. Open the template in the "Sreens" editor, then check the global assignment of the function keys and system keys.

   If your ProTool project contains function keys or system keys, the "system keyboard" or "function keyboard" will be using the bits in the area pointers. In this case, the corresponding function keys and system keys must be assigned system functions in the WinCC flexible project which are used to set bits in the CPU.

   If the number of keys differs in the HMIs of the ProTool project from those in the WinCC flexible project, you can not create a 1:1 map of the key asignment. You may have to discard various key assignments altogether, or implement those in a different way, for example, by using buttons. Depending on the project, it may be useful to reassign keys, for example, to enhance user-friendly operation of the project.

3. Check the log file to identify system function calls which were not fully converted.

4. Check whether PLC jobs are still effective, and whether the triggering events were implemented correctly. Depending on your configuration, you may have to configure the destination project and the PLC program manually.

1.6 Converting a WinCC project

Introduction

A WinCC project can be completely migrated to WinCC flexible V.2.

To give you some idea about WinCC flexible V2, you can now migrate parts of WinCC projects. The migration includes in particular simple screen objects, tags interconnected with an S7 PLC, and the logic links between IO fields and tags.

Requirements

- WinCC flexible Advanced is installed.
- Migration is installed. This is true if you selected the standard setup of WinCC flexible Advanced, or the "Migration" > "WinCC Migration" option in a user-defined setup.
- A WinCC project is available (*.mcp or *.MCP).

Procedure

1. Select "Open" from the "Project" menu.

   The "Open an existing project" dialog box opens.

2. Select as "File type:" "WinCC project (*.mcp, *.MCP)".

3. Set the folder in which the WinCC project resides.

   All WinCC projects in this folder will be displayed.

4. Double-click the required project.

   Migration of the WinCC project starts. The Migration Wizard runs the migration.

5. For information on the scope of migration, refver the log file with the link set on the last page of the Migration Wizard.

When the conversion has been successfully completed, the system outputs a corresponding alarm. The converted project will be opened automatically in WinCC flexible for further editing.

Migration without changing the HMI

2

2.1 Screens

2.1.1 Screen objects

Introduction

WinCC flexible screen objects always cover the entire functionality and and visualization features of ProTool objects.

During migration, most of the objects of the ProTool project are replaced 1:1 by the corresponding WinCC flexible objects. If this is not possible, the ProTool projects wll be replaced with matching WinCC flexible screen objects and automatically configured, i.e. the full functionality and all layout features will be preserved.

IWhen using any third-party ActiveX Controls in your ProTool project, you have to add these to the "Own controls" group before you migrate the project to WinCC flexible. The control elements of this group will be included in the migration from ProTool to WinCC flexible.

Replacing the screen objects

The various screen objects are converted during migration as follows:

ProTool	WinCC flexible
Line	Line
Polyline Configuration: open polyline Configuration: polygon	Polyline Polygon
Ellipse/circle	Ellipse Circle
Rectangle	Rectangle, dimensions are configured accordingly
Rounded rectangle	Rectangle, configuration: Corners rounded

ProTool	WinCC flexible
Graphics	Graphic view
A Text	A Text field
23 Output box	aI IO field, configured as output box
Input box	aI IO field, configuration: Input box
2:37 Date/Time	aI Date/Time field
Symbolic output box	Symbolic IO field configuration: Output box
Drop-down list (input box for symbolic values)	Symbolic IO field, configuration: Input box
Graphic view	Graphic view
Graphics list	Graphic IO field, configuration: Output box
Graphic selection	Graphic IO field, configuration: Input box
Button	Button
Status button • Configuration: Switch • Configuration: Key • Configuration: Key with feedback	• Switch • Button • Button
Status bar • Configuration: Text-based status display • Configuration: Display of a status by graphic • Configuration: Text-based display of two states • Configuration: Display of two states by graphic	• A Text field • Graphic view • Symbolic IO field, configuration: Output box • Graphic IO field, configuration: Output box
Switch	Switch
Hotspot button	Button, configuration: "Hotspot"
Trend view	Trend view
Bar	Bar

ProTool	WinCC flexible
Alarm view	Alarm view, configuration: extended
Simple alarm view	Alarm view, configuration: simple
Status/Control	Status/Control
Password list	Password view
Recipe view	Recipe view The simple recipe view is integrated automatically when compiling the data for HMIs that only support the simple recipe view.
Digital/Analog clock	Clock
Slider control	Slider control
Analog indicator	Gauge

2.1.2   Adapting screen objects

Introduction

During migration, the objects of the ProTool project will be replaced with the corresponding WinCC flexible screen objects and automatically configured. When migration is completed, the WinCC flexible project returns the full functionality and all layout features of the ProTool project.

Adaptation requirements

Because the screen objects in WinCC flexible were automatically configured to match the ProTool objects, you only need to check the screens in the "Screens" editor.

Pay particular attention to the following aspects:

- Does the overall layout of the screens meet project requirements? Do objects overlap, or can I optimize the alignment of the objects?

- How does the program visualie text fields and labeled objects, such as line/page breaks and fonts in buttons?

Optimize your screen directly in the "Screens" editor as required.

Substitution for the status bar screen object of ProTool

The ProTool object status bar will be replaced during migration with appropriately configured WinCC flexible objects. If you want to configure a new control element with the functions of a ProTool status bar in the migrated project, you must configure the following WinCC objects and set them up accordingly:

Configuration of the ProTool status bar	WinCC flexible object
Switch	Switch
Key	Button
Key with feedback	Button
Text-based status display	Text field
Display of a status using graphic elements	Graphic view
Text-based display of two states	Symbolic IO field, configuration: Output box
Display of two states by graphic elements	Graphic IO field, configuration: Output box

2.1.3 Libraries

Converting libraries

Libraries can neither be migrated, nor edited.

Library objects that were included in a ProTool project are migrated like all other objects.

Note

However, you can include library objects nonetheless by editing the ProTool project before you run the migration.

First, copy the required library objects to any screens in ProTool. Next, run the migration, and then insert the migrated library objects from the relevant screens to a WinCC flexible library.

2.2 Tags and area pointers

2.2.1 Tags

Converting tags

During conversion, the program generates tags for all tags created in the ProTool project and writes these to the WinCC flexible project. In addition to the creation of relevant tags, the program also interconnect external tags with the PLC.

Conversion of multiplex tags

A multiplex tag of a ProTool project is replaced by a tag which is assigned the "Indirect Addressing" property in the WinCC flexible project. The tag list and the index tag will be implemented in this property.

Conversion of trend tags

Profile trends of a ProTool project are converted during migration into trends of the type "Buffer bit-triggered." During migration, the program always creates array tags for the trend tags configured in ProTool and writes these to the WinCC flexible project, in order to buffer the tag values ("Buffer" and "Switch buffer".)

If you have used the trend tags at any further positions in the ProTool project (for example, to display single values), you must check whether the use of an array tag is supported at these positions in the WinCC flexible project after you migrated the project.

Conversion of array tags

An array tag of a ProTool project will be replaced by a tag that is subdivided into the corresponding number of array elements in the WinCC flexible project.

Using tags in scripts

In ProTool projects, it is possible to increment the value of a tag in a script using the following expression: "tag+n". During migration, this expression is automatically converted as follows: (SmartTags ("tag")+1). To obtain a syntactically correct expression, a "=" character must be inserted manually to provide the following format of the expression: (SmartTags ("tag")=+1).

Note

To return the "TypeName" of a SmartTag-object tag type using the VBS function "TypeName", use the following syntax:

```
TypeName(SmartTags("FillLevel").value)
```

2.2.2   Area pointer

Introduction

Most area pointers are imported without change during the conversion of a ProTool project to a WinCC flexible project.

Conversion to array tags

However, the following ProTool area pointers are replaced with external array tags in the WinCC flexible project. An array element of the relevant tag corresponds to every bit of the ProTool area pointer after the migration.

- Event alarms
- Fault alarms
- OP acknowledgment
- PLC acknowledgment
- LED mapping
- Trend request 1
- Trend request 2

The array tags for the area pointers of event alarms, fault alarms and LED mapping are assigned refresh identifiers. This is necessary, because the ProTool PLC jobs 42 "Get LED range from the PLC", 43 "Get event message range from the PLC", 44 "Get alarm message range from the PLC" and 45 "Get acknowledgment range from the PLC," which have accessed the area pointer are overridden by the new PLC job "Update tag."

The following update identifiers are assigned to the array tags:

- Event alarms: 1 - 8
- Fault alarms: 9 - 16
- LED mapping: 17 - 24

Conversion of the alarm area pointers

During the conversion of a ProTool project to a WinCC flexible project, the program creates array tags to replace the area pointers "event alarms" and "fault alarms." The array tags will be named accordingly. A 16-bit array element is created for each area pointer. The alarms are assigned to the bit numbers in ascending order of alarm numbers.

When the PLC sets the value of a bit in the array element to "TRUE" status, the HMI detects the assigned event alarm as "received." Vice versa, the HMI interprets the alarm as "outgoing" when the bit in the array element is reset to "FALSE" on the PLC.

Conversion of the "PLC acknowledgment" area pointer

If the "PLC acknowledgment" area pointer exists in the ProTool project, the array tag will be extended for the fault alarms during migration. This provides an additional bit for the acknowledgment of fault alarms.

Example of the assignment of the array tag: Bit 0-15=fault alarms, Bit 16-31= acknowledgment of fault alarms.

Conversion of the LED image

During conversion of a ProTool project to a WinCC flexible project, the "LED mapping" area pointer is replaced by an identically named external array tag with a 16-bit array element. When the conversion is completed, each bit of the ProTool area pointer corresponds with one array element of the new array tags. The order by which the LEDs on the HMI device are assigned to the array elements is the same as in ProTool.

After the PLC has set the array element to "TRUE," the associated LED on the HMI will be switched on.

Converting the "Trend request 1" and "Trend request 2" area pointers

The "Trend request 1" area pointer is used to trigger trends in ProTool projects. The "Trend request 2" area pointer is required in ProTool projects if you configure trends with change buffer.

During conversion of a ProTool project to a WinCC flexible project, the program replaces the "Trend request 1" and "Trend request 2" area pointers with external array tags with a 16-bit array elements and names those accordingly. When the conversion is completed, each bit of the ProTool area pointer corresponds with one array element of the new array tags. Bit assignment remain as had in ProTool.

After the PLC program has set the array element and the trend group array element assigned to the trend, the HMI can detects the trigger event and reads a value or the entire buffer, depending on the configuration.

2.2.3   Adapting the instances of area pointers

Adapting the pointer instances

When converting the area pointers to array tags, the latter will be assigned the address in PLC memory which was assigned to the area pointers in the ProTool project. Aadaptation of the configuration is thus not required.

2.3 System functions and PLC jobs

2.3.1 System functions

Introduction

Most system functions are converted 1:1. Usually, changes only concern function names and the names of function groups.

A comparison of old and new function names is available in the WinCC flexible Information System under "System functions from A to Z."

The following changes go beyond mere renaming:

System functions for the alarm view

ProTool system functions replaced by the WinCC flexible system function "ShowAlarmWindow" during conversion:

- Display alarm page
- Display alarm buffer
- Display alarm page
- Display alarm window
- Display alarm buffer

The "Show alarm window" is configured accordingly during conversion. The alarm window that is displayed with the system function is also automatically created and configured.

Particulare changes to be made in the project:

ProTool system function	WinCC flexible system function
Display alarm page	"ShowAlarmWindow" to display an alarm window
	Configuration of the alarm window: Message class = "error alarms;" alarms are displayed
Display alarm buffer	"ShowAlarmWindow" to display an alarm window
	Configuration of the alarm window: Alarm class = "error alarms;" alarm events are displayed
Display event alarm page	"ShowAlarmWindow" to display an alarm window
	Configuration of the alarm window: Message class = "event alarms;" alarms are displayed; the column display is more comprehensive than the alarm window which is converted from the alarm event window.
Display alarm event window	"ShowAlarmWindow" to display an alarm window
	Configuration of the alarm window: Alarm class = "error alarms;" alarms are displayed;
Display event alarm buffer	"ShowAlarmWindow" to display an alarm window
	Configuration of the alarm window: Message class = "event alarms;" alarm events are displayed

System functions for password administration

Changes to be made in the system functions as a result of the different concept for user authorizations:

ProTool system function	WinCC flexible system function	Change due to migration
Write password level to tags	GetGroupNumber	System function is replaced
Write password to tag	GetPassword	System function is replaced
Report change of password	TraceUserChange	System function is replaced. However, the system function behaves differently in Runtime: A system alarm is triggered on the HMI showing which user is currently logged on to the HMI.
User logon	----	This function is not converted. Instead, configure the "LogOn" system function.

2.3.2 Job mailboxes

Converting the PLC jobs

Certain PLC jobs may be reused in WinCC flexible. Because of the different functions of the HMIs, certain PLC jobs are no longer required, or can no longer be used. You may have to edit the WinCC flexible project and your PLC program to obtain the functionality of the ProTool project.

If the destination project requires editing, check whether such tasks done with PLC jobs in the ProTool project can be done more efficiently with functions in WinCC flexible.

PLC jobs which can be reused after conversion without modification

Editing is not required after conversion with the following PLC jobs:

- 14 "Set time (BCD-coded)"
- 15 "Set date (BCD-coded)"
- 24 "Password logout"
- 40 "Transfer date/time to PLC"
- 41 "Transfer date/time to PLC"
- 51 "Screen selection"
- 69 "Read data record from PLC"
- 70 "Write record to PLC"
- 49 "Clear event buffer"
- 60 "Clear error alarm buffer"

PLC jobs which must be adapted

Instances in which you may have to replace the ProTool PLC jobs in your PLC program with WinCC flexible PLC jobs:

ProTool PLC job	WinCC flexible PLC job
23 "Set password level"	23 "Log on user" This PLC job makes allowances for the changed concept of user administration in WinCC flexible, because it sets the user group instead of the password level.
42 "Get LED range from the PLC" 43 "Get event message range from the PLC" 44 "Get alarm message range from the PLC" 45 "Get acknowledgment range from the PLC"	46 "Update tag" This PLC job reads the values of WinCC flexible tags that replace the ProTool area pointers that replace "event messages," "alarm messages,", "PLC acknowledgment" and "LED mapping" after migration. The parameter "update identifier," to which the update identifiers of the corresponding tags are assigned, is used to control which tag is read. Assignment of the parameter "update identifier": 1...8 warning alarms 9...16 error alarms 17...24 LED mapping

2.4 Alarm system

Changing the alarm number

In ProTool projects the same alarm number can be assigned for an event messages and an alarm messages. In WinCC flexible, one alarm number can be assigned once only.

During migration all message numbers of the ProTool alarm messages are increased by the value of 2000. This prevents duplicate assignment of alarm numbers to event and alarm messages.

Application example

In a ProTool project 200 alarm messages are configured with the numbers 1 to 200. If this project is converted to a WinCC flexible project, the alarm messages are given the numbers 2001 to 2200. An alarm message with the number 1 becomes an error alarm with the number 2001, etc.

Adaptation requirements

In the WinCC flexible project, all instances of the alarm numbers are automatically adapted during migration. Further adaptations are not required.

If the alarm numbers are listed in one of your user documents, such as a list of alarms with alarm numbers, this document must be adapted.

Introduction

In WinCC flexible, objects and function keys used in several screens may be stored in a central template In contrast to the permanent window, the template does not require its own screen area, but is rather active under the screen.

During migration, the following objects visualized in the permanent window of the ProTool project are stored in the template:

- Alarm view
- Alarm window
- Alarm indicator
- Globally assigned function keys

Alarms in the template

During migration, the program always generates at least one alarm window for the visualization of system alarms in the template of the WinCC flexible project.

The alarm window and the alarm bar of a ProTool project are replaced by "alarm window" and "alarm view" type objects in the template of the WinCC flexible project during conversion. The alarm view is configured in such a way that it includes only one (alarm line) line.

During conversion, the alarm window and the alarm view will be configured in accordance with the settings in the ProTool project.

Settings in the ProTool project under "PLC" > "Screen/Keys"	Contents of the template in the WinCC flexible project
Window-window	1 alarm window for displaying event alarms, display at runtime can be controlled with a button which is assigned the "ShowAlarmWindow" system function.
	1 alarm window for displaying error alarms, display mode: "Automatic display" enabled
Window row	1 alarm window for displaying error alarms, display mode: "Automatic display" enabled
	1 alarm view one row high for displaying warning alarms, display mode: "Closable" disabled
Window - Off	1 alarm window for displaying error alarms, display mode: "Automatic display" enabled
Row - row	1 alarm view one row high for displaying warning alarms, display mode: "Closable" disabled
	1 alarm view one row high for displaying error alarms, display mode: "Closable" disabled

Note

◆ Local function keys always remain active in Runtime, regardless whether the screen in which they are assigned is overlaid by an alarm view or an alarm window. This situation may arise in particular on HMI devices with a small display (OP 170B, for example).

Calling alarm windows using system functions

Functions in ProTool to display alarms will be replaced during conversion with the appropriately configured WinCC flexible "ShowAlarmWindow" system function. The program also creates and configures the alarm window which is displayed with the system function.

Alarm indicator in the template

If the alarm indicator is configured in the ProTool project, the program automatically stores the "alarm indicator" screen object the template of the WinCC flexible project.

Global key assignment

Global assignments of function keys are automatically stored in the template of the WinCC flexible project during migration. There, you can always edit the assignments when you completed the migration.

Note

A bit which is assigned to an LED in ProTool is only migrated if the corresponding key is also assigned a function.

2.5 User authorizations

New user authorization concept in WinCC flexible

The concept of user authorization in WinCC flexible differs from that in ProTool.

In WinCC flexible, each users is assigned only to one user group. You assign those user groups characteristic access rights, namely the authorizations. User groups are not organized in a hierarchical structure.

A default "Administrator" user group is created in each project. Users assigned to this group in Runtime have all rights, and may also set up new users.

You determine what rights the user requires for efficient operation of the various control elements.

In Runtime, all users are identified by their user name and password.

Migration of the ProTool password level

During conversion of a ProTool project to a WinCC flexible project, the program creates up to nine user groups. One user group is created for each password level used in the ProTool project.

The "Administrators" user group is always created. All user authorizations for which password level 9 was valid in ProTool are assigned to this "Administrator" group. All other user authorizations that were assigned the same password level in ProTool are also assigned to the same user group in the WinCC flexible project. The user groups are assigned the authorizations as in the password levels of the ProTool project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Advanced options in WinCC flexible

In WinCC flexible, users may be assigned to any user group, without making any allowances for hierarchical structures. During conversion, check the project to see whether to create new user groups or not.

2.5.1 Adapting user authorizations

User administration in WinCC flexible

During migration from a ProTool project to WinCC flexible, the program creates user groups and assigns those the authorizations according to the password level settings in the ProTool project.

In addition, the migration program assigns the various control elements in WinCC flexible the same access protection as in ProTool.

The ProTool password list will be excluded from migration, because it is stored on the relevant HMI, rather than in the project data.

However, you can reuse the ProTool password lists by with the help of the export and import functions.

Export the ProTool password lists of the ProTool project to a text file using the export_import_passwords function. Migrate the ProTool project. Import the *.txt file to the migrated project using the "ExportImportUserAdministration" function.

After migration, the users must be created again. WinCC flexible provides two options in this context:

- You configure new users, user groups and passwords with the help of the Runtime user administration tool.

- Configure a screen which contains a user view. Only members of the "Administrator" user group are granted full access to the user view in Runtime. The user view shows existing users, passwords, and user groups to an administrator. In addition, administrators may also create new users, user groups and passwords with the help of the user view.

Advanced options of WinCC flexible

WinCC flexible supports the user-specific assignment of users to a user group, irrespective of any hierarchical structures. During conversion, check the project to see whether to create new user groups or not. In addition, you can assign meaningful names to the user groups created during migration by selecting "User administration runtime" in the "Groups" editor.

2.6 Timer

Converting timers to jobs

During their migration to WinCC flexible, the "Alarms" of a ProTool project will be converted to jobs which are controlled by time events.

During conversion, the ProTool alarm times are converted to a time event. The function list assigned to the alarm will be entered in the function list of the event.

In ProTool projects, you can configure a date/time field which you can be used to define the start time of a job in Runtime. If such a date/time field exists in a ProTool project, the migration program automatically generates a relevant field and the associated tag in the WinCC flexible project.

In addition to time events, WinCC flexible also lets you configure the initialization of jobs by system events, such as "Alarm buffer overflow" or "Change Screen." In the course of conversion, you should decide whether to initialize jobs by system events, rather than by time events.

Migration with change of the HMI

<div align="right">3</div>

3.1 HMI settings

3.1.1 Global settings on the HMI device

Introduction

Many global device settings in ProTool projects for graphic devices are defined in the "PLC" menu in the engineering phase. When programming a WinCC flexible project, certain settings are not defined in the project, but rather directly in the Control Panel (on Windows CE devices) of the HMI. Certain settings are also replaced with standard Windows functions.

You cannot implement those global settings if migration involves a change of the HMI. Instead, the settings must be made directly at the HMI during commissioning or in Runtime.

HMI settings

ProTool settings which must be replaced by settings made directly at the HMI:

- "PLC > "Settings", "Date/time" group

  The date and time format of WinCC flexible HMI devices is set in the "Regional Settings" tab of the "Control Panel."

- "PLC system > "Settings", "Recipes" group

  The parameter set is not supported in WinCC flexible.

- "PLC" > "Screens/keys", "dynamic position" check box (only in ProTool projects for OP 35/37)

  This option is not required in WinCC flexible, because the WinCC flexible project runs on a standard Windows interface. On touch screen devices with a display size of more than 10", and on keyboard devices with mouse, you can move and resize the WinCC flexible user interface as in Windows.

- "PLC" > "Printer"

  For WinCC flexible HMI devices, configure the printer in the "Printers" tab of the "Control Panel."

- "PLC > "Alarms" > "Settings," group: "Overflow warnings"

  WinCC flexible no longer supports buffer overflow warning or logging. Instead, assign a job to the "Alarm buffer overflow" event using the following system functions:

  – "DisplaySystemAlarm", with the relevant alarm text as a parameter in order to output a user-defined alarm.

  – "PrintLog" to print out an alarm log with the contents of the alarm buffer.

3.1.2 Set HMI device with the control panel

Control Panel instead of the PLC menu

Many of the settings of ProTool projects for graphic devices are defined in the "PLC" menu.

Those settings can not be imported when you convert to a WinCC flexible project and change the HMI. Instead, use the Control Panel on the HMI to make those settings in the course of commissioning or in Runtime.

Calling the Control Panel to make settings

1. Open the Control Panel:

   In the startup phase, press the "Control Panel" button in the start menu.

   Procedure when the system is in operation:

   – If your configuration contains a control element which is logically linked to the "OpenControlPanel" system function, press it.

   – Otherwise: Press <Ctrl + Esc> to open the Windows CE start menu. Select the "Settings" command, then open the "Control Panel."

2. Click the appropriate option to make your the settings:

   "Regional Settings" for country-specific settings, such as the date and time format

   "Date/Time" to set the date and time of the HMI

   "Printers" to configure printers

3. To set the date and time, select the the "Regions" dialog box, select the relevant time zone from the "Regions" tab, and then press "Apply."

   Set the date and time on the "Date" and "Time" tabs, and close the dialog box with "OK."

4. To configure printers, make the relevant settings in the "Printers" dialog box.

3.2 Screens

3.2.1 Screen objects

Converting screen objects

Screen objects are adapted as far as possible to the requirements and options of WinCC flexible during conversion. A brief visual check is useful after a conversion.

Screen objects requiring more extensive changes during conversion:

- Text fields
- IO fields and symbol lists
- Transfer date/time display
- Graphic and semigraphic
- Buttons
- Bars and trend graphs

Editing

Editing is required in the following cases.

- ProTool project with date/time output:

  WinCC flexible provides the "Date/Time" screen object to edit this function.

- ProTool project with semigraphic:

  Replace the serigraphic objects with vector graphics in WinCC flexible. Vector graphics may be assigned dynamic properties.

  Various HMI devices, such as. TP 170micro, TP 170A and OP 77B, do not support vector graphics. For such devices, replace the semi graphic objects with bitmap graphic objects.

- ProTool project contains very small symbolic IO fields:

  During conversion, those symbolic IO fields are replaced by symbolic IO fields of defined minimum size. This may result in space problems in certain screens and thus require editing.

3.2.2 Fonts

Introduction

Windows-based systems provide more fonts and font sizes than graphic devices.

Converting default fonts

During a conversion involving a change of the HMI, all text fonts of all languages are initially converted to the following font types:

- "Tahoma", when changing to devices of the 170 series.
- "Courier New", when changing to one of the other Windows-based devices..

Adaptation of the font in WinCC flexible

When you completed the conversion, you may adapt the fonts as shown below:

- To set the "default font" in the languages and fonts editor, select "Device settings" > "Languages and fonts" from the project window. The default font is used, for example, for the alarm display on the HMI.
- To set the font for texts in screen objects, configure the relevant object in the object properties window or by using the toolbar. The toolbar lets you change the fonts of all highlighted objects in a single step by means of multiple selection.

3.2.3 Text fields

Converting text fields

Changes made during the conversion of text fields:

- The fields will be resized by adding two pixels per left and right side. If the screen is indented accordingly, the field will be located at the edge of the screen.
- The text in the fields will be centered vertically and left aligned horizontally.
- The following fonts will be adjusted:
  - On devices of the 170 series: Tahoma
  - On all other systems: Courier New

Note

WinCC flexible does not support text fields in default font with gaps which are not padded with space characters. In WinCC flexible, a line break will be inserted for each gap in the text.

Conversion of the font size

The font size is converted as follows:

ProTool graphic device (in pixels)	Courier New (in pts)	Tahoma (in pts)
8x8	6	5
8x16	10	10
16x16	10	10
16x24	16	15
24x24	16	15
24x32	22	20
32x32	22	20
32x48	34	30
48x48	34	34

3.2.4 Input/output fields and symbol lists

Conversion of IO fields and symbol lists

Adaptations of O fields and symbol lists during conversion:

- The fields will be resized by adding two pixels per left and right side. If the screen is indented accordingly, the field will be located at the edge of the screen.

- The text in the fields will be centered vertically and aligned horizontally, as in the ProTool project.

- The following font will be set:

  – On devices of the 170 series: Tahoma

  – On all other systems: Courier New

- The font size will be converted as in the text fields.

- Decimal places of tags in the ProTool project will be migrated to the WinCC flexible project in the form of an IO field attribute.

- If any illegal values appear in a text list or a graphics list, ProTool graphics devices show nothing in the graphic list. In WinCC flexible outputs a "###" string in the symbolic IO field, and a cactus image in the graphic IO field.

Conversion of IO fields

ProTool for graphic devices supports the configuration of IO fields with different properties. IO fields will be converted to different WinCC flexible screen objects, based on the properties configured in ProTool.

IO field on the ProTool graphic device with the format	Screen object in WinCC flexible
dec, hexadec, binary, string	[a1] IO field
Text symbol	[▼] Symbolic IO field
Graphic symbol	[▲] Graphic IO field

Editing

The "IO field" screen object used in WinCC flexible represents a standard Windows object of a defined minimum size. If a ProTool project contains very small IO fields, the automatically inserted IO fields may overlap or cover other screen objects after the conversion of your project.

In this case, adapt the relevant screen layout manually. It will usually suffice to simply move the screen objects.

3.2.5 Converting date/time display

"Date/Time" screen object

In WinCC flexible, the system time is output using the "date/time field" screen object. If a tag with a "date input/output" or "time input/output" system function was assigned to an IO field in the ProTool project to output the system time, these must be replaced with a "Date-time field" screen object when the conversion is completed.

Requirements

The ProTool project has been converted.

The screen with the IO field screen object for displaying the date and time is opened.

The properties dialog box is open.

Replacing the date/time display tag with a screen object

Note

TP 170A does not support the "Date-time field" screen object. Because the TP 170A does not have a buffered system clock, the PLC system time must be shown instead. Provide this function by assigning a date/time tag to an IO field.

---

1. Delete the IO field that was used to output the date and time.

2. $\boxed{\text{aI}}$ In the toolbox, select the "Date/time field" screen object from the "Simple graphics" group.

3. Insert the screen object at the relevant screen position. The property view shows the properties of the screen object.

4. Click "General" in the property view to define the basic properties of the object.

5. Select "Mode" to define the function principle of the "Date/time field" object.

   – "Output": The field outputs the current system time and date

   – "Input/Output": The user can also set the system time and date in Runtime using the date/time field.

6. Define the scope of the "Date/time field" display:

   – "Display date": The system date will be output.

   – "Display time": The system time will be output

   – "Date/time long format": The date and weekday will be output.

Those settings, made in "General," are sufficient to configure the date/time field. You can also set advanced properties in the property view.

3.2.6   Buttons

Touch panel buttons

WinCC flexible and ProTool use different libraries for the configuration of touch panel buttons. Allowances are made for this fact by converting those buttons f the ProTool project to corresponding WinCC flexible screen objects.

Conversion

Buttons concerned in particular:

ProTool project	WinCC flexible project
Button of the "visible" field type	▪ Button
Button of the "hotspot" field type	▪ Button, configuration as hotspot

ProTool project	WinCC flexible project
Set/reset bit	Button with corresponding system function of the "Bit processing" group
ActivateScreen	Button assigned the "ActivateScreen" system function
Signal lamp	Graphic IO field, configuration: Output field

Button frames

Buttons in WinCC flexible differ from those in the ProTool project by the standard Windows 3D frames.

Editing the labels

The configuration program automatically inserts line breaks when you configure labeled buttons on a graphic device in ProTool. Those automatic line breaks will be excluded from the conversion. After conversion, check the button labels and add line breaks as required.

3.2.7 Graphic and semigraphic

Converting bitmap graphics

All bitmap graphics will be included in the conversion. The graphics will adapted to the resolution or display size of the WinCC flexible HMI as required. The graphic objects will not be smoothed in the process.

Editing serigraphic objects

Semi graphic characters are no longer supported on WinCC flexible HMIs. After conversion, you thus have to replace those objects with vector graphic objects in the "Screens" editor. Tools available for this replacement:

Line

Rectangle

Circle

Ellipse

Polyline

Polygon

Note

Certain HMI devices do not support vector graphic objects (TP 170micro, TP 170A and OP 77B, for example) . For such devices, replace the serigraphic objects with bitmap objects.

Assigning dynamic attributes to graphic objects

WinCC flexible supports dynamic attributes for screen objects, for example, color change or flashing. This can be used in Runtime to draw the operator's attention to specific plant states.

To assign dynamic attributes to a screen object, select "Animations" from its properties view.

3.2.8 Bar

Converting bars

Bar properties included in a migration involving a change of the HMI:

- Tag (all properties, save for the decimal places)
- Direction
- Bar size (min./max. value)
- Display (frame and limit line)
- Bar color
- Axis display (scaling, axis label / position)
- Name

Differences between the ProTool and WinCC flexible projects after conversion:

ProTool	WinCC flexible
Frame color = scale color	Frame color = black
Places for the axis label will be set automatically	User-defined places for the axis label
If more places are required for the axis label, the object will be resized to provide sufficient space, while the size of the bar always remains constant.	If more places are required for the axis labels, the bar will be visualized in a narrower format. The overall size of the object always remains constant.
Bar tags may be assigned floating numbers.	Existing decimal places in the ProTool project will be discarded after conversion.

Editing bars

Because most of the bar properties are included in the conversion, your project requires only a small amount of editing. Editing is only required at the following points:

- Scale labels

  Define the number of places for the scale label in the "Scale" section of the property view.

- Object size

  Check the size of the bar after conversion. Adjust its size as required.

3.2.9 Trends

Converting trend graphics

WinCC flexible does not support multiplex trends. Those trends are thus excluded from a migration involving a change of the HMI.

Most of the properties of all other trend graphics will be included in a migration with change of the HMI. Resultant differences between a ProTool project for graphic devices and a project for WinCC flexible HMIs:

Visualization on the ProTool graphic device	Visualization on the WinCC flexible HMI
"Background" ("Properties" > "Color" tab) does not contains the scale background	"Background" contains the scale background
The scaling is set automatically	The scaling is configurable (properties view, "axis" section)
Places for axis labels are set automatically	Places for axis labels are configurable (properties view section "Right value axis" and "Left value axis"
Trend tag can be assigned decimal places.	Existing decimal places in the ProTool project are discarded after conversion
Profile trend is drawn from right to left	Profile trend is drawn from left to right
Profile trend is drawn from left to right	Profile trend is drawn from right to left

Editing trend graphics

Because most of the properties of trend graphics are imported during conversion, your project virtually requires no editing. You need to intervene at the following points:

- Scaling

  Define the scaling in the in the "Axes" section of the properties view.

- Axis label

  Define the number of places for the scale labeling in the "Right value axis" and "Left value axis" section of the property view.

- Drawing direction of profile trends

  Change the trend direction in the "x-axis" section in the properties view.

3.2.10 Default screens

Default screens are removed

WinCC flexible does not provide any ProTool default screens. Any default screens of the ProTool project will be cleared, and then removed alongside with the associated objects (such as tags) during conversion. However, you can only delete objects which are not referenced anywhere else in the project.

Note

During conversion, Migration detects default screens and their associated objects by the initial Z_ string in the object name. The migration program might delete any user-specific objects of the ProTool project with initial Z_ in their name string.

Editing

You can now edit the converted screens in the WinCC flexible project, and delete any remaining default objects you no longer require.

Any ProAgent default screens you integrated in the source project can now be replaced by the ProAgent default screens for WinCC flexible.

Using screen templates of the "Project Wizard"

You can use the default screen templates of the "Project wizard" when you create a new project. Those screens support typical project tasks, such as making system settings in Runtime, or for user administration. The Project Wizard only generates system screens with device class 270 10" or higher.

To use those screen templates in a converted project:

1. Create a new project for the same HMI as the converted project using the "Project wizard." Configure the "Project wizard" to create the required screens.

2. Open the converted project, and the project which contains the screen templates, in separate instances of WinCC flexible.

3. Paste the screen templates to the converted project using the clipboard function.

4. In the "Screen navigation" editor, create a logical link of each screen template to the previously created project screens.

   The screen templates are equipped with a navigation bar. Select any screens for which you do not require this bar in "Screen navigation." Disable the "Display navigation bar" option in the properties view of the screen.

3.2.11 Replacing standard ProTool screens

Introduction

WinCC flexible usually provides screen object templates you can use to map any system functions of default screens in the ProTool project to WinCC flexible. For information on the integration of those objects in a WinCC flexible screen and corresponding settings, refer to the relevant key word in the WinCC flexible Information System.

Editing

The section below provides overview of the default screen functions you can replace with WinCC flexible objects:

Default screen in the ProTool project	Task	Replacement in WinCC flexible
Z_BACKUP	Data backup Restoring backup data	On Windows CE devices, make those settings in the Control Panel by selecting the Backup/Restore option
Z_FORCE	Force tag	Screen object "Status/Force"
Z_STATUS_INP	Only for touch panels	No longer required for WinCC flexible HMIs
Z_MESSAGES	Editing alarms Printing alarms	"Alarm view" screen object To print: use reports
Z_MARCHIVE	Settings for alarm output	To print: use reports
Z_PASSWORD	Assigning and changing password Logon	Administration of passwords: "Password list" screen object The "Login dialog" is used to log on. Assign the system function "ShowLogonDialog" to a button you can use to open this dialog in Runtime. The dialog box is opened automatically when the user operates a control element for which he lacks the required authorization.
Z_PRINTER	Configuring printers and printer parameters.	On Windows CE devices, make those settings in the Control Panel by selecting the "Printer" option
Z_HARDCOPY	For graphic devices: Setting parameters for hardcopy (ASCII/graphic)	No longer required in WinCC flexible
Z_COLOR	For graphic devices: Disable specific colors, or modify the color assignments for printing	No longer required in WinCC flexible
Z_RECORD 1	Basic functions for using data records	"Recipe view" screen object
Z_RECORD 2	Advanced system functions for using data records	No longer required in WinCC flexible
Z_SETTINGS	General system settings	"Date-time field" screen object for the input of the date and time "Button" screen object with "SetLanguage" system function for toggling languages. "Button" screen object with "SetOperatingModeDevice" system function used to change to "online," "offline" or "transfer" mode.
Z_STATUS	Status tag for PLC diagnosis	"Status/Force" screen object
Z_SYSTEM_MEN	Overview screen for default screens	No longer required in WinCC flexible

Deleting remaining default objects

Only the objects which are not referenced anywhere else in the project can be deleted automatically during conversion. After the conversion, check whether any of the remaining references are still required. Delete all objects you no longer required manually, or replace those with references to other objects. ProTool default objects can be identified by the initial Z_ in their name string.

The cross-reference function supports you when searching for references:

1. Select the object from the Project View or Object View.

2. Select "Cross-reference" from the shortcut menu.

   The "Cross-reference" editor opens, showing the objects used.

3. Select the required cross-reference.

4. Select "GoToUsage" from the shortcut menu.

   This opens the editor in which the usage point is referenced.

5. Delete the reference, or replace it with a cross-reference to a different object.

3.2.12   Replacing standard ProAgent screens

ProAgent diagnostic screens for WinCC flexible

ProAgent is supplied with a default project you can use for all HMIs.

You can integrate the screens of this default project in a converted WinCC flexible project, and thus replace the diagnosis screens of the ProTool project. The table below shows the allocation of diagnostics screens of the ProTool projects to the WinCC flexible projects.

ProTool project	WinCC flexible project
ZP_PROAGENT	ZP_PROAGENT
ZP_ALARM	ZP_ALARM
ZP_UNITS	ZP_UNIT
ZP_UNITS_S7G	ZP_UNIT
ZP_DETAILAWL	ZP_DETAIL
ZP_DETAILLAD	ZP_DETAIL
ZP_DETAILSIG	ZP_DETAIL
ZP_STEP	ZP_STEP
ZP_MOTION	ZP_MOTION

Replacing the ProAgent default screens

In order to enable the interaction of ProAgent default screens with user-defined screens, integrate the ProAgent default screens in your PLC after you converted the project:

1. Open the converted project. The ProAgent default screens are generally empty after the conversion.

2. Rename the following screens:

   "ZP_DETAILSIG" to "ZP_DETAIL"

   "ZP_UNITS" to "ZP_UNIT"

3. In SIMATIC Manager, open the included STEP7 default project by selecting "Default/ProAgent" from the WinCC flexible folder.

4. Copy all default diagnostics screens of the default project to the converted WinCC flexible project using the clipboard function.

   All old ProAgent screens will be replaced by the screens of the same name of the default project.

5. Delete the ZP_DETAILAWL, ZP_DETAILLAD and ZP_UNITS_S7G screens. Those screens are no longer used.

3.2.13 Function keys

Converting softkeys

During conversion, the key assignment of the source project is transferred to the PLC project. Key assignments:

- The keys on the left and right side of the display are integrated successively, starting at the top left.

- The keys below the display are integrated successively from left to right.

- Any K keys of the PLC are integrated successively, starting at K1.

- The "LED mapping" area pointer will be replaced by an external array tag of the same name during conversion of a ProTool project to a WinCC flexible project. A 16-bit array element will be created for each area pointer. The LEDs on the HMI are assigned to the array elements in the same order as in ProTool. The PLC program therefore does not require any changes.

Conversion of the various keys

The assignment of the various keys, and of the tag for LED mapping, of the ProTool project to those of the WinCC flexible project depends on the HMIs used. A 1:1 assignment is not always possible, particularly if the source device and the destination device have a different number of keys.

Note

A bit that was set in the LED area of a softkey will only be only transferred to the tag corresponding to the LED mapping during migration if a function was assigned to that softkey in ProTool.

Converting OP25, OP27 > OP 170B, OP 270 6"

Keys	"LED mapping" tag
F1...F14 > F1...F14	F1...F8 > F1...F8
K1...K10 > K1...K10	K1...K10 > K1...K10

Converting OP25, OP27 > MP 270, MP 270B, OP 270 10"

Keys	"LED mapping" tag
F1...F8 > F1...F8 F9...F14 > F13...F18	F1...F8 > F1...F8
K1...K10 > K1...K10	K1...K10 > K1...K10

Converting OP25, OP27 > MP 370, Panel PCs

Keys	"LED mapping" tag
F1, F3, F5, F7 > S1...S4 F2, F4, F6, F8 > S9...12 F9...F12 > F1...F6	F1, F3, F5, F7 > S1...S4 F2, F4, F6, F8 > S9...12
Excluded are: K1...K10	Excluded are: K1...K10

Converting OP35, OP37 > OP 170B, OP 270 6"

Keys	"LED mapping" tag
F1...F8 > F1...F8 F13...F18 > F9...F14	F1...F8 > F1...F8
K1...K10 > K1...K10	K1...K10 > K1...K10
Excluded are: F9...F12, F19, F20, K11...K16	Excluded are: F9...F12, F19, F20, K11...K16

Converting OP25, OP27 > MP 270, MP 270B, OP 270 10"

Keys	"LED mapping" tag
F1...F20 > F1...F20	F1...F12 > F1...F12
K1...K16 > K1...K16	K1...K16 > K1...K16

Converting OP35, OP37 > MP 370, Panel PCs

Keys	"LED mapping" tag
F1, F3, F5, F7, F9, F11 > S1...S6 F2, F4, F6, F8, F10, F12 > S9...14 F13...F20 > F1...F8	F1, F3, F5, F7, F9, F11 > S1...S6 F2, F4, F6, F8, F10, F12 > S9...14
Excluded are: K1...K16	Excluded are: K1...K16

Converting pictograms

The pictograms used to describe the keys are adapted to the size of the display of the new HMI. Sizes derived:

ProTool	Pictogram size
OP25, OP27, 6" display	53 x 38
OP35, OP37, 10" display	80 x 58

WinCC flexible	Pictogram size
OP 170B, 6" display	53 x 38
OP 270, 6" display	53 x 38
OP 270, 10" display	80 x 58
MP 270, 10" display	80 x 58
MP 270B, 10" display	80 x 58
PC 670, 10" display	62 x 42
MP 370, 12" display	80 x 56
PC 670, 12" display	78 x 53
PC 870, 12" display	78 x 53
PC 670, 15" display	100 x 69
PC 870, 15" display	100 x 69

Automatic replacement of the "Function keyboard" and "System keyboard" area pointers

The ProTool "Function keyboard" and "System keyboard" area pointers are automatically replaced by an array tag of the same name during migration of a ProTool project to a WinCC flexible project. The length of the tags is is proportional to the number of softkeys or system keys, i.e. the tag is assigned one bit per key.

To evaluate key operation in the PLC, assign the system function "SetBitWhileKeyPressed" to the "Press" event to all softkeys and system keys. When the user presses a key in Runtime, the system sets the corresponding bit in the external tag. This returns the PLC the information that the key is pressed. The external tags are written to the same PLC address as the area pointers in the ProTool project. Hence, the PLC program does not require any changes.

Editing

Check the layout of any graphic objects or text fields used to label the softkeys in the ProTool project on the display of the WinCC flexible HMI.

Change the global assignment of softkeys by editing the project template in the Screens editor as required.

Note

Local function keys always remain active in Runtime, regardless whether the screen in which they are assigned is overlaid by an alarm view or an alarm window. This is a situation may develop in particular on HMIs with a small display, such as OP 270 6."

3.2.14   Evaluating key operation

Replacement of the "system keyboard" and "function keyboard" area pointers

The "System keyboard" and "Function keyboard" area pointers are not used in WinCC flexible.

To evaluate the active/inactive states of a system key or a softkey in the PLC program, assign the "SetBitWhileKeyPressed" system function to the "Press" event of all keys. When the user presses a key in Runtime, the system sets a corresponding bit in an external tag. This returns the PLC the information that the key is pressed.

This is only supported for the ACK, ESC, ENTER and HELP keys system keys.

To avoid having to adapt the PLC program, use the system function to set the same bit that was set in the ProTool project in the system keyboard image or in the function keyboard image.

Requirements

The ProTool project is converted to a WinCC flexible project and is opened.

The "Properties" dialog box is open.

Procedure

1. In the Tag Editor, create an external array tag for the function keys, and another for the system keys.

2. In each tag, create an "Array elements" for each softkey or system key of the HMI.

3. in the PLC, enter the tag addresses which correspond with the area pointer addresses in the ProTool project.

4. Open the project template in the Screens editor.

5. Select the key that you want to assign. The properties of the key are shown in the properties dialog box.

6. Click "Press" in the "Events" group.

7. Assign the "SetBitWhileKeyPressed" system function to the event

8. As parameter, define the array tag that accesses the same bit that was assigned to the key in the ProTool project.

   Bits in the system keyboard image assigned to the system keys:

   ACK (acknowledgment key): 1. Word bit 1

   ESC: 1. Word bit 3

   ENTER: 1. Word bit 6

   HELP: 1. Word bit 0

9. Repeat steps 5 to 8 for all softkeys and system keys.

Result

When one of the system keys or softkeys is pressed in Runtime, a corresponding array element reports the active state of the key to the PLC.

Note

If you want to evaluate in the PLC whether a softkey was pressed in a specific screen, assign the "SetBitWhileKeyPressed" system function to the relevant softkey locally in this screen. This ensures that the PLC only evaluates the key in runtime when this screen is open.

If a softkey is assigned a screen-specific (local) system functions, the screen-independent (global) assignment of the softkey will be disabled.

3.3 Tags and area pointers

3.3.1 Tags

Introduction

The migration program creates tags corresponding to the tags in the ProTool project in the WinCC flexible project. In addition, it generates an interconnection to the PLC for the external tags if those do not yet exits.

Converting STRING tags

The "STRING" data type is converted to the "STRINGCHAR" data type for the conversion of a ProTool project for graphic devices to a WinCC flexible project. If the HMIs of the ProTool project and WinCC flexible project have different character sets, special characters might be interpreted differently. In WinCC flexible projects, the hexadecimal value 00H is interpreted as an identifier for the end of the string. The following string characters are not displayed.

Attribute

Attributes not supported in WinCC flexible:

- Indirect writing of tags

  The tag values can not be transferred to the PLC using the "Record" area pointer ("data mailbox" in the ProTool area pointer). For detailed information on project adaptation, refer to "Recipes."

- "Offline" switching of tags

  Single tags with a logical link to the PLC can not be temporarily decoupled from the PLC in WinCC flexible. However, all tags which are integrated in the same PLC can be temporarily decoupled from the HMI, for example, during commissioning. This is done using the "ChangeConnection" system function.

Importing decimal places

WinCC flexible does not support the assignment of decimal places to a new tag. Tags that have decimal places in the ProTool project are converted as follows:

- Tags asigned to an IO field retain the decimal places.
- Recipes apply the decimal places of tags in the recipe view.
- The tags for bars, trends and alarms do not have decimal places.

Converting trend tags

The Migration tool converts profile trends of a ProTool project to trends of the type "Buffer bit-triggered." During migration, the program always creates array tags for the trend tags configured in ProTool and writes these to the WinCC flexible project, in order to buffer the tag values ("Buffer" and "Switch buffer".)

If you have used the trend tags at any further positions in the ProTool project (for example, to display single values), you must check whether the use of an array tag is supported at these positions in the WinCC flexible project after you migrated the project.

Different handling of the start value

In WinCC flexible, the start value not only is written to the tag during the initial startup of the HMI, but rather at every startup. Tag values are not buffered, i.e. the last value can not be written to the tag on startup.

In WinCC flexible, tags are only initialized during startup of the HMI. An initialization at a later time is only possible indirectly using a user script. The user script evaluates the delta value of the tag which contains the start bit: When the HMI is connected to power, the value of the start bit changes from 0 to 1. The value of the tag that can be initialized with the start value, depending on the delta value of the start bit.

Example of polling the start bit in an S7 PLC program

In the PLC program, the "AND NOT" tatus of a start bit can be polled as follows:

- PLC program with symbolism:

```
UNStart bitSFlag bit//set after startup
```

- PLC program without symbolism, coordination range starts at DB10.DBW0:

```
UNDB10.DBX1.0SM2.0//set after startup
```

Note

In WinCC flexible, the tag values are not buffered at system restart. If the value is written from the PLC to the tag after the connection to the PLC is up and the start value is thus overwritten, the tag has a control link.

3.3.2 Area pointer

Introduction

The following area pointers, save for the change of name, are migrated 1:1 during migration with change of the HMI:

Area pointers in ProTool	Area pointers in WinCC flexible
User version	Project ID
Screen number	Screen number

Area pointers which are replaced by array tags:

- Event alarms
- Error alarms
- LED mapping
- Trend request 1
- Trend request 2

Structural changes are made to the remaining area pointers.

Modified area pointers

The following area pointers are modified during migration with a change of HMI device:

Area pointers in ProTool	Area pointers in WinCC flexible
Event alarms	Area pointer discarded. It is replaced during conversion by an external array tag with a 16-bit array element and a corresponding name. Following conversion, there is one array element of the newly created array tags for every bit of the ProTool area pointer. The alarms are assigned to the bit in increasing order of alarm numbers. The array elements automatically receive the update identifiers 1-8.
Data mailbox	Data mailbox n WinCC flexible, this area pointer always has a length of 5 words. Data words 6 to n are not required in WinCC flexible, because data sets records not be transferred indirectly to the PLC indirectly via the data mailbox. This area pointer is automatically converted during conversion. The length of the data mailbox remains the same when connected with an S7 PLC; the structure changes as follows: 1. Word: Recipe number (unchanged) 2. Word: Record number (previously 4th word) 3. Word: Reserved 4. Word: Record acknowledgment (previously byte n+3 of the interface range) 5. Word: Free
Function keyboard	Area pointer discarded Substitute: Configure an external tag with a length exactly the same as the number of softkeys so the tag has one bit for each key. Configure the "SetBitWhileKeyPressed" system function to the previously assigned function key. Set the parameters of the system function so the key is set to the corresponding bit of the external tag. The status of the tags can then be evaluated in the PLC program.
Trend request	Area pointer discarded
Trend transfer 1	Area pointer discarded Substitute: An identically named external array tag with a 16-bit array element will be created, rather than the "trend transfer 1" area pointer. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. After the PLC program has set the array element and the trend group array element assigned to the trend, the HMI can detects the trigger event and reads a value or the entire buffer, depending on the configuration.
Trend transfer 2	Area pointer discarded Substitute: An identically named external array tag with a 16-bit array element will be created, rather than the "trend transfer 2" area pointer. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. After the PLC program has set the array element and the trend group array element assigned to the trend, the HMI can detects the trigger event and reads a value or the entire buffer, depending on the configuration.

Area pointers in ProTool	Area pointers in WinCC flexible
LED mapping	Area pointer discarded. It is replaced during conversion by an external array tag with a 16-bit array element and a corresponding name. Following conversion, there is one array element of the newly created array tags for every bit of the ProTool area pointer. The LEDs on the HMI device are assigned to the bit numbers in the same order as in ProTool. The array elements automatically receive the update identifiers 17-24.
OP acknowledgment	Area pointer discarded It is replaced during conversion by an external array tag with a 16-bit array element and a corresponding name. Following conversion, there is one array element of the newly created array tags for every bit of the ProTool area pointer.
PLC acknowledgment	Area pointer discarded Substitute: The array tag for error alarms is extended during migration. This provides an additional bit for the acknowledgment of error alarms. Example of the assignment of the array tag: Bit 0-15=error alarms, Bit 16-31= acknowledgment of error alarms.
Interface area	The data are exchanged with the PLC via the following area pointers: • "Coordination" • "Job mailbox" • "Date/time" "Conversion" Once migration is complete, you must create the area pointer in the project. This distribution simplifies data exchange between HMI device and PLC. The PLC program must be configured accordingly after conversion.
Error alarms	Area pointer discarded. The area pointer is replaced during conversion by an external array tag with a 16-bit array element and a corresponding name. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers. The array elements automatically receive the update identifiers 9-16. As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." Vice versa, the HMI interprets the alarm as "outgoing" when the bit in the array element is reset to "FALSE" on the PLC.
System keyboard	Area pointer discarded Substitute: Configure an external tag with a length exactly the same as the number of softkeys so the tag has one bit for each key. Assign the "SetBitWhileKeyPressed" system function to the previously assigned system key. Set the parameters of the system function so bit of the external tag which corresponds with the tag will be set. The tag status can then be evaluated in the PLC program. Note that system keys such as <ACK> or <Esc> can not be configured.

3.3.3 Area pointer "Interface area"

Introduction

During conversion to WinCC flexible, the ProTool "interface range" area pointer is automatically distributed to three area pointers:

- "Coordination"
- "PLC job"
- "Date/time"

When migration is completed, you must create those area pointers in the project.

This distribution simplifies data exchange between the HMI and the PLC. The PLC program must be adapted accordingly after conversion.

Distribution to S7 PLCs

The interface range is distributed as follows for S7 PLCs:

ProTool	WinCC flexible
Interface range n+0 to n+31	Coordination n+0 to n+1
	PLC job n+4 to n+11
	Date/time n+15 to n+26

Handling of the start bit (coordination area pointer)

The coordination area pointer contains the start bit. If the value of this bit is read out in the PLC program, you can see whether the HMI device has started up from the PLC. The start bit is temporarily set to "0" by the HMI device during the startup process. As soon as the boot process is complete, the start bit is set to "1".

Special features in SIMATIC S5

The interface DB in the ProTool project can be reused after migration.

Example:

Area pointer	AS of group 1	AS of group 2
Job mailbox	DW32 to DW35	DW32 to DW35
Date/time	DW42 to DW47	DW66 to DW71

For AS of group 2: WinCC flexible HMIs support only one job mailbox. The PLC job with its three parameters is written directly to this mailbox.

3.4 System functions and PLC jobs

3.4.1 System functions

Conversion

During migration with a change of HMI device, the ProTool system functions are replaced with corresponding WinCC flexible system functions.

In WinCC flexible, the names of the system functions are assigned according to a specific convention. For this reason, the names of the WinCC flexible system functions do not match the names of the corresponding ProTool system functions. The system functions groups in WinCC flexible also have different names from those in ProTool.

On certain some system functions, only the name has changed. Certain system functions are no longer required or can no longer be configured, due to the different functionality of the HMI devices. You may have to edit the WinCC flexible project and your PLC program to obtain the functionality of the ProTool project.

Renamed system functions

The following system functions do not require editing after conversion, because there is no difference between ProTool and WinCC flexible other than the name:

ProTool	WinCC flexible
Screens	Screens
Display selection	ActivateScreen
Dynamic screen selection 1	ActivateScreenByNumber
Dynamic screen selection 2	ActivateScreenByNumber
Screen selection previous	ActivatePreviousScreen
PROFIBUS screen number (TP only)	PROFIBUS screen number (TP only)

ProTool	WinCC flexible
Editing bits	Bit processing
Reset bit	ResetBit
Reset bit in word	ResetBitInTag
Set bit	SetBit
Set bit while key pressed	SetBitWhileKeyPressed
Set bit in word	SetBitInTag
Direct key (TP only)	Direct key (TP only)

ProTool	WinCC flexible
Print	**Print**
Hardcopy start/stop	PrintScreen

ProTool	WinCC flexible
Alarms	**Alarms**
Clear alarm buffer	The function "ClearAlarmBufferProTool" is used for migration. The parameters of this system function are identical to those in the ProTool function.
	Use the system function "ClearAlarmBuffer" only for new projects. This has new parameters.
Alarm report ON/OFF	SetAlarmReportMode
Clear error alarm buffer	The "ClearAlarmBufferProTool" function is used for migration. The parameters of this system function are identical to those in the ProTool function.
	Always use the system function "ClearAlarmBuffer" in new projects. This has new parameters.

ProTool	WinCC flexible
Calculate	
Tag: Value minus/plus	IncreaseValue
Tag: Set value	SetValue

ProTool	WinCC flexible
System settings	**System**
Display help (TP only)	ShowOperatorNotes
Cleaning screen (TP only)	EnableCleaningScreen (TP only)
Acoustic signal (TP only)	SetAcousticSignal (TP only)
Touch calibration (TP 37 only)	CalibrateTouchScreen (TP only)

ProTool	WinCC flexible
Keyboard	**Keyboard operation for screen objects**
Page up	ScreenObjectPageUp
Page down	ScreenObjectPageDown

ProTool	WinCC flexible
Toggle	**Settings**
Changing the operating mode	SetDeviceMode
Language change	SetLanguage

Note

System functions of the "Edit bits" group assigned to the global event "buffer overflow" in ProTool are excluded in a migration with change of HMI device.

Instead, assign a job to the "Alarm buffer overflow" event in WinCC flexible. Configure the desired system functions of the "Edit bits" group in the job function list.

Replacing system functions

In the following system functions, either the basic implementation of the function has changed or it can no longer be configured in WinCC flexible. The table shows possible workarounds that can be configured in an editing step in the WinCC flexible project. Additional information, such as step-by-step instructions, can be found in other sections of the WinCC flexible Information System.

ProTool	WinCC flexible	Explanation/replacement/workaround
Editing bits	Bit processing	
"Initialize bit at startup"	not required	In WinCC flexible, the start value not only is written to the tag during the initial startup of the HMI, but rather at every startup.
"Reset BOOL tag"	not supported	---
"Set BOOL tag"	not supported	---

ProTool	WinCC flexible	Explanation/replacement/workaround
Data set	Recipes	
"Select data record" "Edit data record" "DeleteDataRecord" "Data record: DAT > OP" "Data record: DAT > OP/PLC" "Data record: OP > DAT" "Data record: OP > PLC" "Data record: PLC > OP" "Data record: PLC > OP/DAT" "Format data medium"	System functions are differently mapped	The "Recipe display" screen object and new system functions for record transfer are available in WinCC flexible.

ProTool	WinCC flexible	Explanation/replacement/workaround
Date/time		
Date and time" "Date input/output" "Time input/output"	System functions are not required	WinCC flexible provides the "Date-time field" screen object. For this reason, the input and display of the date and time has been changed fundamentally.

ProTool	WinCC flexible	Explanation/replacement/workaround
Print	**Print**	
"Automatic graphic print"	not supported	Alarms can be output as follows in WinCC flexible: • During alarm reporting, the printer automatically detects • whether the alarms should be output in ASCII mode or graphic mode. • Alarms for projects in languages with non-ASCII characters can only be be output in a report which contains an alarm report.
"Graphic mode with print screen list"	not required	In WinCC flexible, screens and reports are always printed in graphic mode
"Print screen list"	not supported	Multiple screens can be output as follows in WinCC flexible: Create a report for all screens to which you want to copy all screen objects. You can not output objects which can not be copied, such as bars, to a printer. Configure an IO field so the user can select one of the screens from a text list. Configure a tag for the IO field to save the result of the selection. Assign the "PrintReport" system function to a button and set the tag value as parameter.
"Print alarms"	not required	Add the "Print alarm" object to a report. Enable the "Error alarms" and/or "Warning alarms" alarm classes on the "Alarms" tab. Print the report using the "PrintReport" system function.
"Print alarms with filter 1" "Print alarms with filter 2"	System functions are not supported	Replace those system functions with a report with the "Print report" object. WinCC flexible only supports alarm filtering by alarm type. All other alarms properties are read only and may be output to a printer (setting in properties view of the alarm report in the layout section). Print the report using the "PrintReport" system function. In Runtime, the alarm view can be sorted by this criterion by double-clicking a column header. If only time-limited content is printed, you can refer to the alarm log in WinCC flexible. There, your can output the report of a defined logging period to a printer.
"Printer parameters" "Printer parameter (anchor)color assignment for printer" "Hardcopy parameters" "Hardcopy parameters (anchor)"	System functions are not required	Make the appropriate settings in the Control Panel of the HMI (option: "Printers")

ProTool	WinCC flexible	Explanation/replacement/workaround
Alarms	Alarms	
"Screen for short alarm page" (TP only)	System function is not required	Behavior of the error alarm indicator in WinCC flexible Runtime:
		When the flashing error alarm indicator is touched, only the error alarm window is brought to the foreground. When the non-flashing error alarm indicator is touched, the error alarm page will open.
"Change OA page/buffer"	"ShowAlarmWindow"	In WinCC flexible, the warning alarm page and event buffer are implemented with differently configured screen objects of the "alarm window" in the template. The corresponding alarm window is displayed using the "ShowAlarmWindow" system function.
		Configuration of the alarm window for warning alarm page: Alarm class = "warning alarms;" alarms are displayed.
		Configuration of the alarm window for event buffer: Alarm class = "warning alarms;" alarm events are displayed.
		A switchable display of alarm buffer and alarm page can be implemented with the aid of a user script.
"Change SM page/buffer"	"ShowAlarmWindow"	In WinCC flexible, the warning alarm page and event buffer are implemented with differently configured screen objects of the "alarm window" type. The corresponding alarm window is displayed with the "ShowAlarmWindow" system function.
		Configuration of the alarm window for warning alarm page: Alarm class = "error alarms;" alarms are displayed.
		Configuration of the alarm window for error alarm page: Alarm class = "error alarms;" alarm events are displayed.
		A switchable display of alarm buffer and alarm page can be implemented with the aid of a user script.
"Warning alarm window"	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template.
		Configuration of the alarm window: Alarm class = "warning alarms;" alarms are displayed.

ProTool	WinCC flexible	Explanation/replacement/workaround
Alarms	**Alarms**	
"Display warning alarms" Parameter "display" = 0 = page	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Message class = "warning alarms;" alarms are displayed; the column display is more comprehensive that the alarm window that is converted from the warning alarm window.
"Display warning alarms" Parameter "display" = 1 = buffer	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Message class = "warning alarms;" alarm events are displayed
"Display alarms" Parameter "alarm type" = 0 = error alarms	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Alarm class = "error alarms;" alarms are displayed.
"Display alarms" Parameter "alarm type" = 1 = warning alarms	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Alarm class = "warning alarms;" alarms are displayed
"Display error alarms" Parameter "display" = 0 = page	"ShowAlarmWindow"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Message class = "error alarms;" alarms are displayed
"Display error alarms" Parameter "display" = 1 = buffer	"Show alarm window"	Replaced by the "ShowAlarmWindow" system function to display the "alarm window" screen object in the project template. Configuration of the alarm window: Alarm class = "error alarms;" alarm events are displayed.
"Operate alarm screen" (TP only)	System function is not required	In WinCC flexible, alarm window and alarm display have a scrollbar for scrolling. Help can be called with the "Help" button. A switchable display of alarm buffer and alarm page can be implemented with the aid of a user script.

ProTool	WinCC flexible	Explanation/replacement/workaround
Alarms	**Alarms**	
"Filter alarms"	System function is not supported	In WinCC flexible, alarms are displayed with the "alarm display" or "alarm window" screen object.
		The filter criteria can only be set in the properties view of the screen object in the "General" group. Filtering by "Priority," "Acknowledgment group" and"Status" is not supported.
		In Runtime the alarm display can be sorted by the criterion shown by double-clicking on a column header (only with the "Date" and "Time" columns).
"Buffer overflow warning"	System function is not supported	Configure a job to the "Alarm buffer overflow" event as a replacement for the "BufferOverflowWarning" system function. Configure the "ShowSystemAlarm" system function in the job function list. You can output your own alarm text with this system function. The alarm text is restricted to one language.
"First/last error alarms"	System function is not supported	In WinCC flexible, specify the sorting order for the error alarms when configuring the alarm display or the alarm window in the properties view.
		In Runtime the alarm display or the alarm window can be sorted by the criterion shown in the column by double-clicking on a column header (only with the "Date" and "Time" columns).

ProTool	WinCC flexible	Explanation/replacement/workaround
Password	**User Administration**	
"Define password" "Define password level" "Display passwords"	System functions are not required	Windows-based systems have the "Password display" screen object.
		This gives the administrator the option of managing user names and passwords of other users and assign user authorizations.
		In WinCC flexible, user names, passwords and user groups can be set during configuration and transferred to the HMI device with the project.

ProTool	WinCC flexible	Explanation/replacement/workaround
Password	**User Administration**	
"Input password"	"ShowLogonDialog"	Substitute: Configure the "ShowLogonDialog" system function to show the logon dialog for user logon.
		Most WinCC flexible objects can be given access protection in the properties view. In this case the objects can only be operated by members of specific user groups. If a protected operation control element is operated in runtime and the user does not belong to the required user group, a login window automatically opens so the user can log in.
		Use the "ExportImportUserAdministration" system function to transfer all passwords of a project to another project or another HMI device.

ProTool	WinCC flexible	Explanation/replacement/workaround
Calculate	**Calculation**	
"Set initial value of tag"	System function is not supported	The tags with WinCC flexible HMI devices are basically initialized every time the device is started. To write the start value of a tag to the PLC, evaluate the start bit in the "Coordination" data range and set an external tag to the desired value at the "Value change" event with the "SetValue" system function.
"Conversion linear 1" "Conversion linear 2"	Property "Linear scaling" in the properties view of the tags	The "Linear scaling" property is assigned appropriately in the properties view of the tags during conversion instead of the conversion function.
		Note:
		The "LinearScaling" and "InvertLinearScaling" system functions are based on the conversion of the value of a tag to the value of a second tag. Therefore it cannot be used here.
"Conversion square 1" "Conversion square 2"	System function is not supported	Use corresponding user scripts instead.

ProTool	WinCC flexible	Explanation/replacement/workaround
Status force tag		
"S5 Status force Init 1" "S5 Status force Init 2" "S5 control tag" "S5/S7 Status start/stop" "S5/S7 Status tag" "S5/S7 Status force deselection" "S5/S7 Status force selection" "S5/S7 Status input" "S5/S7 Status force hide" "S5/S7 Status force identification" "S7 Status force Init 1" "S7 control tag"	System functions are not required	Insert the screen object "Status force" into the screen that contains the system functions from the "Status force tag" group. All system functions of the "Status force tag" group can be replaced with this screen object. Note: The "Status force" screen object cannot be used for an S7 200 PLC.

ProTool	WinCC flexible	Explanation/replacement/workaround
System setting	**System**	
"Backup/Restore Module"	System function is not required	Use the Backup/Restore option in the Control Panel in Windows CE devices to transfer all data of a WinCC flexible HMI device to a memory card. As an alternative the "Backup/Restore" function of the ProSave service tool can be used. Use the "SaveRecord" and "LoadRecord" system functions to back up individual records to a memory card or to read them from a card.
"Darken screen"	System function is not required	The brightness of WinCC flexible HMI devices can be dimmed or a screen saver can be activated after a configurable period. Both settings can be made in the control panel with the "Screensaver" option.

ProTool	WinCC flexible	Explanation/replacement/workaround
System setting	System	
"Change contrast" (TP27)	This is replaced by the "AdjustContrast" system function on touch panels with STN display. This system function is not supported on touch panels with TFT display.	The contrast can be changed in runtime as follows on devices with foil keyboards: + cursor left: Reduce contrast + cursor right: Increase contrast

3.4.2 Configure switchable display of alarm buffer and alarm page

Sample configuration

A switchable display of the event buffer and warning alarm page can be configured with a procedure:

Requirements

An internal tag with the name "ChangePageBuffer" of the "Bool" type is created.

Two alarm windows with the following configuration are created in the template:

Name	Alarm class	The following are displayed:
Event page	Operation	Alarms
Event buffer	Operation	Alarm events

Procedure

1. Create a script with following content under the name "SwitchPageBuffer":

   If ChangePageBuffer = 0 Then

   ShowAlarmWindow(Eventbuffer, hmiOff) ShowAlarmWindow(Eventpage, hmiOn)

   SetBit(ChangePageBuffer)

   Else

   ShowAlarmWindow(Eventpage, hmiOff) ShowAlarmWindow(Eventbuffer, hmiOn) ResetBit(ChangePageBuffer)

   End if

2. Assign the procedure to a key or button.

Result

In runtime the user can switch between event buffer and warning alarm page with the key or button.

Switching between error alarm page and error alarm buffer

If you want to configure a switch between error alarm page and error alarm buffer, change the example as follows: Configure the alarm window to display "error alarms" as alarm class.

3.4.3 PLC jobs

Converting job mailboxes

Some of the PLC jobs can continue to be used in WinCC flexible. Because of the different functions of the HMI devices some PLC jobs are no longer required or can no longer be used. In some circumstances the WinCC flexible project and the PLC program must be edited to achieve the functionality of the ProTool project.

If the destination project requires editing in any case, check whether the tasks done with PLC jobs in the ProTool project can be done more efficiently with system functions in WinCC flexible.

Unmodified PLC jobs

Editing is not required after conversion with the following PLC jobs:

- 14 "Set time (BCD-coded)"
- 15 "Set date (BCD-coded)"
- 24 "Password logout"
- 41 "Transfer date/time to PLC"
- 51 "Screen selection"
- 49 "Clear event buffer"
- 60 "Clear error alarm buffer"

PLC jobs that must be edited

In the following PLC jobs either the basic implementation of the function has changed or it can no longer be used in WinCC flexible HMI devices. The table lists information on possible workarounds that can be imported to the destination project or the PLC program in an editing step.

ProTool	WinCC flexible	Explanation/replacement/workaround
2 "Screen switch-off"	PLC job is not required	The brightness of WinCC flexible HMI devices can be dimmed or a screen saver can be activated after a configurable period. Both settings can be made in the Control Panel with the "Screensaver" option.
3 "Hardcopy"	PLC job is not supported	Instead configure the "PrintScreen" system function to the "Value change" event of a tag with PLC connection.

ProTool	WinCC flexible	Explanation/replacement/workaround
4 "Control port/set relay"	PLC job is not required	There are no ports or relays on WinCC flexible HMI devices.
12 "Switch alarm logging on/off"	PLC job is not supported	Instead configure the "SetAlarmReportMode" system function to the "Value change" event of a tag with PLC connection.
13 "Switch language"	PLC job is not supported	Instead configure the "SetLanguage" system function to the "Value change" event of a tag with PLC connection.
21 "Error alarm display type"	PLC job is not supported	In WinCC flexible, alarms are displayed with the "alarm display" screen object or with the "Alarm window" screen object. The filter criteria for the alarms can only be set in the properties view of the screen object in the "General" group. Filtering by "Priority," "Acknowledgment group" and "Status" is not supported. With a touch panel, the alarm display in Runtime can be sorted by the criterion shown by double-clicking on a column header (only with the "Date" and "Time" columns).
23 Set password level	23 Set user group	The PLC job "23 Set user group" takes the changed concept of user administration in WinCC flexible into account because it sets the user group instead of the password level. If you use PLC job 23, the PLC program must be modified accordingly.
37 "Switch overflow warning for warning alarms on/off" 38 "Switch overflow warning for error alarms on/off"	PLC job is not required	There is no overflow warning buffer in WinCC flexible. Instead configure a job at the "Alarm buffer overflow" event. Configure the "ShowSystemAlarm" system function in the job function list. You can output your own alarm text with this system function. The alarm text is restricted to one language.
42 "Get LED range from the PLC" 43 "Get event message range from the PLC" 44 "Get alarm message range from the PLC" 45 "Get acknowledgment range from the PLC" 47 "transfer LED range directly to the HMI device"	PLC jobs are not supported	Instead use PLC job "46 update tag" with appropriate parameters. This PLC job reads the values of WinCC flexible tags that replace the ProTool area pointers "event messages," "alarm messages,", "PLC acknowledgment" and "LED mapping" after migration. Control which tag is read with the "Update identifier" parameter. Assign the update identifier of the desired tag to this parameter. Assignment of the parameter "update identifier": 1...8 warning alarms 9...16 error alarms 17...24 LED mapping

ProTool	WinCC flexible	Explanation/replacement/workaround
69 "Transfer recipe record from PLC to TP/OP"	69 "Read data record from PLC"	In WinCC flexible, "Recipe number" and "Record number" must be given as parameters.
		If you want to transfer the recipe name or record name from the PLC to the HMI device, use the "GetDataRecordFromPLC" system function instead of the PLC job 69.
70 "transfer recipe record from TP/OP to PLC"	70 "Write record to PLC"	In WinCC flexible, "Recipe number" and "Record number" must be given as parameters.
		If you want to transfer the recipe name or record name from the HMI device to the PLC, use the "WriteRecordTagToPLC" system function instead of the PLC job 70.
72 "Cursor positioning in current screen"	PLC job is not supported	Use PLC job 51 "Screen selection" instead.

3.4.4 Replacing job mailboxes with system functions

Introduction

Some PLC jobs that can no longer be used with WinCC flexible HMI devices can be replaced by system functions.

If the WinCC flexible project requires editing for any reason, check whether the tasks that were done with PLC jobs in the ProTool project can be done more efficiently with system functions in WinCC flexible.

Configuring

If you want the PLC to trigger the execution of a system function configured on the HMI device, configure a tag with the property "Always update" in WinCC flexible. Configure the corresponding system function for the "Change value" event on this tag.

Execution

If the value of the tag changes in the PLC, the system function is executed on the HMI device as soon as it detects the change.

Example: Trigger system function "SetAlarmReportMode" from the PLC

The following configuration can be used to replace the PLC job 12 "Switch alarm logging on/off" used in the ProTool project with a system function in the WinCC flexible project.

In WinCC flexible

1. Create an external tag of the INTEGER type.

2. Select "Tag > General" to set the "continuous cyclic" acquisition cycle.

3. Configure the "SetAlarmReportMode" system function to the event "Change value." Input the tag itself as parameter.

On the PLC

1. Set the value of the tag in the PLC program to 0 (alarm logging off) or 1 (alarm logging on).

Result

The alarm logging is switched on or off depending on the value of the tag.

3.5 Alarm system

3.5.1 Alarms

Converting alarms

The alarms are automatically converted during conversion of a ProTool project to a WinCC flexible project. However, some properties and functions are not supported in WinCC flexible. The project must be edited and other configuration solutions must be found for specific functions.

In details there are changes in the following areas:

- Assignment of alarm number
- Alarm range indicator
- Behavior with alarm buffer overflow
- Rest alarm
- Tag type of alarm tag
- Multi-line alarm texts without line break

Changing the alarm number

In ProTool projects the same alarm number can be assigned for an event messages and an alarm messages. In WinCC flexible, one alarm number can be assigned once only.

During migration all numbers of the ProTool error alarms are increased by the value of 2000. This prevents duplicate alarm numbers from being assigned.

Application example

A ProTool project contains 200 faults alarms numbered from 1 to 200. If this project is converted to a WinCC flexible project, the error alarms in the new project are given the numbers 2001 to 2200. An error alarm with the number 1 becomes the error alarm with the number 2001, etc.

Required changes

If the error alarm numbers are used for project control in the ProTool project, e.g. in user scripts, these points must be edited in the converted project.

Conversion of the alarm area pointers

During conversion of a ProTool project for graphic devices to a WinCC flexible project, correspondingly named external array tags are created to replace the "event messages" and "alarm messages" area pointers as follows. The "alarm message" array is extended for the "PLC acknowledgment" area pointer. A new array is created for the "OP acknowledgment" area pointer.

Alarm buffer overflow

The "Alarm buffer overflow" event is triggered when the alarm buffer is completely filled. In WinCC flexible, the overflow of the alarm buffer cannot be queried separately for system alarms and warning alarms. The "ClearAlarmBuffer" can however be used to delete alarms of specific classes from the alarm buffer and thus empty the alarm buffer when it overflows.

Create a job that is triggered by the "Alarm buffer overflow" event to respond to the buffer overflow. Input the following system functions into the job function list depending on the project requirements:

- "ShowSystemAlarm" with the desired alarm text as parameter to notify the user of the buffer overflow.

- "PrintReport" to print a report with alarm log; the alarm log is configured to print the contents of the alarm buffer.

- "ClearAlarmBuffer" to delete alarms of specific classes from the alarm buffer

Alarm tag of the type "Dec,Dec"

The tag type "Dec,Dec" does not exist in WinCC flexible. After conversion, an alarm tag of this type is treated as "decimal."

Rest alarm

The rest alarm is no longer supported.

Multi-line alarm texts without line break

After migration multi-line alarm texts are not always displayed in WinCC flexible. This is because the alarm text is captured as running text in ProTool and is now too long to be imported during migration. The log file contains the information that the alarm texts are too long. The alarm texts must be input into the migrated project again.

3.5.2 Configuring and printing a report with the contents of the buffer overflow

Introduction

In WinCC flexible, there is no buffer overflow warning or logging.

Instead a job can be created to display a system alarm on buffer overflow and to print a report with the contents of the buffer overflow.

Procedure

If "Logging on overflow" is configured in the ProTool project, this behavior is not automatically converted. The same behavior can be generated in WinCC flexible with the following configuration:

1. Create a report with the "Print alarm" object.

2. In the properties view of the alarm report select "Alarm events" as "Source for alarms."

3. In the scheduler create a job that will be triggered by the "Alarm buffer overflow" system event.

4. Configure the "PrintReport" system function in the job function list. Input the report with the alarm log as parameter.

5. If a system alarm should also be output, configure the "ShowSystemAlarm" system function with the desired alarm text as the parameter as the second system function in the function list of the job.

3.6 User authorizations

New user authorization concept in WinCC flexible

The concept of user authorization in WinCC flexible differs from that in ProTool.

In WinCC flexible, each users is assigned only to one user group. You assign those user groups characteristic access rights, namely the authorizations. User groups are not organized in a hierarchical structure.

A default "Administrator" user group is created in each project. Users assigned to this group in Runtime have all rights, and may also set up new users.

You determine what rights the user requires for efficient operation of the various control elements.

In Runtime, all users are identified by their user name and password.

Migration of the ProTool password level

During conversion of a ProTool project to a WinCC flexible project, the program creates up to nine user groups. One user group is created for each password level used in the ProTool project.

The "Administrators" user group is always created. All user authorizations for which password level 9 was valid in ProTool are assigned to this "Administrator" group. All other user authorizations that were assigned the same password level in ProTool are also assigned to the same user group in the WinCC flexible project. The user groups are assigned the authorizations as in the password levels of the ProTool project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Advanced options in WinCC flexible

In WinCC flexible, users may be assigned to any user group, without making any allowances for hierarchical structures. During conversion, check the project to see whether to create new user groups or not.

3.6.1 Adapting user authorizations

User administration in WinCC flexible

During migration from a ProTool project to WinCC flexible, the program creates user groups and assigns those the authorizations according to the password level settings in the ProTool project.

In addition, the migration program assigns the various control elements in WinCC flexible the same access protection as in ProTool.

The ProTool password list will be excluded from migration, because it is stored on the relevant HMI, rather than in the project data.

However, you can reuse the ProTool password lists by with the help of the export and import functions.

Export the ProTool password lists of the ProTool project to a text file using the export_import_passwords function. Migrate the ProTool project. Import the *.txt file to the migrated project using the "ExportImportUserAdministration" function.

After migration, the users must be created again. WinCC flexible provides two options in this context:

• You configure new users, user groups and passwords with the help of the Runtime user administration tool.

• Configure a screen which contains a user view. Only members of the "Administrator" user group are granted full access to the user view in Runtime. The user view shows existing users, passwords, and user groups to an administrator. In addition, administrators may also create new users, user groups and passwords with the help of the user view.

Advanced options of WinCC flexible

WinCC flexible supports the user-specific assignment of users to a user group, irrespective of any hierarchical structures. During conversion, check the project to see whether to create new user groups or not. In addition, you can assign meaningful names to the user groups created during migration by selecting "User administration runtime" in the "Groups" editor.

3.7 Recipes

3.7.1 Recipes

Introduction

In WinCC flexible, recipes and records are treated very differently than in ProTool. In some cases it may make sense to implement recipes in a converted WinCC flexible project differently from the ProTool project.

Conversion of recipes

During conversion, recipes are imported from the ProTool project and modified as much as possible for the requirements and options of WinCC flexible.

Transfer and synchronization

In WinCC flexible, recipes are transferred directly via tags with a connection to the PLC. Synchronization is only possible with a PLC.

The following settings are imported unchanged during conversion of transfer and synchronization:

- "direct transfer"
- "no synchronization"
- "synchronization to PLC_n"

The following settings are automatically adapted:

ProTool	WinCC flexible
Indirect transfer	Direct transfer
Synchronization to "All PLCs"	No synchronization

Name, Version, Number

The settings are fully imported. If there are no settings in the ProTool project, valid presets are input for the WinCC flexible project.

Identifier

The identifier is discarded in WinCC flexible. The recipes are identified by recipe number and recipe name in the PLC.

Input records again

In WinCC flexible, records are treated very differently than in ProTool. Because the records cannot be transferred from the ProTool project to the WinCC flexible project, they must be input again.

Replacing default screen

During conversion, the default screens "Z_RECORD 1" and "Z_RECORD 2", which contain the basic function or records, are removed.

The default screens can be replaced by screens that contain the "Recipe display" screen object in the WinCC flexible project.

3.7.2 Editing data record transfers

Transferring data records

Record transfer is much simpler in WinCC flexible than in ProTool. The WinCC flexible project and the PLC program must be edited because the basic functioning has changed. The editing method depends on how the record transfer was implemented in the ProTool project.

Implementation by PLC jobs 69 "Read record from PLC" and 70 "Write record to PLC"

In this case apply the "recipe number" and "record number" parameters in the WinCC flexible project again. The parameters "Identifier1" to "Identifier3" are discarded in WinCC flexible.

The "Status word" (4th word) in the "Record" area pointer must be evaluated for coordination of recipe transfer in the PLC program.

Implementation with system functions

If record transfer is implement with system functions in the source project, record transfer must be configured again. Two options are available:

- Use screen object "recipe display" to transfer records

  Configure a screen with the "Recipe display" screen object. In the properties view of the recipe display in the "Buttons" section enable the checkbox for the buttons "Read from PLC" and "Write to PLC." In runtime the operator can transfer the records to the recipe display with two buttons.

- Record transfer with system functions

  Configure the following system functions as buttons:

  – For transfer with a data medium "SetDataRecordToPLC" and "GetDataRecordFromPLC"

  – For transfer with recipe tags "ReadRecordTagFromPLC" and "WriteRecordTagToPLC"

  In runtime the operator can transfer the records with these buttons.

Editing: Checking status of record transfer

The following options are available for checking the status of record transfer:

- If system functions were used to transfer records: Evaluate the return value.

- In all other implementations: Evaluate the "status word" (4th word) in the "Record" area pointer. The PLC program may require updating after a change in the WinCC flexible project. The status word can return the following return values:

    – 0 = transfer allowed, "Record" area pointer free

    – 2 = transfer running

    – 4 = transfer completed without error

    – 12 = transfer completed with error

3.8 PLC driver

3.8.1 PLC driver

Automatic conversion

Most PLC drivers available in ProTool are also available in WinCC flexible. They are automatically converted during conversion. This applies to the following PLC drivers:

- SIMATIC S7-300/400
- SIMATIC S7-200 (one PLC)
- SIMATIC 500/505
- Allen-Bradley DF1 PLC-5
- Allen-Bradley DF1 SLC 500
- Allen-Bradley DH485
- GE Fanuc SNP
- MITSUBISHI FX
- Modicon Modbus

PLC drivers that are not automatically converted

The following PLC drivers cannot be used directly after migration:

- SIMATIC S5 DP
- SIMATIC S5-FAP
- SIMATIC S5 - AS511
- SIMATIC S7-200 (2 PLCs with PPI coupling)
- Omron Host-Link/Multilink

In most cases a different PLC driver must be selected before or after conversion.

PLC drivers without equivalents in WinCC flexible

The following PLC drivers are not available in WinCC flexible:

- FREE SERIAL

- Telemecanique Adjust

- SIMATIC S7-NC

If a project that is linked to a PLC for which there is no protocol available in WinCC flexible, the SIMATIC S7-300/400 PLC driver will be set in the destination project. All tags will be separated from the PLC.

3.8.2 Converting a project for the PLC SIMATIC-S5-DP

Converting with S5 AS511 PLC driver

The SIMATIC S5-L2 DP PLC driver in ProTool corresponds to the SIMATIC S5-DP PLC driver in WinCC flexible. However, direct conversion is not possible.

If the SIMATIC S5 L2 DP PLC driver was used in the ProTool project, a SIMATIC S5 AS511 must be selected in ProTool before migration. After migration switch to a SIMATIC S5 DP in WinCC flexible.

Procedure

1. Open the ProTool project in ProTool.

2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 L2 DP" protocol is now shown on the right.

3. Double-click on the input PLC to change the PLC protocol.

4. Select the SIMATIC S5 AS511 PLC protocol from the "PLC" dialog. Confirm with "OK".

5. Save the project.

6. Close ProTool.

7. Open the project in WinCC flexible.

8. Select the HMI device that you want to convert for your project.

   The project is converted and then opened in WinCC flexible.

   It is now connected to the SIMATIC S5 AS511.

9. Select the "Connections" entry in the project view for the HMI device. The connections editor opens.

10. Select "SIMATIC S5 DP" as connection

11. Replace the default FB (FB 58) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

3.8.3 Converting a project for the PLC SIMATIC-S5-FAP

Introduction

There is no direct equivalent in WinCC flexible for the SIMATIC S5 FAP PLC driver which is available in ProTool. However, the project can be converted to SIMATIC S5 AS511 or to SIMATIC S5 DP.

Converting with S5 AS511 PLC driver

If the SIMATIC S5 FAP PLC was used in the ProTool project, a SIMATIC S5 AS511 must be selected in ProTool before migration. After migration you can switch to a SIMATIC S5 DP in WinCC flexible.

Procedure

1. Open the ProTool project in ProTool.

2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 FAP" protocol is now shown on the right.

3. Double-click on the input PLC to change the PLC protocol.

4. Select the SIMATIC S5 AS511 PLC protocol from the "PLC" dialog. Confirm with "OK".

5. Save the project.

6. Close ProTool.

7. Open the project in WinCC flexible.

8. Select the HMI device that you want to convert for your project.

   The project is converted and then opened in WinCC flexible.

   It is now connected to the SIMATIC S5 AS511.

9. To switch to a SIMATIC S5 DP: Select the "Connections" entry in the project view for the HMI device. The connections editor opens.

10. Select "SIMATIC S5 DP" as connection

11. If you are working with a SIMATIC S5 DP, replace the default FB (FB 58) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

3.8.4 Editing a project for the PLC SIMATIC-S5-AS511

After converting

After conversion, of a project with the SIMATIC S5 AS511, the following items must be noted:

- Delete the default FB (FB51) in the PLC program because it is not required for communications. Otherwise errors may be encountered during communications between PLC and HMI device.

- The standard 6XV1440-2A cable can continue to be used for the MP 270 and MP 370. All other devices (including the MP 270B) require a converter from V.24 to TTY (MLFB no.: 6ES5 734-1BD20).

3.8.5 Converting a project with two SIMATIC S7-200 PLCs

Converting after profile change

With ProTool projects for graphic devices it is possible to configure multiple PLCs with the SIMATIC S7-200 PLC driver when the PPI profile is used.

In WinCC flexible projects two PLCs can only be coupled with the MPI communication network. The PLC profile must be changed in the ProTool project before conversion to ensure a successful conversion.

Procedure

1. Open the ProTool project in ProTool.

2. From the project view select the "PLC" object. The two PLCs with the SIMATIC S7-200 protocol are now displayed on the right.

3. Double-click on the first PLC to open the "PLC" dialog box.

4. Click on "Parameter" and select the entry "MPI" as "Profile." Confirm the setting with "OK."

5. Save the project.

6. Close ProTool.

7. Open the project in WinCC flexible.

8. Select the HMI device that you want to convert for your project.

   The project is converted and then opened in WinCC flexible.

3.8.6 Editing a project for the PLC Omron Host-Link/Multilink

After converting

The station address is set to 0 during conversion of a project with the Omron Host-Link/Multilink PLC driver. After conversion, the correct station address is set in the OP 77B project as follows:

Procedure

1. Click in the project window on "Communication"> "Connections."

2. Select the entry "Omron Host-Link/Multilink" in the "Communications driver" column.

3. Select the "Parameter" entry in the properties view.

4. Enter the correct station address.

Migration from OP3 to OP 73 4

4.1 Basic principles

Introduction

The OP3 HMI device is not supported by WinCC flexible. If you want to continue to use ProTool projects created for this HMI device and migrate them, the HMI device is automatically changed to the OP 73 during migration.

If you want to continue to use an OP3 project on an OP 73micro, you must first perform a migration to OP 73. You then switch the HMI device type in your OP 73 project to an OP 73micro.

If the ProTool project has more than one editing language, you will be prompted to select one of the editing languages as the reference language when starting the conversion. The reference language influences the conversion of screens that consist of static texts and IO fields positioned between the texts. The layout of text sections and IO fields is optimized for the selected reference language during migration. For the other editing languages the relevant graphics may need editing in some cases.

Conversion

Because the OP3 and OP 73 HMI devices have different functional scopes and principles of operation, the configuration must be changed during migration. The required changes are performed automatically for the most part during migration.

The fundamental changes cover the following areas:

- Screens
- Tags and area pointers
- System functions, events, and PLC jobs
- Alarm system
- User authorizations
- PLC driver

Migration of C7-621 projects

The C7-621 HMI device is not supported by WinCC flexible. ProTook projects created for the C7-621 are migrated to a WinCC flexible project for the OP 73 HMI device.

4.2 Screens

4.2.1 Screen entries

Conversion

Every screen in OP3 projects can be subdivided into up to 20 screen entries through which the user can scroll. During migration of an OP3 project with screen entries, every single screen entry is converted to an independent screen. The cursor keys are also automatically assigned with ChangeScreens during migration. The user can therefore scroll through the screens that were created from the screen entries of the same OP3 screen in the OP 73 as in the OP3 project.

Assignment of screen names and screen numbers

The screens created during migration that replace other screen entries are given screen names and screen numbers according to the following convention:

- The screen that was created to replace the first screen entry of an OP3 screen receives the name and number of the OP3 screen.

- All subsequent screen entries are given the name and screen number of the OP3 screen extended by one digit in ascending order from 02 to 20.

Screen series whose screens were created from the screen entries of the same OP3 screen can be identified after migration from the matching components of the new screen names and screen numbers.

Application example

An OP3 project that you would like to convert to an OP 73 project contains a "screen" with screen number 1 that is divided into three screen entries.

The following screen series is created during migration:

OP3 project, "screen", screen number 1	Screen series of the OP 73 project
Screen entry 1	"Screen", screen number 1
Screen entry 2	"screen_2", screen number 102
Screen entry 3	"screen_3", screen number 103
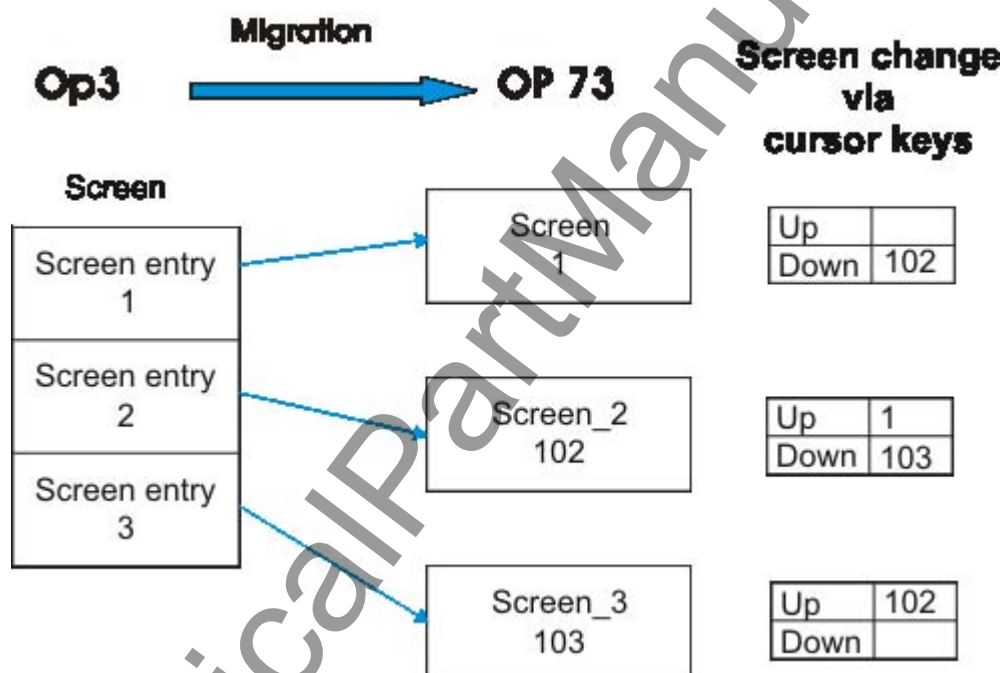
Conversion of screen entries of the OP3 to screens for the OP 73

Navigation

During migration the cursor keys are assigned so the user can navigate through the screens of a screens series as accustomed to in OP3.

Screen title

Screen title display on the OP 73 is not supported.

4.2.2 Navigation

Introduction

During migration of an OP3 project with screen entries every single screen entry is converted to an independent screen of the OP 73 project. Even after migration, associated screen series whose screens come from the screen entries of the same OP3 screen can be identified. During conversion to an OP 73 project, the cursor keys and the <Esc> key are assigned so the user can navigate within a screen and among the screens of a series in the same way as in OP3. The <Tab> key can also be used to navigate, in addition to the cursor keys.

Navigation on the OP 73

The following key assignments are set up in the OP 73 for navigation within a screen and to the other screens of a screen series:

- Cursor keys

Cursor key	Within a screen	Among the screens of a series
<Down>	To the next lower IO field	To the next screen of the same series; at the bottom end of the last screen of a series: unassigned
<Up>	To the next higher IO field	To the previous screen of the same series; at the top end of the first screen: unassigned
<Right>	To the next IO field in the same row	unassigned
<Left>	To the previous IO field in the same row	unassigned

- Key <TAB>

  <Tab> switches to the next IO field within a screen.

- <Esc> key

  – If <Esc> is pressed while editing an IO field, the input is discarded.

  – In an OP 73 project, press <Esc> twice outside of the editing process to return to the previous screen.

  In an OP 73 project, a return always goes back to the calling screen (implemented by means of the "ActivatePreviousScreen" system function). A return within a screen series causes the previous screen of the series to open. This is different from the behavior of the OP3 project, in which a return operation always goes back to the first entry of the series.

4.2.3 Default screens and special screens

Introduction

The default screens of an OP3 project ("Z_COUNTER", "Z_PASSWORD", "Z_SETTINGS", "Z_SYSTEM_MEN", and "Z_TIMER") are migrated in the same way as configured screens.

Instead of special screens, which cannot be edited in an OP3 project, new, preconfigured screens with the same functionality are created in the OP 73 project during migration. Some special screens do not require replacement or cannot be replaced because the different functional scope and principle of operation of the OP 73 are different. During the conversion, the screen layout is adapted to fit the OP73.

Conversion of Special Screens

The special screens of the OP3 are converted as follows:

Special screen on OP3	Replacement on the OP 73
Changing the operating mode	Replaced by the preconfigured screen "P_MODE_SCREEN". "P_MODE_SCREEN" contains three buttons with corresponding labels. A different value of the "SetDeviceMode" system function depending on the operating mode is configured to every one of these buttons.
Language	Replaced by the preconfigured screen "P_LANGUAGE_SCREEN". "P_LANGUAGE_SCREEN" contains three buttons, labeled accordingly.
	Button 1: Set language ("SetLanguage" system function)
	Button 2: Increase contrast ("AdjustContrast(1)" system function)
	Button 3: Reduce contrast ("AdjustContrast(0)" system function)
Time/date	Replaced by the preconfigured screen "P_DATE_TIME_SCREEN". "P_DATE_TIME_SCREEN" contains two date-time fields with corresponding labels, one to show the date and the other to show the time.
	The day of the week cannot be input but is calculated internally from the date so it can be used in PLC jobs.
MPI address	Not supported because on the OP 73 the communication settings are made via the loader menu.
Tag status	Not supported
	Instead, a screen can be configured in which the relevant positions of the PLC can be accessed via tags. The tags are linked with IO fields to display (tag status) and modify (tag control) the contents of the tags on the HMI device.
Force tag	Not supported
	Instead, a screen can be configured in which the relevant positions of the PLC can be accessed via tags. The tags are linked with IO fields to display (tag status) and modify (tag control) the contents of the tags on the HMI device.
Password - edit	Replaced by the preconfigured screen "P_USER_SCREEN". "P_USER_SCREEN" contains a user display.

4.2.4 Screen objects

Introduction

During conversion of an OP3 project to an OP 73 project, screen objects generally are automatically changed to function with the OP 73. A brief visual check is useful after a conversion.

Converting IO fields

During conversion, the output fields, input fields, and input/output field are converted to an IO field with a similar configuration. IO fields can be operated after conversion with the same function keys as in the OP3 project (cursor keys, <Enter>).

In detail the fields are converted as follows:

OP3	OP 73
Output field	IO field, configured as output field
Input field	IO field, configured as input field
IO field	IO field, configured as input/output field
Output field of text type	Symbolic IO field, configured as output field
Input field of text type	Symbolic IO field, configured as input field
Input/output field of text type	Symbolic IO field, configured as input/output field

Converting date/time fields

The length of date and time formats may vary in multi-lingual projects. Adjust the length of the date/time fields manually in the configuration, if necessary.

Converting text

The following changes and restrictions are involved when converting text:

- Font: The OP3 font is converted to the default OP 73 font on the OP 73.

- The special screens of OP3 are converted. Because only three rows are available on the OP 73 screen, the screen layout is adjusted accordingly during conversion.

- If you want to convert to "Tahoma" font in the OP 73 project after conversion, the following characters are not supported when "European" and "European II" fonts are used:

Characters not supported			
←	↑	→	↓
α	π	Σ	σ
Ω	∞	ε	√

Converting the layout of a screen entry

During conversion of a screen entry of the OP3 to a screen on the OP 73, all static texts are converted into one large text field, which is as large as the complete screen. The IO fields are laid out to optimize them for the reference language. For this reason in multilingual projects the positions of the IO fields in the non-reference languages must be checked. If necessary, the IO field positions must be adjusted.

Flashing

The text property "Flashing" is only imported to the OP 73 during conversion if the attribute is assigned to the complete text of a screen object. Flashing text sections are no longer supported on the OP 73. If a text field consists of several words and one of the words should flash, the text field must be separated into several text fields.

4.2.5 Softkeys

Introduction

The OP 73 has four function keys - one less than OP3. Numerical values can only be entered on the OP 73 by means of the cursor keys.

Migration of function keys

Migration of the function key assignment depends on the number of keys assigned in the OP3 project:

- OP3 project with five assigned function keys

  The assignment of <F5> is omitted during migration. When migration is complete, an alarm to this effect is displayed in the log file.

- OP3 project with four assigned function keys and no <F5> assignment

  The function key assignment is accepted 1:1.

- OP3 project with at most four assigned function keys and an <F5> assignment

  During migration, the first unassigned key is determined from right to left (beginning with <F5>). The assignment of all keys to the right of the unassigned key is shifted one key to the left. When migration is complete, an alarm is displayed in the log file for each shifted key assignment.

System keyboard mapping

The system keyboard mapping cannot be migrated because WinCC flexible does not have a corresponding area pointer.

You must reconfigure the system key assignment in the migrated WinCC flexible project, whereby functions can only be assigned to the <ACK>, <ESC>, and <ENTER> system keys on the OP 73.

To replace the "system keyboard" area pointer in ProTool, create an external tag in the WinCC flexible project with the same address in the PLC as the area pointer in the ProTool project. The length of the tags is based on the number of system keys, such that one bit of the tag is available for each assignable key.

If you want to evaluate the hotkey in the PLC, configure the system function "SetBitWhileKeyPressed" to the "Press" event for all system keys. When the user presses a key in runtime the corresponding bit in the external tag is set. This gives the PLC the information that the key is pressed. Because the external tag has the same address in the PLC as the area pointer in the ProTool project, the PLC program does not require any changes.

Migration of the property "Bit in tag"

If the property "Bit in tag" is configured for a function key or system key in the OP3 project, it is replaced during migration to an OP 73 project. As part of the migration process, the system function "SetBitWhileKeyPressed" is configured for the event "Press".

4.3 Tags and area pointers

4.3.1 Tags

Introduction

During migration, tags corresponding to the tags in the OP3 project are created in the OP 73 project. In addition, a connection to the PLC is created for external tags.

Importing decimal places

Decimal places cannot be configured for a new tag on the OP 73. Tags that had decimal places in the OP3 project are converted as follows:

- Tags that are configured to an IO field retain decimal places.

- The decimal places of tags that are displayed in the alarm text are discarded in alarms.

4.3.2 Area pointer

Introduction

During migration from an OP3 to an OP 73, the area pointers are converted as follows:

- "Screen number" area pointer

  During migration, three words are added to the Screen number area pointer.

  Under ProTool, a screen change was caused by evaluation of the first word of the screen number area and the subsequent writing of the new screen number to the second word of the screen number area. After migration, use the "ActivateScreenByNumber" system function to implement this process in WinCC flexible.

- "Process events" area pointer

  This area pointer is not used.

  As a replacement, an external array tag with a 16-bit array element and a corresponding name is created during the conversion. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the OP3 area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers.

  As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." In reverse, the alarm is interpreted as "sent" after the bit in the array element is reset to "FALSE" on the PLC.

- Area pointer "interface area"

  This area pointer is not used. Instead, data are exchanged with the PLC in WinCC flexible by means of the following area pointers:

  - "Coordination"
  - "Date/time control"

  Once migration is complete, you must create the area pointer in the project.

  This distribution simplifies data exchange between HMI device and PLC. The PLC program must be configured accordingly after conversion.

- "System keyboard" area pointer

  This area pointer is not used.

  To replace this area pointer, proceed as follows: Configure an external 3-bit tag according to the number of assignable system keys. The tag now has one bit for each system key (ACK, ESC, and ENTER). In the template of the OP 73 project, configure the "SetBitWhileKeyPressed" system function for each previously assigned system key. Set the parameters of the system function so the key is set to the corresponding bit of the external tag. The tag status can then be evaluated in the PLC program.

4.3.3 "Interface area" area pointer

Introduction

The "interface area" pointer is not used. Instead, data are exchanged with the PLC in WinCC flexible by means of the following area pointers:

- "Coordination"
- "Date/time control"

  This area pointer is used to transfer the date and time from the PLC to the OP. Based on the refresh time, the OP system time will be overwritten with the data from this PLC area. Prerequisite is, that all data which are valid according to the defined format are stored on the PLC.

Once migration is complete, you must create the area pointer in the project.

This distribution simplifies data exchange between HMI device and PLC. The PLC program must be configured accordingly after conversion.

Distribution to S7 PLCs

The interface range is distributed as follows for S7 PLCs:

ProTool "Interface area" Pointer	Replaced by WinCC flexible area pointer
The Interface area n+0 to n+31 pointer is distributed as follows: • Coordination n+0 to n+1 • Coupling ID n+13 • Date/time n+15 to n+26	Coordination n+0 to n+1 area pointer Length of the date/time control area pointer: 6 words • Byte 1: Year (in BCD code) • Byte 2: Month (in BCD code) • Byte 3: Day (in BCD code) • Byte 4: Hour (in BCD code) • Byte 5: Minute (in BCD code) • Byte 6: Second (in BCD code) • Byte 7: Free • Byte 8: Weekday (in BCD code) • 9. - 11. Free byte Coupling ID is not required.

Handling of the start bit (coordination area pointer)

The coordination area pointer contains the start bit. If the value of this bit is read out in the PLC program, you can see whether the HMI device has started up from the PLC. The start bit is temporarily set to "0" by the HMI device during the startup process. As soon as the boot process is complete, the start bit is set to "1".

4.4 System functions

4.4.1 System functions

Conversion

During migration from an OP3 to an OP 73 the ProTool system functions of the OP3 project are replaced by the corresponding WinCC flexible system functions.

In WinCC flexible, the names of the system functions are assigned according to a specific convention. For this reason the names of the WinCC flexible system functions are not identical to same as the names of the corresponding ProTool system functions. The system functions groups in WinCC flexible also have different names from those in ProTool.

The system functions are converted as follows:

OP3 project: ProTool system function	OP 73 project: WinCC flexible system functions
Calculate	
Linear conversion	Defined as property of the tag.
Editing bits	
Set bit	Replaced by the "SetBit" system function.
Reset bit	Replaced by the "ResetBit" system function.
Password	
Password logout	Replaced by the "Logoff" system function.
Screens	
Screen branching	No longer required; navigation among the screens created from the screen entries during migration is implemented with the "ActivateScreen" system function.
Special screen selection	No longer required.
Display selection	Replaced by the "ActivateScreen" system function.
Screen table of contents	No longer required.
Toggle	
Language change	Replaced by the "SetLanguage" system function.
Alarm level	Replaced by the "ActivateScreen" system function which can be used to call a screen with an alarm view.
Changing the operating mode	Replaced by the "SetDeviceMode" system function. The "5 = MPI transfer" parameter value is changed to the "2 = (serial) transfer" parameter value.

4.4.2 Executing system functions from the PLC

Configuring

If you want the PLC to trigger the execution of a system function configured on the HMI device, configure a tag with the property "In consecutive cycles" in WinCC flexible. Configure the corresponding system function for the "Change value" event on this tag.

Execution

If the value of the tag changes in the PLC, the system function is executed on the HMI device as soon as it detects the change.

Example: Trigger system function "ActivateScreenByNumber" from the PLC

With the following configuration, you can page to the screen with the next highest screen number from the PLC using the "ActivateScreenByNumber" system function.

Requirements

The "Screen Number" area pointer is created.

In WinCC flexible

1. Create an external tag of the INTEGER type.
2. Set in the "Basic Settings" area of the properties window of the tag for the "In consecutive cycles" update.
3. Configure the "ActivateScreenByNumber" system function to the event "Change value." Input the tag itself as parameter.

On the PLC

1. In the PLC program, read out the "Screen number" area pointer.
2. Increase the value read out by 1.
3. Write the new value to the external tag.

Result

The tag value changes. The "ActivateScreenByNumber" system function is executed. The corresponding screen is displayed, and the value contained in the "Screen Number" area pointer is updated.

If the PLC program is executed again, the next screen is displayed.

4.5 Alarm system

Introduction

During migration from an OP3 to an OP 73, alarms of all existing alarm classes are converted; as a result, the following alarms classes are still available:

- Warning alarms

- System alarms

Tags and text lists that are included in the alarm text are imported during conversion.

Tags displayed in the alarm message indicate the value that the tag had when the alarm came in and are not continuously updated in WinCC flexible.

Preconfigured screen replaces alarm level

The alarm level is not supported on the OP 73. Instead, during migration a preconfigured screen called "P_MESSAGE_SCREEN" is created to show the alarms. This screen shows the alarms with the aid of the "Simple alarm display" screen object. The alarm display includes the system date and time. A display in the alarm text is therefore no longer required. The date/time display is removed from the alarm text during conversion. The alarm display shows the alarm number, alarm text and the abbreviation of the alarm class in a two-line view for every alarm.

"P_MESSAGE_SCREEN" is called during runtime with the "ActivateScreen" system function, which replaces the ProTool function "Alarm Level".

Process events are displayed in the alarm window.

During conversion, an alarm window is created in the template for automatic display of process events. The alarm window is configured so that it opens automatically when a process event arrives.

Rest alarm

The rest alarm is not supported.

Priority of alarms

In the OP 73 priorities can only be assigned for alarm classes, not for individual alarms. Priorities that were assigned to single alarms in the OP3 are discarded after conversion.

Conversion of the alarm area pointers

During conversion of an OP3 project to an OP 73 project, external array tags with a 16-bit array element are created to replace the "Process events" area pointer. Corresponding names are given to the array tags. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the OP3 area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers.

As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." In reverse, the alarm is interpreted as "sent" after the bit in the array element is reset to "FALSE" on the PLC.

4.6 User authorizations

4.6.1 Security concept in WinCC flexible

New user authorization concept in WinCC flexible

The concept of user authorization in WinCC flexible differs from that in ProTool.

In WinCC flexible, each users is assigned only to one user group. You assign those user groups characteristic access rights, namely the authorizations. User groups are not organized in a hierarchical structure.

A default "Administrator" user group is created in each project. Users assigned to this group in Runtime have all rights, and may also set up new users.

You determine what rights the user requires for efficient operation of the various control elements.

In Runtime, all users are identified by their user name and password.

Migration of the ProTool password level

During conversion of a ProTool project to a WinCC flexible project, the program creates up to nine user groups. One user group is created for each password level used in the ProTool project.

The "Administrators" user group is always created. All user authorizations for which password level 9 was valid in ProTool are assigned to this "Administrator" group. All other user authorizations that were assigned the same password level in ProTool are also assigned to the same user group in the WinCC flexible project. The user groups are assigned the authorizations as in the password levels of the ProTool project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Advanced options in WinCC flexible

In WinCC flexible, users may be assigned to any user group, without making any allowances for hierarchical structures. During conversion, check the project to see whether to create new user groups or not.

4.6.2 Adapting user authorizations

Introduction

During conversion of an OP3 project to an OP 73 project, user groups are created with the same authorizations that were assigned to the password levels in the OP3 project.

In addition, the various control elements are assigned the same access protection as in ProTool after migration.

Password list is not migrated

The password list is not migrated because it is saved on the specific HMI device and is not in the project data.

After migration, the users must be created again. There are two options for this on the OP 73:

- During configuration, new users, user groups, and passwords are created with the runtime user administration for the panel.

- A screen with a user view is configured. The user view shows existing users, passwords, and user groups to an administrator. An administrator can also create new users, user groups and passwords with the user view.

Conversion of special screen for password editing

The call of the Password - Edit special screen is replaced by the call of the preconfigured "P_USER_SCREEN." "P_USER_SCREEN" contains a user display.

Advanced Options on the OP 73

On the OP 73 users can be assigned to a user group as desired without worrying about hierarchies. During conversion, check the project to see whether new user groups should be created. In addition, appropriate names for the user groups created during migration can be assigned in the "Groups" editor in the "User administration runtime" group.

4.7 PLC driver

1:1 conversion

The following PLC drivers are available for both the OP3 and the OP 73.

- SIMATIC S7-300/400 (not for OP 73micro)
- SIMATIC S7-200 (one PLC)

These PLC drivers are automatically converted during conversion.

Project with two SIMATIC S7-200 PLCs

With OP3 projects, it is possible to configure multiple PLCs with the SIMATIC S7 200 PLC driver protocol when the PPI profile is used.

On the OP 73, only two PLCs can be coupled with the MPI communication network. Therefore, after the conversion, set "MPI" as the network for the communication for both SIMATIC S7 PLCs.

Migration of OP7 to OP 77

5

5.1 Basic principles

Introduction

The OP7 HMI device is not supported by WinCC flexible. If you want to continue to use ProTool projects created for this HMI device and migrate them, the HMI device is automatically changed to the OP 77B during migration.

If you want to migrate a ProTool project for the OP7 HMI device to the OP 77A HMI device, you have to first migrate the project to the OP 77B HMI device. After migration, execute a change of HMI device to OP 77A in WinCC flexible.

If the ProTool project has more than one editing language, you will be prompted to select one of the runtime languages as the reference language when starting the conversion. The reference language influences the conversion of screens that consist of static texts and IO fields positioned between the texts. The layout of text sections and IO fields is optimized for the selected reference language during migration. For the other runtime languages the relevant graphics may need editing in some cases.

Conversion

Because the OP7 and OP 77B HMI devices have different functional scopes and functioning, the configuration must be changed during migration. The required changes are performed automatically for the most part during migration.

The fundamental changes cover the following areas:

- Screens
- Tags and area pointers
- System functions, events, and PLC jobs
- Alarm system
- User authorizations
- Print
- Recipes
- PLC driver

5.2 Screens

Conversion

Every screen in OP7 projects can be subdivided into up to 99 screen entries through which the user can scroll. During migration of an OP7 project with screen entries, every single screen entry is converted to an independent screen. The cursor keys are also automatically assigned with ChangeScreens during migration. The user can therefore scroll through the screens that were created from the screen entries of the same OP7 screen in the OP 77B as in the OP7 project.

Assignment of screen names and screen numbers

The screens created during migration that replace other screen entries are given screen names and screen numbers according to the following convention:

* The screen that was created to replace the first screen entry of an OP7 screen receives the name and number of the OP7 screen.

* All subsequent screen entries are given the name and screen number of the OP7 screen extended by one digit in ascending order from 02 to 99.

Screen series whose screens were created from the screen entries of the same OP7 screen can be identified after migration from the matching components of the new screen names and screen numbers.

Example of an application

The following screens and screen entries are contained in an OP7 project that you want to convert to an OP 77B project:

* "Screen" with screen number 1 divided into three screen entries

* "Record" with screen number 2 also divided into three screen entries

The following screen series are created during migration:

OP7 project, screen "Screen", screen number 1	Screen series of the OP 77B project
Screen entry 1	"Screen", screen number 1
Screen entry 2	"screen_2", screen number 102
Screen entry 3	"screen_3", screen number 103

OP7 project, screen "Record", screen number 2	Screen series of the OP 77B project
Screen entry 1	Screen "Record", screen number 2
Screen entry 2	Screen "Record_3", screen number 202
Screen entry 3	Screen "Record_3", screen number 203
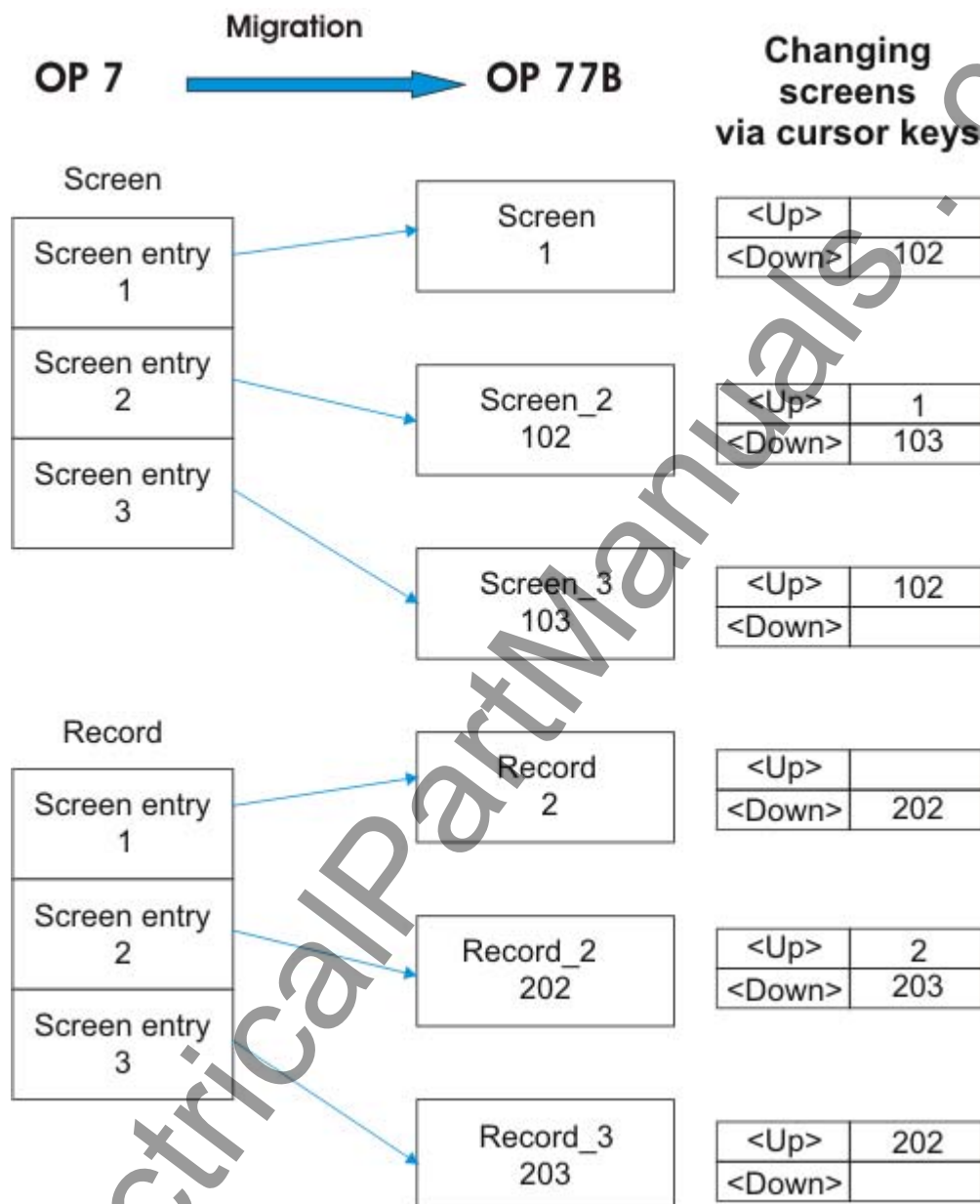
Conversion of screen entries of the OP7 to screens for the OP 77B

Navigation

During migration the cursor keys are assigned so the user can navigate through the screens of a screens series as accustomed to in OP7.

Screen title

Screen title display on the OP 77B is not supported.

Introduction

During migration of an OP7 project with screen entries every single screen entry is converted to an independent screen of the OP 77 project. Even after migration associated screen series whose screens come from the screen entries of the same OP7 screen can be identified. During conversion to an OP 77B project, the cursor keys and the <Esc> key are assigned so the user of OP7 can navigate within a screen and among the screens of a series in the same way as in OP7. The <Tab> key can also be used to navigate, in addition to the cursor keys.

Navigation on the OP 77B

The following key assignments are set up in the OP 77B for navigation within a screen and to the other screens of a screen series:

- Cursor keys

Key	Within a screen	Among the screens of a series
<Down>	To the next lower IO field	To the next screen of the same series; at the bottom end of the last screen of a series: unassigned
<Up>	To the next higher IO field	To the previous screen of the same series; at the top end of the first screen: unassigned
<Right>	To the next IO field in the same row	unassigned
<Left>	To the previous IO field in the same row	unassigned

- Key <TAB>

Key	Within a screen	Among the screens of a series
<Tab>	To the next IO field	unassigned
<Shift + Tab>	To the previous IO field	unassigned

- <Esc> key

  – If <Esc> is pressed while editing an IO field, the input is discarded.

  – The following assignment is made for the <Esc> key for the migration depending on the assignment in the OP7 project:

Function of <Esc> in the OP7 project	Function of <Esc> in the OP 77B project
Switch to screen directory	Screen directory no longer available. Instead, return to the calling screen (implemented with the system function "ActivatePreviousScreen")
Switch to alarm level	Assignment with the system function "ActivateScreen"; triggers change to preconfigured screen "P_MESSAGE_SCREEN" with an alarm view. "P_MESSAGE_SCREEN" was created during conversion. Key assignment in "P_MESSAGE_SCREEN": • <Esc> key with return to the called screen (implemented with the system function "ActivatePreviousScreen")

Function of <Esc> in the OP7 project	Function of <Esc> in the OP 77B project
Switch to recipe table of contents	Assignment with the system function "ActivateScreen"; triggers change to preconfigured screen "P_RECIPE_SCREEN" with a recipe display. "P_RECIPE_SCREEN" was created during conversion. Key assignment in "P_RECIPE_SCREEN": • <Esc> key with return to the called screen (implemented with the system function "ActivatePreviousScreen")
Switch to previous screen	Return to the called screen (implemented with the system function "ActivatePreviousScreen") A return within a screen series causes the previous screen of the series to open. This is different from the behavior of the OP7 project in which it always jumps back to the first entry of the series.
Switch to a specified screen	Screen switch to a specified screen (implemented with the system function "ActivateScreen", parameter "Name of the specified screen")

Introduction

During migration to the OP 77B project, new preconfigured screen, with the same functions are created to replace most of the default screens of the OP7 project. Some default screens do not require replacement or cannot be replaced because of the changed scope of function and changed functions of the OP 77B.

Conversion of default screens

The default screens of the OP7 are converted as follows:

Default screen in the OP7	Replacement on the OP 77B
Error alarms - appearance	Converted in screen with alarm view The alarm view is configured so the error alarms are displayed from the error alarm buffer.
Error alarms - printing	Replaced by the "PrintReport" system function which is used to trigger printing of a report. The report is configured so the contents of the error alarm buffer are printed.
Error alarms - number	Discarded. All error alarms are shown in the alarm view that displays the error alarms.
Error alarms - delete	The function "ClearAlarmBufferProTool" is used for the migration. The parameters of this system function are identical to those in ProTool function. Only use the system function "ClearAlarmBuffer" for new projects. Has new parameters.
Error alarms - overflow	Discarded.
Error alarms - texts	Discarded. The texts of the error alarms are also shown in the alarm view that displays the error alarms.
Warning alarms - appearance	Converted in screen with alarm view The alarm view is configured so the warning alarms are displayed from the event buffer.

Default screen in the OP7	Replacement on the OP 77B
Warning alarms - printing	Replaced by the "PrintReport" system function which is used to trigger printing of a report. The report is configured so the contents of the event buffer are printed.
Warning alarms - number	Discarded. All warning alarms are shown in the alarm view that displays the warning alarms.
Warning alarms - delete	The function "ClearAlarmBufferProTool" is used for the migration. The parameters of this system function are identical to those in ProTool function. Only use the system function "ClearAlarmBuffer" for new projects. Has new parameters.
Warning alarms - overflow	Discarded.
Warning alarms - texts	Discarded. The texts of the warning alarms are shown in the alarm view that displays the warning alarms.
Screens - editing	Discarded. In Runtime screens can be opened directly and edited.
Screens - printing	Discarded. Printing of screens can be configured as follows: • With the "PrintScreen" system function to print the currently displayed screen • Configure the screen objects again in a report and print the report using the system function "PrintReport". Printing of all screens at once is not supported. If you want to configure reporting of process values from different screens, corresponding reports must be created in the project. The menu item "Project" > "Print project documentation" can be used to document the configuration phase of the project.
Data records - editing	Is replaced with the "P_RECIPE_SCREEN" screen template which has an advanced recipe view. On this screen, you can display and edited the data records of recipes, and transfer these to the HMI.
Data records - transfer	Is replaced with the "P_RECIPE_SCREEN" screen template which has an advanced recipe view. On this screen, you can display and edited the data records of recipes, and transfer these to the HMI.
Data records - printing	Discarded. Can be replaced by printing a report with the "Recipe print" object.
System settings - operation	Will be replaced with the "P_MODE_SCREEN" screen template with three accordingly labeled buttons. A different value of the "SetDeviceMode" system function depending on the operating mode is configured to every one of these buttons.
System settings - AlarmView	Not supported The sequence in which the alarms are to be displayed (oldest or newest first) can be specified when the alarm view is configured. This setting cannot be changed in runtime.
System settings - SysAlarm	Converted in a screen with an alarm view The alarm view is configured so the system alarms are displayed from the system alarm buffer.
System settings - languages	Will be replaced with the "P_LANGUAGE_SCREEN" screen template with three accordingly labeled buttons. • Button 1: Set language ("SetLanguage" system function) • Button 2: Increase contrast ("AdjustContrast(1)" system function) • Button 3: Reduce contrast ("AdjustContrast(0)" system function)

Default screen in the OP7	Replacement on the OP 77B
System settings - date/time	Will be replaced with the "P_DATE_TIME_SCREEN" screen template which contains two date-time fields with corresponding labels, one to show the date and the other to show the time.
	The day of the week cannot be input but is calculated internally from the date so it can be used in the job planner or PLC jobs.
	The date/time field of the OP 77B is longer in the OP7. Adapt the screen configuration after migration if needed.
System settings - printer	No longer required; the printer parameters are set with the loader menu on the OP 77B.
System settings - IF1A/RS232 IF1A/TTY IF1B IF1A/B IF2	Not supported because on the OP 77B the communication settings are made via the loader menu.
StatVAR	Not supported
	Instead, a screen can be configured in which the relevant positions of the PLC can be accessed via tags. The tags are linked with IO fields to display the contents of the tags on the HMI device (StatVar) and to change them (SteuVar).
SteuVAR	Not supported
	Instead, a screen can be configured in which the relevant positions of the PLC can be accessed via tags. The tags are linked with IO fields to display the contents of the tags on the HMI device (StatVar) and to change them (SteuVar).
Password processing - login	Replaced by calling the "ShowLogonDialog" system function.
Password processing - logout	Replaced by calling the "Logoff" system function.
Password processing - edit	Replaced by the preconfigured screen "P_USER_SCREEN". This screen contains the password view.

Introduction

During conversion of an OP7 project to an OP 77B project, screen objects are generally automatically changed to function with the OP 77B. A visual check is useful after a conversion.

Converting IO fields

During conversion, the output fields, input fields, and input/output field are converted to an IO field with a similar configuration. IO fields can be operated after conversion with the same function keys as in the OP7 project (cursor keys, <Enter>).

In detail the fields are converted as follows:

OP7	OP 77B
Output mode	IO field, configured as output field
Input mode	IO field, configured as input field
Input/output field	IO field, configured as input/output field
Output field of text type	Symbolic IO field, configured as output field
Input field of text type	Symbolic IO field, configured as input field
Input/output field of text type	Symbolic IO field, configured as input/output field

Converting date/time fields

The length of date and time formats may vary in multi-lingual projects. Adjust the length of the date/time fields manually in the configuration, if necessary.

Converting text

The following changes and restrictions are involved when converting text:

- Font: The OP7 font is converted to the default OP 77B font on the OP 77B.

- If you want to convert to the "Tahoma" font in the OP 77B project after conversion, the following characters are not supported, depending on the character set:

OP7 character set "European" and "EuropeanII:

OP7 character set "Russian"

OP7 character set "Czech", "Hungarian" and "Polish"

OP7 character set "Greek"

OP7 character set "Turkish"

Converting the layout of a screen entry

During conversion of a screen entry of the OP7 to a screen on the OP 77B all static texts are converted into one large text field which is as large as the complete screen. The IO fields are laid out to optimize them for the reference language. For this reason in multilingual projects the positions of the IO fields in the non-reference languages must be checked. If necessary, the IO field positions must be adjusted.

Flashing

The text property "Flashing" is only imported to the OP 77B during conversion if the attribute is assigned to the complete text of a screen object. Flashing text sections are no longer supported on the OP 77B. If a text field consists of several words and one of the words should flash, the text field must be separated into several text fields.

Tag list

The tag list is not supported on the OP 77B.

Screen object Status force

The status force screen object is not supported.

Introduction

The OP7 has the same function keys as the OP77B.

During migration the greater part of the function key assignment can be imported 1:1. The user can therefore continue to work as usual after switching from an OP7 to an OP77B.

Deviations in key assignments

The following differences in key assignment are found after conversion of an OP7 project to an OP77B project.

- The <Shift+function key> combination is not supported. On the OP77B only the <F1> to <F4> and <K1> to <K4> can be assigned with calls of system functions.

- Setting the contrast with <Shift+ +/-> is not supported.

  The contrast can be set as follows on the OP77B:

  – In the loader menu.

  – With the preconfigured screen "P_LANGUAGE_SCREEN", which contains the operator control elements for setting the contrast and the language. The screen is created during migration.

  – With operator control elements on which the "AdjustContrast" system function can be configured.

Migration of the system keyboard image

The system keyboard image cannot be migrated, because WinCC flexible does not have a corresponding area pointer.

However, you can map the functionality in a reworking step, but without being able to assign a new function to all of the system keys (ACK, Enter, for example.)

If you want to evaluate keyboard operation in the PLC, assign the system function "SetBitWhileKeyPressed" to the "Press" event of all system keys. When the user presses a key on the HMI in Runtime, the system sets the corresponding bit in the external tag. This returns the PLC the information that the key is pressed. The external tags are written to the same PLC address as the area pointers in the ProTool project. Hence, the PLC program does not require any changes.

Migration of the function keyboard image

OP 77B does not support the "function keyboard image" area pointer. However, you can map the functionality in a reworking step by reconfiguring the key assignments.

If you want to evaluate keyboard operation in the PLC, configure the system function "SetBitWhileKeyPressed" to the "Press" event of all function keys. When the user presses a key on the HMI in Runtime, the system sets the corresponding bit in the external tag. This returns the PLC the information that the key is pressed. The external tags are written to the same PLC address as the area pointers in the ProTool project. Hence, the PLC program does not require any changes.

Migration of the property "Bit in tag"

If the property "Bit in tag" is configured for a function key or system key in the OP7 project, it is replaced during migration to an OP77B project. The system function "SetBitWhileKeyPressed" is configured during migration for all function and system keys in the OP7 project with property.

5.3 Tags and area pointers

5.3.1 Tags

Introduction

During migration tags corresponding to the tags in the OP7 project are created in the OP77B project. In addition, a connection to the PLC is created for external tags.

Conversion of data types

For converting a OP7 project in an OP77B project, the "STRING" data type is converted to the "STRINGCHAR" data type.

Array tags of the "STRING" type are not supported on the OP77B. Every array tag is replaced during migration by the creation of a special tag of the "STRINGCHAR" type.

Importing decimal places

Decimal places cannot be configured for a new tag on the OP 77B. Tags that have decimal places in the OP7 project are converted as follows:

- Tags that are configured to an IO field retain decimal places.

- Recipes take the decimal places of tags in the recipe display.

- The decimal places of tags that are displayed in the alarm text are discarded in alarms.

5.3.2 Area pointer

Introduction

Some area pointers are converted 1:1 during migration from an OP7 to an OP 77B.

The following area pointers are replaced by array tags:

- Process events
- Error alarms
- OP acknowledgment
- PLC acknowledgment
- LED mapping

Structural changes are made to the remaining area pointers.

Converting area pointers

The following area pointers are imported 1:1:

Area pointers on the OP7	Area pointer on the OP 77B
User version	Project ID
Screen number	Screen number

The following area pointers are not applicable or have another structure:

Area pointers on the OP7	Area pointer on the OP 77B
Process events	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Process events" with a 16-bit array element and a corresponding name. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers. The bit numbers automatically receive the update identifiers 1-8.
	As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." In reverse the alarm is interpreted as "sent" after the bit in the array element is reset to "FALSE" on the PLC.

Area pointers on the OP7	Area pointer on the OP 77B
Data mailbox	Data set
	This area pointer is always 5 words in size on the OP 77B. Data words 6 to n are not required on the OP 77B because data records cannot be transferred to the PLC indirectly via the "Data record" area pointer.
	This area pointer is automatically converted during conversion.
	The length of the area pointer remains the same when connected with an S7 PLC; the structure changes as follows:
	1. Word: Recipe number (unchanged)
	2. Word: Record number (previously 4th word)
	3. Word: Reserved
	4. Word: Record acknowledgment (previously byte n+3 of the interface range)
	5. Word: Free
Function keyboard	Area pointer discarded
	Substitute: Configure an external tag with its length exactly the same as the number of function keys so the tag has one bit for every function key. Configure the "SetBitWhileKeyPressed" system function to the previously assigned function key. Set the parameters of the system function so the key is set to the corresponding bit of the external tag. The status of the tags can then be evaluated in the PLC program.
OP acknowledgment	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Acknowledgement _OP" with a 16-bit array element and a corresponding name. Following conversion, there is one array element of the newly created array tags for every bit of the ProTool area pointer.
PLC acknowledgment	Area pointer discarded.
	Substitute: The "PT_Error alarms" array tag for error alarms is extended during migration.
Recipe box (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Recipe continuation box (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Recipe number (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Interface area	The data are exchanged with the PLC via the following area pointers:
	• "Coordination"
	• "PLC job"
	• "Date/time"
	• "Conversion"
	Once migration is complete, you must create the area pointer in the project.
	This distribution simplifies data exchange between HMI device and PLC. The PLC program must be configured accordingly after conversion.

Area pointers on the OP7	Area pointer on the OP 77B
Error alarms	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Error alarms" with a 16-bit array element and a corresponding name. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers. The array elements automatically receive the update identifiers 9-16.
	As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned warning alarm as "received." In reverse the alarm is interpreted as "sent" after a bit in the array element is reset to "FALSE" on the PLC.
System keyboard	Area pointer discarded. However, you can map the functionality in a reworking step, but without being able to assign a new function to all of the system keys (ACK, Enter, for example.)
	Substitute: Configure an external tag with its length exactly the same as the number of system keys so the tag has one bit for every system key. Configure the "SetBitWhileKeyPressed" system function to the previously assigned function key. Set the parameters of the system function so the key is set to the corresponding bit of the external tag. The status of the tags can then be evaluated in the PLC program.
LED mapping	Area pointer discarded.
	The conversion program replaces the area pointer with a predefined external array tag template "PT_LED mapping". The LEDs on the HMI device are assigned to the array elements of the newly created array tags in the same sequence as in the OP7 area pointer. The array elements automatically receive the update identifiers 17-24.
	Every array tag is two bits long. The following states can be implemented by setting the bits belonging to an LED in runtime:
	LED function (Bit1/Bit2)
	Off (0/0)
	Slow flashing (1/0)
	Rapid flashing (0/1)
	Steady (1/1)

5.3.3 Area pointer "Interface area"

Introduction

During conversion to WinCC flexible, the ProTool "interface range" area pointer is automatically distributed to three area pointers:

- "Coordination"
- "PLC job"
- "Date/time"

When migration is completed, you must create those area pointers in the project.

This distribution simplifies data exchange between the HMI and the PLC. The PLC program must be adapted accordingly after conversion.

Distribution to S7 PLCs

The interface range is distributed as follows for S7 PLCs:

ProTool	WinCC flexible
Interface range n+0 to n+31	Coordination n+0 to n+1
	PLC job n+4 to n+11
	Date/time n+15 to n+26

Handling of the start bit (coordination area pointer)

The coordination area pointer contains the start bit. If the value of this bit is read out in the PLC program, you can see whether the HMI device has started up from the PLC. The start bit is temporarily set to "0" by the HMI device during the startup process. As soon as the boot process is complete, the start bit is set to "1".

Special features in SIMATIC S5

The interface DB in the ProTool project can be reused after migration.

Example:

Area pointer	AS of group 1	AS of group 2
Job mailbox	DW32 to DW35	DW32 to DW35
Date/time	DW42 to DW47	DW66 to DW71

For AS of group 2: WinCC flexible HMIs support only one job mailbox. The PLC job with its three parameters is written directly to this mailbox.

5.4 System functions, events, and PLC jobs

Conversion

During migration from an OP7 to an OP 77B the ProTool system functions of the OP7 project are replaced by the corresponding WinCC flexible system functions.

In WinCC flexible, the names of the system functions are assigned according to a specific convention. For this reason the names of the WinCC flexible system functions are not identical to same as the names of the corresponding ProTool system functions. The system functions groups in WinCC flexible also have different names from those in ProTool.

The system functions are converted as follows:

OP7 project: ProTool system function	OP 77B project: WinCC flexible system functions
Calculation	
Conversion linear 1	Replaced by the "LinearScaling" system function.
	Note: Change the converted functionality so that the "x" and "y" tags are not identical.
Conversion linear 2	Replaced by the "InvertLinearScaling" system function.
	Note: Change the converted functionality to the effect that the "x" and "y" tags are not identical.
Tag: Convert value	Replaced by the "LinearScaling" system function.
Tag: Set value	Replaced by the "SetValue" function.
Tag: Value minus/plus	Replaced by the "IncreaseValue" and "DecreaseValue" system functions.
Tag: Value minus/plus - cursor	Replaced by the "IncreaseFocusedValue" and "DecreaseFocusedValue" system functions.
Data set	
Data record: OP -> PLC	Replaced by the "SetDataRecordToPLC" system function.
Data record: PLC > OP	Replaced by the "GetDataRecordFromPLC" system function.
Delete record	Replaced by the "DeleteDataRecord" system function.
Recipe table of contents	Replacement as follows depending on configured parameter:
	• Parameter "2 = edit"
	Replaced by the "ActivateScreen" system function which can be used to call a screen with a recipe view. The data records can be edited in the recipe view.
	• Parameter "5 = print"
	No longer supported; recipe data records can be printed with the aid of a recipe report.
	• Parameter "7 = transfer"
	Replaced by the "ActivateScreen" system function which can be used to call a screen with a recipe view. The data records can be transferred to the PLC using the recipe view.
Call data record	No longer required because data records are viewed and edited with the recipe view.
Alarms	
Clear error alarm buffer	The migration program creates the "ClearAlarmBuffer" function with error alarm parameters.
First/last error alarms	Not supported; the sequence in which error alarms are displayed can be set during configuration of the alarm view. In runtime the sequence cannot be changed.
Error alarm overflow warning	Not supported A job can be assigned to the "Alarm buffer overflow" event as a substitute for the overflow warning. The job is configured in the scheduler. A system alarm can also be output for the event.
Clear event buffer	The migration program creates the "ClearAlarmBuffer" function with warning alarm parameters.

OP7 project: ProTool system function	OP 77B project: WinCC flexible system functions
Alarms	
Event overflow warning	Not supported A job can be assigned to the "Alarm buffer overflow" event as a substitute for the overflow warning. The job is configured in the job planner. A system alarm can also be output for the event.
Alarm report ON/OFF	Replaced by the "SetAlarmReportMode" system function.
FM functions, MCU functions, NC functions	Not supported.
Password	
Password logout	Replaced by the "Logoff" system function.
Print	
Print error alarm buffer	Replaced by the "PrintReport" system function which can be used in migration to print an alarm report for the "Error alarms" alarm class.
Print event buffer	Replaced by the "PrintReport" system function which can be used in migration to print an alarm report for the "Warnings" alarm class.
Print all data records	Replaced by the "PrintReport" system function which can be used in migration to print a recipe report of all records.
Print data record	Replaced by the "PrintReport" system function which can be used in migration to print a recipe report of a record with a specified record number.
Hardcopy	Replaced by the "PrintScreen" system function.
Print all screens	Not supported If you want to print all screens, copy them into a report and print the report.
Screen table of contents	No longer required.
Screens	
Screen branching	No longer required; navigation among the screens created from the screen entries during migration is implemented with the "ActivateScreen" system function.
Special screen selection	No longer required.
Display selection	Replaced by the "ActivateScreen" system function.
Screen table of contents	No longer required.
Refresh screen	No longer required.
System settings	
S7 backup	Not supported
Toggle	
S7 change operating status	Not supported
S7 change PLC	Replaced by the "ChangeConnection" system function.
Language switching	Replaced by the "SetLanguage" system function.
Alarm level	Replaced by the "ActivateScreen" system function which can be used to call a screen with an alarm view.
Changing the operating mode	Replaced by the "SetDeviceMode" system function.

5.4.1 Events

Conversion

Almost all events are imported without change during migration from OP7 to OP 77B. The following events are replaced during migration:

- "Input value"

  The "Input value" event is replaced by the "Change value" event during migration.

  If you want to trigger an event with an input at the HMI device (regardless of whether the value of the tag changes), proceed as follows. Configure an additional system function of the "Bit processing" group to the <Enter> key (with which the operator completes the input) to set the bit and evaluate the set bit.

- "Output value"

  The "Output value" event is replaced by the "Change value" event during migration. This event is always triggered when the value of the tag changes, regardless of whether the change occurs in the PLC or by input at the HMI device.

5.4.2 PLC jobs

Introduction

Only some of the PLC jobs of an OP7 project can be used in an OP 77B project after conversion. Because of the different concepts of ProTool and WinCC flexible some PLC jobs are no longer required or can no longer be used. In some circumstances the OP 77B project and the PLC program must be edited to achieve the functionality of the OP7 project.

If the converted project requires editing in any case, check whether the tasks done with PLC jobs in the OP7 project can be done more efficiently with system functions in the OP 77B project.

Conversion

The PLC jobs are converted as follows during migration from OP7 to OP 77B:

OP7	OP 77B	Explanation/replacement/workaround
3 "Hardcopy"	Not supported	Instead configure the "PrintScreen" system function to the "Value change" event of a tag with PLC connection.
5 "Select table of contents"	No longer required.	
7 "Print all screens"	Not supported	If you want to print all screens, copy them into a report. You can then configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection.
10 "Print recipe with all data records"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. Recipe data records can be printed with the aid of a "Print recipe" object.

OP7	OP 77B	Explanation/replacement/workaround
11 "Special screen selection"	No longer required.	To select screens use either PLC job 51 "Screen selection" or configure the "ActivateScreen" system function.
		You can use system function to implement other functions out of special screens, "AdjustContrast" to alter the contrast, for example.
12 "Switch alarm logging on/off"	Not supported	Instead configure the "SetAlarmReportMode" system function to the "Value change" event of a tag with PLC connection.
13 "Switch language"	Not supported	Instead configure the "SetLanguage" system function to the "Value change" event of a tag with PLC connection.
14 "Set time (BCD-coded)"	14 "Set time (BCD-coded)"	---
15 "Set date (BCD-coded)"	15 "Set date (BCD-coded)"	---
16 "Internal interface parameters (RS232)"	Not supported	Make the communication settings directly at the OP 77B with the loader menu.
17 "Module interface parameters (TTY)"	Not supported	Make the communication settings directly at the OP 77B with the loader menu.
19 "Printer parameters"	Not supported	Set the printer parameters directly at the OP 77B with the loader menu.
21 "Error alarm display type"	Not supported	In WinCC flexible, alarms are displayed with the "Alarm display" screen object or with the "Alarm window" screen object.
		The filter criteria for the alarms can only be set in the properties view of the screen object in the "General" group. Filtering by "Priority," "Acknowledgment group" and"Status" is not supported.
22 "Set display contrast"	Not supported	Instead configure the "AdjustContrast" system function to the "Value change" event of a tag with PLC connection.
		Otherwise set the preconfigured default screen "P_LANGUAGE_SCREEN." "P_LANGUAGE_SCREEN" contains three buttons, labeled accordingly.
		• Button 1: Set language ("SetLanguage" system function)
		• Button 2: Increase contrast ("AdjustContrast(1)" system function)
		• Button 3: Reduce contrast ("AdjustContrast(0)" system function)
23 "Set password level"	23 "Set user group"	The PLC job "23 Set user group" takes the changed concept of user administration in WinCC flexible into account because it sets the user group instead of the password level. If you use PLC job 23, the PLC program must be modified accordingly.
24 "Password logout"	24 "Password logout"	---

OP7	OP 77B	Explanation/replacement/workaround
31 "Print error alarm buffer"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. This function can be used to print an alarm report for the alarm class "Error alarms."
32 "Print event buffer"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. This function can be used to print an alarm report for the alarm class "Warning alarms."
37 "Switch overflow warning for warning alarms on/off" 38 "Switch overflow warning for error alarms on/off"	PLC job is not required.	There is no buffer overflow warning in WinCC flexible. Configure the "Alarm buffer overflow" event as a replacement for the system function "ShowSystemAlarm." You can output your own alarm text with this system function. The alarm text is restricted to one language.
41 "Transfer date/time to PLC"	41 "Transfer date/time to PLC"	---
42 "Get LED range from the PLC" 43 "Get event message range from the PLC" 44 "Get alarm message range from the PLC" 45 "Get acknowledgment range from the PLC" 47 "transfer LED range directly to the HMI device"	PLC jobs are not supported.	Instead use PLC job "46 update tag" with appropriate parameters. This PLC job reads the values of WinCC flexible tags that replace the ProTool area pointers "event messages," "alarm messages,", "PLC acknowledgment" and "LED mapping" after migration. Control which tag is read with the "Update identifier" parameter. Assign the update identifier of the desired tag to this parameter. Assignment of the parameter "update identifier": 1...8 warning alarms 9...16 error alarms 17...24 LED mapping
49 "Clear event buffer"	49 "Clear event buffer"	---
60 "Clear error alarm buffer"	60 "Clear error alarm buffer"	---
51 "Screen selection"	51 "Screen selection"	---
52 "Print out screen"	Not supported	Instead configure the "PrintScreen" system function to the "Value change" event of a tag with PLC connection.
53 "Recipe selection"	Not supported	Instead configure the "ActivateScreen" system function to the "Value change" event of a tag with PLC connection with the "P_RECIPE_SCREEN" parameter. The preconfigured screen "P_RECIPE_SCREEN" contains recipe view and was created during conversion.

OP7	OP 77B	Explanation/replacement/workaround
54 "Print out recipe"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. A recipe can be printed with a recipe report.
69 "Transfer recipe record from PLC to TP/OP"	69 "Read data record from PLC"	In WinCC flexible, "Recipe number" and "Record number" must be given as parameters. If you want to transfer the recipe name or record name from the PLC to the HMI device, use the "GetDataRecordFromPLC" system function instead of the PLC job 69.
70 "transfer recipe record from TP/OP to PLC"	70 "Write record to PLC"	In WinCC flexible, "Recipe number" and "Record number" must be given as parameters. If you want to transfer the recipe name or record name from the HMI device to the PLC, use the "WriteRecordTagToPLC" system function instead of the PLC job 70.
71 "Partial screen update"	No longer required.	---
72 "Cursor positioning in current screen"	Not supported	Use PLC job 51 "Screen selection" instead.
73 "Cursor positioning in current special screen"	Not supported	Use PLC job 51 "Screen selection" instead.
74 "Keyboard simulation"	Not supported	---

5.4.3 Replacing job mailboxes with system functions

Introduction

Some PLC jobs that can no longer be used with WinCC flexible HMI devices can be replaced by system functions.

If the WinCC flexible project requires editing for any reason, check whether the tasks that were done with PLC jobs in the ProTool project can be done more efficiently with system functions in WinCC flexible.

Configuring

If you want the PLC to trigger the execution of a system function configured on the HMI device, configure a tag with the property "Always update" in WinCC flexible. Configure the corresponding system function for the "Change value" event on this tag.

Execution

If the value of the tag changes in the PLC, the system function is executed on the HMI device as soon as it detects the change.

Example: Trigger system function "SetAlarmReportMode" from the PLC

The following configuration can be used to replace the PLC job 12 "Switch alarm logging on/off" used in the ProTool project with a system function in the WinCC flexible project.

In WinCC flexible

1. Create an external tag of the INTEGER type.

2. Select "Tag > General" to set the "continuous cyclic" acquisition cycle.

3. Configure the "SetAlarmReportMode" system function to the event "Change value." Input the tag itself as parameter.

On the PLC

1. Set the value of the tag in the PLC program to 0 (alarm logging off) or 1 (alarm logging on).

Result

The alarm logging is switched on or off depending on the value of the tag.

Introduction

During migration from an OP7 to an OP 77B alarms of all existing alarm classes are converted resulting in the following alarms classes continuing to be available:

- Warning alarms
- Error alarms
- System alarms
- Custom alarm classes

Tags and text lists that are included in the alarm text are imported during conversion.

Preconfigured screen replaces alarm level

The alarm level is not supported on the OP 77B. Instead, during migration a preconfigured screen called "P_MESSAGE_SCREEN" is created to show the alarms. This screen shows the alarms with the aid of the "Simple alarm display" screen object. The alarm display includes the system date and time. A display in the alarm text is therefore no longer required. The date/time display is removed from the alarm text during conversion. The alarm display shows the alarm number, alarm text and the abbreviation of the alarm class in a two-line view for every alarm.

The "P_MESSAGE_SCREEN" screen is opened in runtime with <Esc> or with the "ActivateScreen" system function, depending on the former implementation.

Conversion of display variations

The "Mixed" and "Separate" display variations are not supported. The display variations are converted as follows:

- Display variation "Separate"

  The "P_MESSAGE_SCREEN" contains two alarm views: one for display of the error alarms and one for display of the warning alarms and system alarms.

- Display variation "Mixed"

  The screen that replaces the alarm level contains one alarm view to display system alarms, error alarms and warning alarms together. They are displayed in chronological order of receipt. Sorting the alarms can be set during configuration and cannot be changed in runtime.

  The screen with the alarm display is opened with a function key or with the "ActivateScreen" system function after conversion, depending on the former implementation of calling the alarm level.

Rest alarm

The rest alarm is not supported.

Priority of alarms

In the OP 77B priorities can only be assigned for alarm classes, not for single alarms. Priorities that were assigned to single alarms in the OP7 are discarded after conversion.

Conversion of the alarm area pointers

During conversion of an OP7 project to an OP 77B project, array tags with a 16-bit array element are created to replace the area pointers "event messages" and "alarm messages." Corresponding names are given to the array tags. Following conversion, there is one bit in the array element of the newly created array tags for every bit of the ProTool area pointer. The alarms are assigned to the bit numbers in increasing order of alarm numbers.

As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." Vice versa, the HMI interprets the alarm as "outgoing" when the bit in the array element is reset to "FALSE" on the PLC.

Alarm buffer overflow

The "Alarm buffer overflow" event is triggered when the alarm buffer is completely filled. The overflow of the alarm buffer cannot be queried separately for system alarms and warning alarms. The "ClearAlarmBuffer" can be used to delete alarms of specific classes from the alarm buffer and thus empty the alarm buffer when it overflows.

Alarm logging

The alarm log with printing of alarms on receipt and transmission is retained after conversion.

Printing headers and footers with the alarm log is not supported.

5.5 User authorizations

New user authorization concept in WinCC flexible

The concept of user authorization in WinCC flexible differs from that in ProTool.

In WinCC flexible, each users is assigned only to one user group. You assign those user groups characteristic access rights, namely the authorizations. User groups are not organized in a hierarchical structure.

A default "Administrator" user group is created in each project. Users assigned to this group in Runtime have all rights, and may also set up new users.

You determine what rights the user requires for efficient operation of the various control elements.

In Runtime, all users are identified by their user name and password.

Migration of the ProTool password level

During conversion of a ProTool project to a WinCC flexible project, the program creates up to nine user groups. One user group is created for each password level used in the ProTool project.

The "Administrators" user group is always created. All user authorizations for which password level 9 was valid in ProTool are assigned to this "Administrator" group. All other user authorizations that were assigned the same password level in ProTool are also assigned to the same user group in the WinCC flexible project. The user groups are assigned the authorizations as in the password levels of the ProTool project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Advanced options in WinCC flexible

In WinCC flexible, users may be assigned to any user group, without making any allowances for hierarchical structures. During conversion, check the project to see whether to create new user groups or not.

5.5.1 Adapting user authorizations

Introduction

During conversion of an OP7 project to an OP 77B project, user groups are created that have the same authorization that were assigned to the password levels in the OP7 project.

In addition, the various control elements are assigned the same access protection as in ProTool after migration.

Password list is not migrated

The password list is not migrated because it is saved on the specific HMI device and is not in the project data.

After migration, the users must be created again. There are two options for this on the OP 77B:

- New users, user groups and passwords are created during configuration with the runtime user administration.

- A screen with a user view is configured. The user view shows existing users, passwords and user groups to an administrator. An administrator can also create new users, user groups and passwords with the user view.

Conversion of default screen for password editing

The default screens for working with passwords are converted as follows:

Default screen in the OP7	Replacement on the OP 77B
Password processing - login	Replaced by calling the "ShowLogonDialog" system function.
Password processing - logout	Replaced by calling the "Logoff" system function.
Password processing - edit	Replaced by the preconfigured screen "P_USER_SCREEN". "P_USER_SCREEN" contains a user display.

Navigation on the OP 77B

On the OP 77B users can be assigned to a user group as desired without worrying about hierarchies. During conversion, check the project to see whether new user groups should be created. In addition, appropriate names for the user groups created during migration can be assigned in the "Groups" editor in the "User administration runtime" group.

5.6 Print

Introduction

The printing of project data in WinCC flexible is controlled by reports. Reports contain objects that enable screens, the alarm buffer and recipes to be printed.

The user of an OP 77B can only print the alarm log and the currently displayed screen (hardcopy) without requiring a report.

Headers and footers are by default not printed on the OP 77B.

Alarm logging

During migration of OP7 to OP 77B the settings for the alarm logging are retained 1:1. During migration the corresponding OP7 system function is replaced by the OP 77B "SetAlarmReportMode" system function.

Printing on alarm buffer overflow

If "Logging on overflow" is configured in the OP7 project, use the following configuration to set the same behavior in the OP 77B project:

1. Create a report with the "Print alarm" object.

2. In the properties view of the alarm report select "Alarm events" as "Source for alarms."

3. In the scheduler create a job that will be triggered by the "Alarm buffer overflow" system event.

4. Assign the "PrintReport" system function to the job with the report and alarm log to be printed.

The alarm log is printed when the alarm buffer overflows.

Printing the alarm buffer

On the OP7, system functions control the printing of the event buffer and the error alarm buffer. During migration from an OP7 to an OP 77B, this functionality is converted as follows:

Two reports are generated with a "Print report" object ("Alarm log"). The reports are configured so that the warning alarms or error alarms are printed from the alarm buffer. The corresponding system functions of the OP7 project are replaced by the "PrintReport" system function for printing the desired report.

The alarms can be printed in ascending or descending chronological order. They cannot be sorted by alarm number.

Printing the displayed screen (hardcopy)

When an OP7 project that has the option of printing the displayed screen (hardcopy) is converted, the OP 77B project retains this function. During migration the corresponding OP7 system function is replaced by the OP 77B "PrintScreen" system function.

Printing screens

Printing all screens in one print job is not supported on the OP 77B. Use the project documentation to document screens in a project. The project documentation used to output selected or all configuration data of a project.

If you want to allow the user to print screens in runtime even if they are not displayed, proceed as follows:

1. Create a report for every screen by configuring all objects from the screen. Objects that cannot be configured can also not be printed, e.g. bars.

2. Configure an IO field so the user can select one of the screens from a text list. Configure a tag for the IO field to save the result of the selection.

3. Configure the "PrintReport" system function to a button. Send the value of the tag as a parameter.

In runtime the user can select a screen from the text list and print with the button.

If you have used printouts of screens for the project documentation, call the menu command "File > Print Project Documentation".

Printing recipes

On the OP7 printing of all or one specific data record of a recipe is controlled by system functions. During migration from an Op7 to an OP 77B this function is converted as follows:

Two reports are generated with a "Print recipe" object ("Recipe log"). The recipe reports are configured so that all data records are printed with one recipe report and the currently selected data record is printed with the other recipe report. The corresponding system functions of the OP7 project are replaced by the "PrintReport" system function for printing the desired report.

Printer setting

On the OP 77B a connected printer is configured via the loader menu. Printer configuration with system functions is not supported. Detailed information about the loader menu can be found in the operating instructions of the OP 77B..

Introduction

During migration from an OP7 project to an OP 77B project the recipes are also converted.

Converting the recipes

In detail the recipes are converted as follows:

- The recipe header becomes the localized recipe name.

- The recipe number is imported unchanged.

- The recipe entries and the associated tags are converted 1:1. The names of the recipe entries are formed from the total text of the recipe entry. The tag value on the OP 77B will always be displayed after the name of the recipe entry.

  Example (display of the tag value is shown by <...>):

Recipe entry on the OP7	Recipe entry on the OP 77B
Sugar <1.5> kg	Sugar (kg) <1.5>
Water <5> l	Water (l) <5>

WinCC flexible does not support password protection for separate recipe entries. Instead, configure access control by assigning a password when you call the P_RECIPE_SCREEN screen.

Conversion of data records

The data records of the recipes cannot be converted. They must be input again. The data records can be created in the "Recipes" editor on the configuration computer and then sent to the OP 77B.

Display of recipes

The OP 77B has the "Recipe display" screen object for displaying the recipes. The data records of a recipe can be displayed, edited and sent to the PLC with this display on the HMI device.

During migration a screen with a recipe view is created instead of the recipe table of contents and the individual recipes. The recipe view is automatically configured in accordance with the recipes existing in the OP7 project.

The "Flashing" property is not supported for the names of the recipe elements.

Area pointer for transfer of data records

The area pointer used for transferring data records of a recipe depends on the PLC to which the OP 77B is linked.

- Connection to the S7

  On connection to the S7 the "Data record" area pointer is used to transfer data records of a recipe.

  After migration from the OP7 to the OP 77B the length of the data record remains the same, and the structure is converted as follows:

  1. Word: Recipe number (unchanged)

  2. Word: Record number (previously 4th word)

  3. Word: Reserved

  4. Word: Record acknowledgment (previously byte n+3 of the interface range)

  5. Word: Free

- Connection to the S5

  On connection to the S5 the transfer of the data records is converted from indirect to direct transfer. The recipe number box, the recipe box and the recipe subsequent box are discarded.

5.7 PLC driver

1:1 conversion

Most PLC drivers available for OP7 are also available for the OP 77B. They are automatically converted during conversion. This applies to the following PLC drivers:

- SIMATIC S7-300/400
- SIMATIC S7-200 (one PLC)
- Allen-Bradley DF1 PLC-5
- Allen-Bradley DF1 SLC 500
- Allen-Bradley DH485
- GE Fanuc SNP
- MITSUBISHI FX
- Modicon Modbus
- Omron Host-Link/Multilink

Project with connection to the SIMATIC S5-AS511

The conversion is run automatically. After conversion, the standard FB (FB51 must be deleted from the PLC program).

A converter from V.24 to TTY is required to connect the OP77 (MLFB no.: 6ES5 734-1BD20).

Project with connection to the Automation S5-FAP

If the SIMATIC S5 FAP PLC was used in the OP7 project, a SIMATIC S5 AS511 must be selected in ProTool before migration. After migration, you can change to a SIMATIC S5 DP in WinCC flexible.

Procedure:

1. Open the OP7 project in ProTool.
2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 FAP" protocol is now shown on the right.
3. Double-click on the input PLC to change the PLC protocol.
4. Select the SIMATIC S5 AS511 PLC protocol from the "PLC" dialog. Confirm with "OK".
5. Save the project.
6. Close ProTool.
7. Open the OP7 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."
8. Select the reference language.

The project is automatically converted to an OP 77B project and then opened in WinCC flexible.

It is now connected to the SIMATIC S5 AS511.

To change to a SIMATIC S5 DP:

1. Double-click the "Connections" entry in the project view for the OP77B. The "Connections" editor opens.

2. Select "SIMATIC S5 DP" as the connection.

3. Replace the default FB (FB 52 or FB 53) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

Project with connection to the SIMATIC S5-L2-DP

If the SIMATIC S5-L2-DP PLC was used in the OP7 project, the SIMATIC S5 DP PLC must be used on the OP 77B. Before migration, switch to a SIMATIC S5 AS511 in ProTool. After migration, change to a SIMATIC S5 DP in WinCC flexible.

Procedure:

1. Open the OP7 project in ProTool.

2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 L2 DP" protocol is now shown on the right.

3. Double-click on the input PLC to change the PLC protocol.

4. Select SIMATIC S5 AS511 as PLC protocol in the "PLC" dialog and confirm with "OK".

5. Save the project.

6. Close ProTool.

7. Open the OP7 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."

8. Select the reference language.

    The project is automatically converted to an OP 77B project and then opened in WinCC flexible.

    It is now connected to the SIMATIC S5 AS511.

9. Double-click the "Connections" entry in the project view for the OP 77B. The "Connections" editor opens.

10. Select "SIMATIC S5 DP" as the connection.

11. Replace the default FB (FB 58) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

Project with two SIMATIC S7-200 PLCs

With OP7 projects it is possible to configure multiple PLCs with the SIMATIC S7 200 PLC driver protocol when the PPI profile is used.

On the OP 77B, only two PLCs can be coupled with the MPI communication network. The PLC profile must be changed in the OP7 project before conversion to ensure a successful conversion.

Procedure:

1. Open the OP7 project in ProTool.

2. From the project view select the "PLC" object. The two PLCs with the SIMATIC S7-200 protocol are now displayed on the right.

3. Double-click on the first PLC to open the "PLC" dialog box.

4. Click on "Parameter" and select the entry "MPI" as "Profile." Confirm the setting with "OK."

5. Save the project.

6. Close ProTool.

7. Open the OP7 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."

8. Select the reference language.

   The project is automatically converted to an OP 77B project and then opened in WinCC flexible.

Project with connection to the Omron Host-Link/Multilink

If the Omron Host-Link/Multilink PL was used in the OP7 project, the specified station address is set to 0 during conversion to an OP 77B project. After conversion, the correct station address is set in the OP 77B project as follows:

1. Click in the project window on "Communication"> "Connections."

2. Select the entry "Omron Host-Link/Multilink" in the "Communications driver" column.

3. Select the "Parameter" entry in the properties view.

4. Enter the correct station address.

Non-supported PLCs

OP 77B does not support the "Telemecanique Adjust" and "SIMATIC S7-NC" PLCs.

If you convert a project that has a connection to the "Telemecanique Adjust" or "SIMATIC S7-NC" PLC, a warning is shown during conversion. The "SIMATIC S7-300/400" PLC driver is set on the OP 77B project as a substitute. All tags will be separated from the PLC.

Migration from OP17 to OP 177B

6

6.1 Basic principles

Introduction

WinCC flexible does not support the OP17 HMI. If you want to continue to use ProTool projects created for this HMI device and migrate them, the migration program automatically changes to the OP 177BmonoDB HMI. In the following referred to as OP 177B.
WinCC flexible lets you change the HMI if you require a different version of OP 177B for your project.

If the ProTool project has more than one editing language, you will be prompted to select one of the runtime languages as the reference language when starting the conversion. The reference language influences the conversion of screens that consist of static texts and IO fields positioned between the texts.

Compared to OP17, the OP 177B is equipped with a larger display. The migration tool arranges the text elements and IO fields on the OP 177B display. You need to adapt the screens to the 6" display of OP 177B.

Conversion

OP17 and OP 177B HMI devices feature different functionality and function principles, i.e. the configuration requires certain changes during migration. Those changes are performed automatically for the most part during migration.

The fundamental changes cover the following areas:

- Screens
- Tags and area pointers
- System functions, events, and PLC jobs
- Alarm system
- User authorizations
- Print
- Recipes
- PLC driver
- Timer

6.2 Screens

6.2.1 Screen entries

Conversion

All screen in OP17 projects can be subdivided into up to 99 screen entries through which the user can scroll. During migration of an OP17 project with screen entries, every single screen entry is converted to an independent screen. The softkeys are also automatically assigned a ChangeScreens function during migration. The user can therefore scroll through the screens that were created from the screen entries of the same OP17 screen in the OP 177B as in the OP17 project.

Assignment of screen names and screen numbers

The screens created during migration that replace other screen entries are given screen names and screen numbers according to the following convention:

- The screen that was created to replace the first screen entry of an OP17 screen receives the name and number of the OP17 screen.

- All subsequent screen entries are given the name and screen number of the OP17 screen extended by one digit in ascending order from 02 to 99.

Screen series whose screens were created from the screen entries of the same OP17 screen can be identified after migration from the matching components of the new screen names and screen numbers.

Application example

The following screens and screen entries are contained in an OP17 project that you want to convert to an OP 177B project:

- "Screen" with screen number 1 divided into three screen entries

- "Record" with screen number 2 also divided into three screen entries

The following screen series are created during migration:

OP17 project, screen "Screen", screen number 1	Screen series of the OP 177B project
Screen entry 1	"Screen", screen number 1
Screen entry 2	"screen_2", screen number 102
Screen entry 3	"screen_3", screen number 103

OP17 project, "Record" screen, screen number 2	Screen series of the OP 177B project
Screen entry 1	Screen "Record", screen number 2
Screen entry 2	Screen "Record_3", screen number 202
Screen entry 3	Screen "Record_3", screen number 203
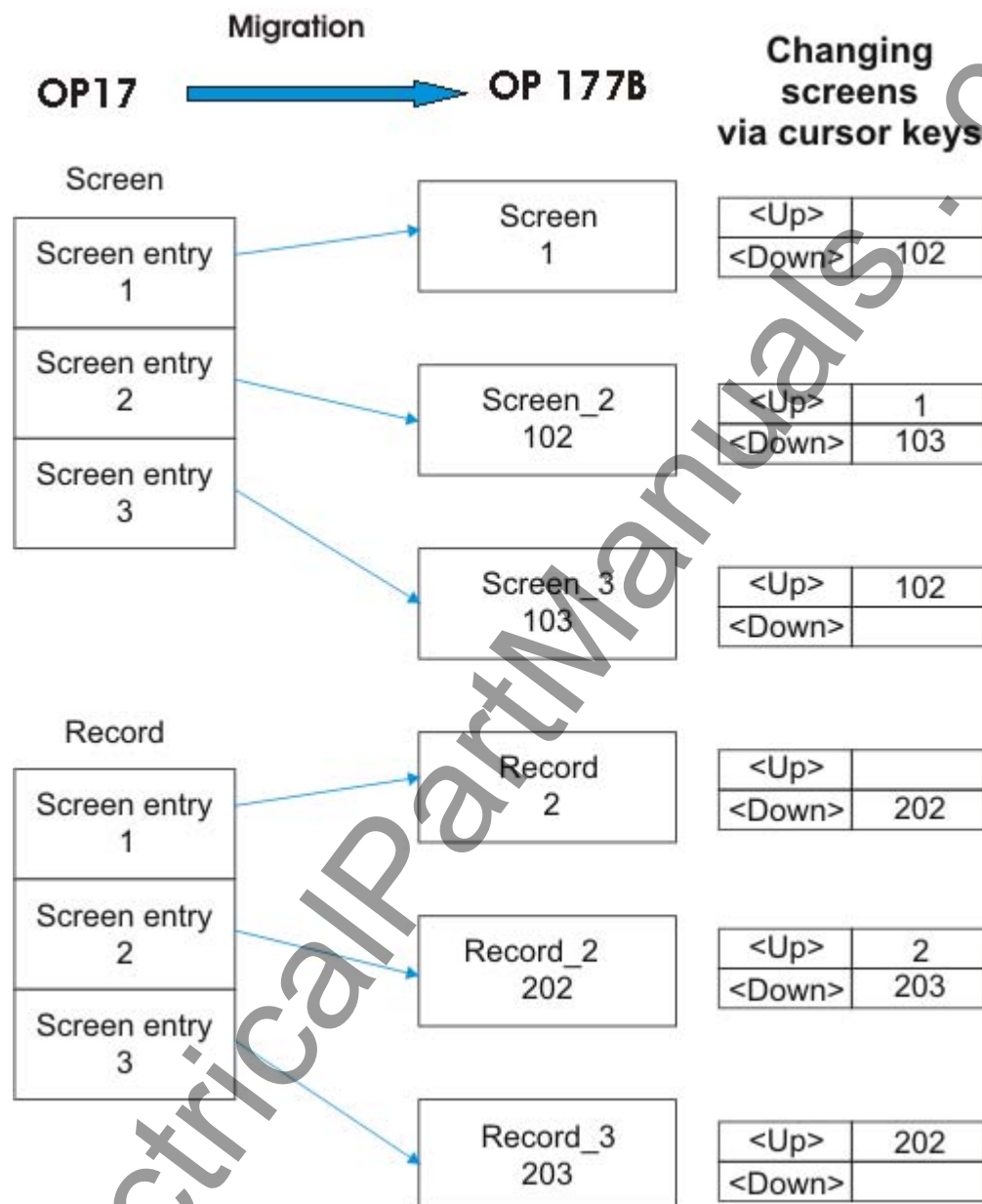
Conversion of screen entries of the OP17 to screens for OP 177B

Navigation

The migration program configures the softkeys <F1> , <F3> and <F5> so the user can navigate through a screens series as usual on OP17.

Screen title

OP 177B does not support the display of screen titles.

6.2.2 Navigation

Introduction

During migration of an OP17 project with screen entries every single screen entry is converted to an independent screen of the OP 177B project. Even after migration associated screen series whose screens come from the screen entries of the same OP17 screen can be identified. During conversion to an OP 177B project, the softkeys <F1>, <F3> and <F5> are assigned so the OP17 user can navigate through a screen series in the same way as on OP17.

Navigation on OP 177B

The following key assignments are set up in the OP 177B for navigation within a screen and to the other screens of a screen series:

- Softkeys <F1> and <F3>

Key	Among the screens of a series
<F1>	To the previous screen of the same series; at the top end of the first screen: unassigned
<F3>	To the next screen of the same series; at the bottom end of the last screen of a series: unassigned

- Softkey <F5>

  In the OP17 project, softkey <F5> will be assigned one of the following functions, based on the <Esc> key settings:

Function of <Esc> in the OP17 project	Functionality of <F5> in the OP 177B project
Change to screens list	Screens list no longer available. Instead, return to the previous screen (implemented using the system function "ActivatePreviousScreen")
Change to alarm level	Assignment with the system function "ActivateScreen"; triggers change to preconfigured screen "P_MESSAGE_SCREEN" with an alarm view. "P_MESSAGE_SCREEN" was created during conversion. Key assignment in "P_MESSAGE_SCREEN": • <F5> key with return to the calling screen (implemented using the "ActivatePreviousScreen" system function)
Change to recipe table of contents	Assignment with the system function "ActivateScreen"; triggers change to preconfigured screen "P_RECIPE_SCREEN" with a recipe display. "P_RECIPE_SCREEN" was created during conversion. Key assignment in "P_RECIPE_SCREEN": • <F5> key with return to the calling screen (implemented using the "ActivatePreviousScreen" system function)
Switch to previous screen	Return to the called screen (implemented with the system function "ActivatePreviousScreen") A return within a screen series causes the previous screen of the series to open. This is different from the behavior of the OP17 project in which it always jumps back to the first entry of the series.
Switch to a specified screen	Screen switch to a specified screen (implemented with the system function "ActivateScreen", parameter "Name of the specified screen")

6.2.3 Default screens

Introduction

During migration to the OP 177B project, new preconfigured screen, with the same functions are created to replace most of the default screens of the OP17 project. Some default screens do not require replacement or cannot be replaced because of the changed scope of function and changed functions of the OP 177B.

Conversion of default screens

The default screens of OP17 are converted as follows:

Default screen of OP17	Replacement on OP 177B
Error alarms - viewing	Converted to screen with alarm view The alarm view is configured so the error alarms are displayed from the error alarm buffer.
Error alarms - printing	Replaced by the "PrintReport" system function which is used to trigger printing of a report. The report is configured so the contents of the error alarm buffer are printed.
Error alarms - number	Discarded. All error alarms are shown in the alarm view that displays the error alarms.
Error alarms - delete	The function "ClearAlarmBufferProTool" is used for the migration. The parameters of this system function are identical to those of the ProTool function. Always use the system function "ClearAlarmBuffer" in new projects. This has new parameters.
Error alarms - overflow	Discarded.
Error alarms - texts	Discarded. The texts of the error alarms are also shown in the alarm view that displays the error alarms.
Warning alarms - appearance	Converted to screen with alarm view The alarm view is configured so the warning alarms are displayed from the event buffer.
Warning alarms - printing	Replaced by the "PrintReport" system function which is used to trigger printing of a report. The report is configured so the contents of the event buffer are printed.
Warning alarms - number	Discarded. All warning alarms are shown in the alarm view that displays the warning alarms.
Warning alarms - delete	The function "ClearAlarmBufferProTool" is used for the migration. The parameters of this system function are identical to those of the ProTool function. Always use the system function "ClearAlarmBuffer" in new projects. This has new parameters.
Warning alarms - overflow	Discarded.
Warning alarms - texts	Discarded. The texts of the warning alarms are shown in the alarm view that displays the warning alarms.
Screens - editing	Discarded. In Runtime screens can be opened directly and edited.

Default screen of OP17	Replacement on OP 177B
Screens - printing	Discarded. Printing of screens can be configured as follows: • With the "PrintScreen" system function to print the currently displayed screen • Configure the screen objects again in a report, and print the report using the "PrintReport" system function. Printing of all screens at once is not supported. If you want to configure reporting of process values from different screens, corresponding reports must be created in the project. The menu item "Project" > "Print project documentation" can be used to document the configuration phase of the project.
Data records - editing	Is replaced with the "P_RECIPE_SCREEN" screen template which has an advanced recipe view. On this screen, you can display and edited the data records of recipes, and transfer these to the HMI.
Data records - transfer	Is replaced with the "P_RECIPE_SCREEN" screen template which has an advanced recipe view. On this screen, you can display and edited the data records of recipes, and transfer these to the HMI.
Data records - printing	Discarded. Can be replaced by printing a report with the "Recipe print" object.
System settings - operation	Will be replaced with the "P_MODE_SCREEN" screen template with three accordingly labeled buttons. A different value of the "SetDeviceMode" system function depending on the operating mode is configured to every one of these buttons.
System settings - AlarmView	Not supported The sequence in which the alarms are to be displayed (oldest or newest first) can be specified when the alarm view is configured. This setting cannot be changed in runtime.
System settings - SysAlarm	Converted in a screen with an alarm view The alarm view is configured so the system alarms are displayed from the system alarm buffer.
System settings - languages	Will be replaced with the "P_LANGUAGE_SCREEN" screen template with three accordingly labeled buttons. • Button 1: Set language ("SetLanguage" system function) • Button 2: Increase contrast ("AdjustContrast(1)" system function) • Button 3: Reduce contrast ("AdjustContrast(0)" system function)
System settings - date/time	Will be replaced with the "P_DATE_TIME_SCREEN" screen template which contains two date-time fields with corresponding labels, one to show the date and the other to show the time. The day of the week cannot be input but is calculated internally from the date so it can be used in the job planner or PLC jobs. The date/time field of OP 177B is longer than that of OP17. After migration, adapt the screen configuration as required.
System settings - printer	No longer required. The printer parameters of OP 177B are configured in the Control Panel.
System settings - IF1A/RS 232 IF1A/TTY IF1B IF1A/B IF2	Not supported, because on the OP 177B the communication settings are made in the loader menu.

Default screen of OP17	Replacement on OP 177B
StatVAR	Replaced with the "P_STATUS_FORCE_SCREEN" screen template with Status/Force indication. The special screen selection function will be replaced by the "ActivateScreen" system function which is used to call a screen with Status/Force display. The <F2> key of this screen will be assigned a return to calling screen function. (implemented using the "ActivatePreviousScreen" system function.)
SteuVAR	Replaced with the "P_STATUS_FORCE_SCREEN" screen template with Status/Force indication. The special screen selection function will be replaced by the "ActivateScreen" system function which is used to call a screen with Status/Force display. The <F2> key of this screen will be assigned a return to calling screen function. (implemented using the "ActivatePreviousScreen" system function.)
Password processing - login	Replaced by calling the "ShowLogonDialog" system function.
Password processing - logout	Replaced by calling the "Logoff" system function.
Password processing - edit	Replaced by the preconfigured screen "P_USER_SCREEN". "P_USER_SCREEN". This screen contains the password view.

6.2.4 Screen objects

Introduction

During conversion of an OP17 project to an OP 177B project, the screen objects are automatically adapted to the greater extent to suit OP 177B functionality. However, certain editing work is required after a conversion.

Converting IO fields

During conversion, the output fields, input fields, and input/output field are converted to an IO field with a similar configuration. The IO fields of a converted project can be operated using the touch screen keyboard. OP 177B automatically opens this keyboard when you touch the IO field. Based on the configuration of the input object, the OP opens a touch screen keyboard only for the input of numeric or alphanumeric values.

The fields are converted in particular as follows:

OP17	OP 177B
Output field	IO field, configured as output field
Input field	IO field, configured as input field
Input/output field	IO field, configured as input/output field
Text output field	Symbolic IO field, configured as output field
Text input field	Symbolic IO field, configured as input field Operation on the touch screen.
Text input/output field	Symbolic IO field, configured as input/output field Operation on the touch screen.

Note

After migration, it is possible that the right-aligned characters arranged in symbolic I, O or I/O fields are only partially displayed on the HMI device display. The affected fields have to be enlarged in a post-editing step or the font used has to be decreased.

Converting text

The following changes and restrictions are involved when converting text:

- Font: The OP17 font is converted to the default OP 177B font. The font size is determined by the number of lines set on OP17. On OP 177B, a 10pt font is used for a display consisting of 8 lines with 40 char/line. On OP 177B, a 20pt font is used for a display consisting of 4 lines with 20 char/line.

- If you want to convert to the "Tahoma" font in the OP 177B project after conversion, the following characters are not supported, depending on the character set:

OP17 character set "European" and "EuropeanII"

OP17 character set "Russian"

OP17 character set "Czech", "Hungarian" and "Polish"

OP17 character set "Greek"

OP17 character set "Turkish"

Converting the layout of a screen entry

The conversion of an OP17 screen entry to an OP 177B screen includes the conversion of all static texts to a single, large text field. The layout of the IO fields will be adapted to suit the OP 177B display. Compared to OP17, the OP 177B is equipped with a larger display. You need to adapt the screens to the 6" display of OP 177B and the modified layout of softkeys.

An example of the adaptation of the screen layout is shown below.
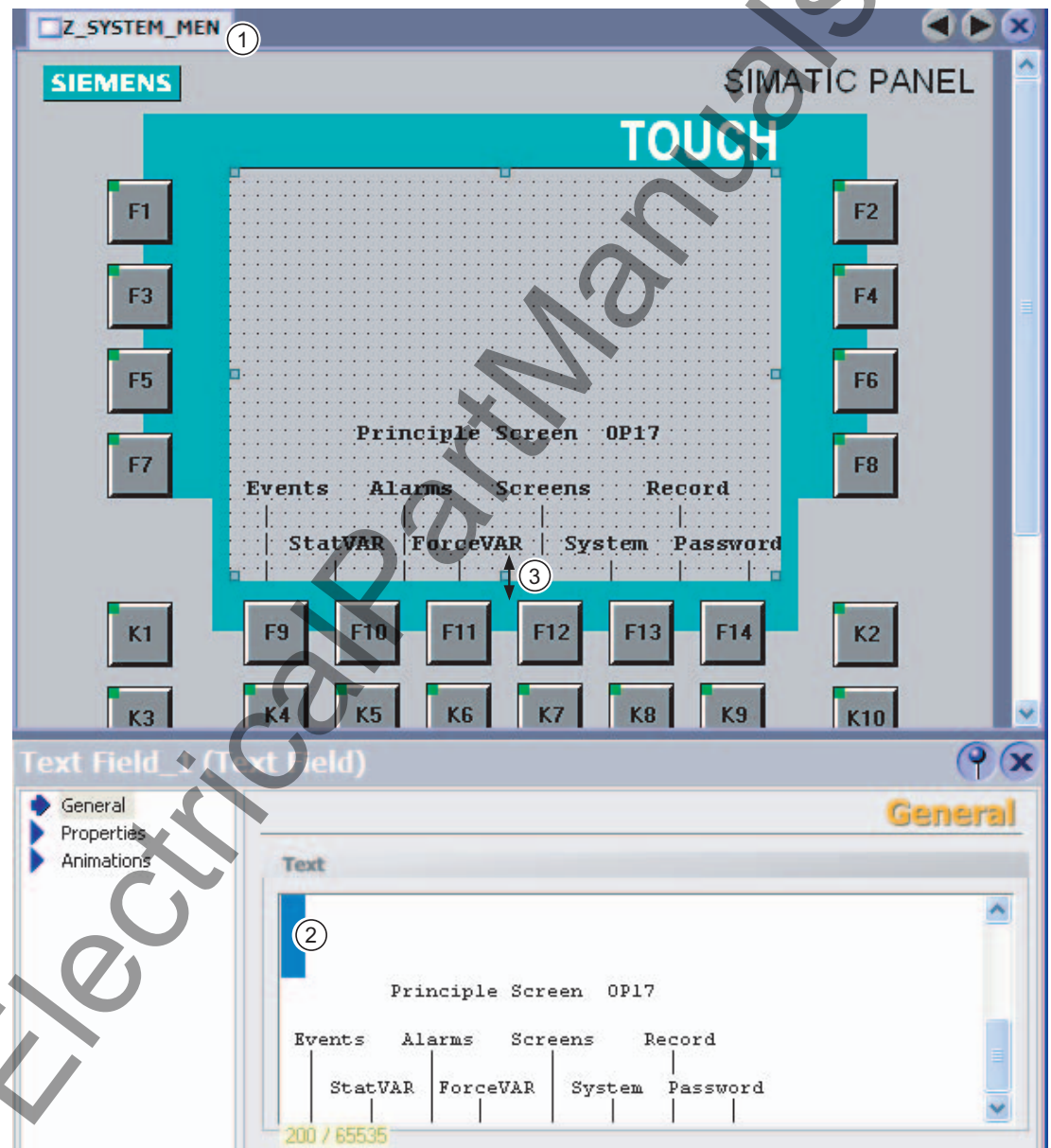
①   Select the screen to edit in WinCC flexible.

②   In the Screens Editor, select and delete lines which are empty and not required.

③ Reduce the size of the text field by dragging its handles from the bottom towards the top.
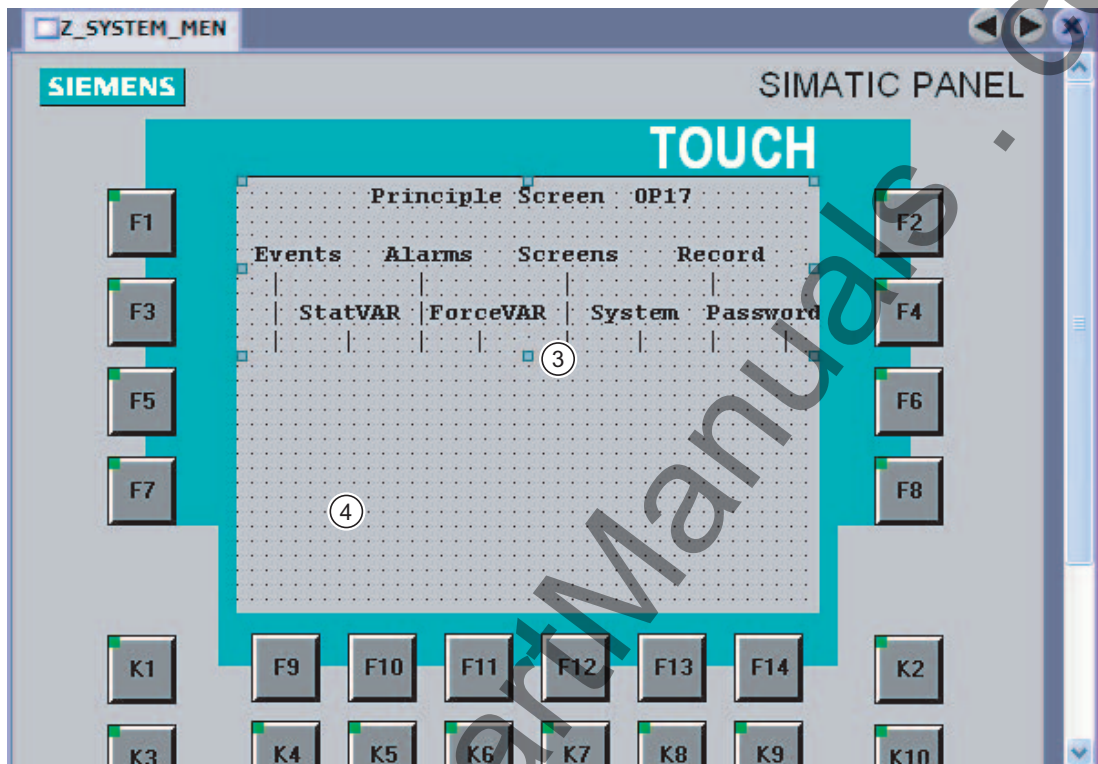
④ Use the cut/paste function to move text elements from the original text field to the bottom screen section; these fields can be positioned freely.

This is an easy means of adapting your text elements to the new design of OP 177B.

Flashing

The text property "Flashing" is only imported to the OP 177B during conversion if the attribute is assigned to the complete text of a screen object. Flashing text sections are no longer supported on the OP 177B. If a text field consists of several words and one of the words should flash, the text field must be separated into several text fields.

Tag list

The tag list is not supported on the OP 177B.

Screen object Status force

The Status/Force screen object is supported.

6.2.5 Softkeys

Introduction

The layout of softkeys differs between OP17 and OP 177B. When you migrate the project, the softkey assignments of the two softkey rows (K1 - K16) of OP17 on the bottom will be converted to the two bottom softkey rows (K3 - K18) of OP 177B.

In contrast to OP17, OP 177B is not equipped with any system keys (cursor keys, ACK, ESC, Help.) The system key functionality is assigned to the OP 177B softkeys.

Deviations in key assignments

The following differences in key assignment are found after conversion of an OP17 project to an OP 177B project:

OP17 key assignments	OP 177B key assignments
<Up>	<F1> key, implemented using the "ActivateScreen" system function.
<Down>	<F3> key, implemented using the "ActivateScreen" system function.
<ESC>	<F5> key
<ACK>	<K2> key, implemented using the "ACK" system function.
<Help>	<F1> key, implemented using the "ShowOperatorNotes" system function.
<F1>	<F7> key
<ENTER>	Not supported On OP 177B, input is confirmed using the <Enter> key of the touch screen keyboard.
<Shift + Function key>	The <Shift+function key> keystroke is not supported. On OP 177B, only the <F1> to <F14> and <K1> to <K10> keys can be assigned system function calls.
<Shift+ +/->	Setting the contrast using <Shift+ +/-> is not supported.
	The contrast can be set as follows on the OP 177B:
	• In the loader menu.
	• With the preconfigured screen "P_LANGUAGE_SCREEN", which contains the control elements for setting the contrast and the language. The screen is created during migration.
	• Using control elements assigned the "AdjustContrast" system function.

Migration of the system keyboard image

The system keyboard image is obsolete, because OP 177B is not equipped with system keys.

Migration of the function keyboard image

The function keyboard mapping cannot be migrated, because WinCC flexible does not provide a corresponding area pointer. However, you may create an image of this functionality in a subsequent step.

If you want to evaluate keyboard operation in the PLC, configure the system function "SetBitWhileKeyPressed" to the "Press" event of all function keys. When the user presses a key on the HMI in Runtime, the system sets the corresponding bit in the external tag. This gives the PLC the information that the key is pressed. The external tags are written to the same PLC address as the area pointers in the ProTool project. Hence, the PLC program does not require any changes.

Migration of the property "Bit in tag"

If the property "Bit in tag" is configured for a function key or system key in the OP17 project, it is replaced during migration to an OP 177B project. The system function "SetBitWhileKeyPressed" is configured during migration for all function and system keys in the OP17 project with property "Press".

6.3 Tags and area pointers

6.3.1 Tags

Introduction

During migration, tags corresponding to the tags in the OP17 project are created in the OP 177B project. In addition, a connection to the PLC is created for external tags.

Conversion of data types

The "STRING" data type is converted to the "STRINGCHAR" data type for conversion of an OP17 project to a OP 177B project.

OP 177B does not support array tags of the "STRING" type. Every array tag is replaced during migration by the creation of a special tag of the "STRINGCHAR" type.

Importing decimal places

Decimal places cannot be configured for a new tag on the OP 177B. Tags that have decimal places in the OP17 project are converted as follows:

- Tags that are configured to an IO field retain decimal places.
- Recipes apply the decimal places of tags in the recipe display.
- The decimal places of tags that are displayed in the alarm text are discarded in alarms.

6.3.2   Area pointer

Introduction

Certain area pointers are converted 1:1 during migration from an OP17 to an OP 177B.

The following area pointers are replaced by array tags:

- Warning alarms
- Error alarms
- OP acknowledgment
- PLC acknowledgment
- LED mapping

Structural changes are made to the remaining area pointers.

Converting area pointers

The following area pointers are imported 1:1:

Area pointers on OP17	Area pointers on OP 177B
User version	Project ID
Screen number	Screen number

The following area pointers are not applicable or have another structure:

Area pointers on OP17	Area pointers on OP 177B
Warning alarms	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Process events" with a 16-bit array element and a corresponding name. Following conversion, each bit of the OP17 area pointer corresponds with one bit in the array element of the new array tags. The alarms are assigned to the bit numbers in ascending order of alarm numbers. The bit numbers automatically receive the update identifiers 1-8.
	As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." In reverse the alarm is interpreted as "sent" after the bit in the array element is reset to "FALSE" on the PLC.
Data mailbox	Data set
	This area pointer is always 5 words in size on the OP 177B. Data words 6 to n are not required on the OP 177B because data records cannot be transferred to the PLC indirectly via the "Data record" area pointer.
	This area pointer is automatically converted during conversion.
	The length of the area pointer remains the same when connected with an S7 PLC; the structure changes as follows:
	1. Word: Recipe number (unchanged)
	2. Word: Record number (previously 4th word)
	3. Word: Reserved
	4. Word: Record acknowledgment (previously byte n+3 of the interface range)
	5. Word: Free

Area pointers on OP17	Area pointers on OP 177B
Function keyboard	Area pointer discarded
	Substitute: Configure an external tag with its length exactly the same as the number of function keys so the tag has one bit for every function key. Configure the "SetBitWhileKeyPressed" system function to the previously assigned function key. Set the parameters of the system function so the key is set to the corresponding bit of the external tag. The tag status can then be evaluated in the PLC program.
OP acknowledgment	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Acknowledgement _OP" with a 16-bit array element and a corresponding name. Following conversion, there is one array element of the newly created array tags for every bit of the ProTool area pointer.
PLC acknowledgment	Area pointer discarded.
	Substitute: The array tag for error alarms is extended during migration.
Recipe box (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Recipe continuation box (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Recipe number (S5 only)	Area pointer discarded
	Indirect transfer of recipes is no longer supported. The transfer is made directly instead.
Interface area	The data are exchanged with the PLC via the following area pointers:
	• "Coordination"
	• "PLC job"
	• "Date/time"
	• "Conversion"
	Once migration is complete, you must create the area pointer in the project.
	This distribution simplifies data exchange between HMI device and PLC. The PLC program must be configured accordingly after conversion.
Error alarms	Area pointer discarded.
	The area pointer is replaced during conversion by the external array tag template "PT_Error alarms" with a 16-bit array element and a corresponding name. Following conversion, each bit of the OP17 area pointer corresponds with one bit in the array element of the new array tags. The alarms are assigned to the bit numbers in ascending order of alarm numbers. The array elements automatically receive the update identifiers 9-16.
	As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned warning alarm as "received." In reverse the alarm is interpreted as "sent" after a bit in the array element is reset to "FALSE" on the PLC.
System keyboard	The area pointer is discarded, because the OP is not equipped with system keys.

Area pointers on OP17	Area pointers on OP 177B
LED mapping	Area pointer discarded.
	The conversion program replaces the area pointer with an external array tag template "PT_LED mapping". The LEDs on the HMI are assigned to the array elements of the new array tags by the same order as in the OP17 area pointer. The array elements automatically receive the update identifiers 17-24.
	Every array tag is two bits long. The following states can be implemented by setting the bits belonging to an LED in runtime:
	LED function (Bit1/Bit2)
	Off (0/0)
	Slow flashing (1/0)
	Rapid flashing (0/1)
	Steady (1/1)

6.3.3 Distribution of the "Interface area" area pointer

Introduction

During conversion to WinCC flexible, the ProTool "interface range" area pointer is automatically distributed to three area pointers:

- "Coordination"
- "PLC job"
- "Date/time"

When migration is completed, you must create those area pointers in the project.

This distribution simplifies data exchange between the HMI and the PLC. The PLC program must be adapted accordingly after conversion.

Distribution to S7 PLCs

The interface range is distributed as follows for S7 PLCs:

ProTool	WinCC flexible
Interface range n+0 to n+31	Coordination n+0 to n+1
	PLC job n+4 to n+11
	Date/time n+15 to n+26

Handling of the start bit (coordination area pointer)

The coordination area pointer contains the start bit. If the value of this bit is read out in the PLC program, you can see whether the HMI device has started up from the PLC. The start bit is temporarily set to "0" by the HMI device during the startup process. As soon as the boot process is complete, the start bit is set to "1".

Special features in SIMATIC S5

The interface DB in the ProTool project can be reused after migration.

Example:

Area pointer	AS of group 1	AS of group 2
Job mailbox	DW32 to DW35	DW32 to DW35
Date/time	DW42 to DW47	DW66 to DW71

For AS of group 2: WinCC flexible HMIs support only one job mailbox. The PLC job with its three parameters is written directly to this mailbox.

6.4 System functions, events, and PLC jobs

6.4.1 System functions

Conversion

During migration from an OP17 to an OP 177B, the ProTool system functions of the OP17 project are replaced by the corresponding WinCC flexible system functions.

In WinCC flexible, the names of the system functions are assigned according to a specific convention. For this reason the names of the WinCC flexible system functions are not identical to same as the names of the corresponding ProTool system functions. The system functions groups in WinCC flexible also have different names from those in ProTool.

The system functions are converted as follows:

OP17 project: ProTool system function	OP 177B project: WinCC flexible system functions
Calculation	
Conversion linear 1	Replaced by the "LinearScaling" system function.
	Note: Change the converted functionality to the effect that the "x" and "y" tags are not identical.
Conversion linear 2	Replaced by the "InvertLinearScaling" system function.
	Note: Change the converted functionality to the effect that the "x" and "y" tags are not identical.
Tag: Convert value	Replaced by the "LinearScaling" system function.
Tag: Set value	Replaced by the "SetValue" system function.
Tag: Value minus/plus	Replaced by the "IncreaseValue" and "DecreaseValue" system functions.
Tag: Value minus/plus - cursor	Replaced by the "IncreaseFocusedValue" and "DecreaseFocusedValue" system functions.

OP17 project: ProTool system function	OP 177B project: WinCC flexible system functions
Data set	
Data record: OP -> PLC	Replaced by the "SetDataRecordToPLC" system function.
Data record: PLC > OP	Replaced by the "GetDataRecordFromPLC" system function.
Delete record	Replaced by the "DeleteDataRecord" system function.
Recipe table of contents	Replacement as follows depending on configured parameter: • Parameter "2 = edit" Replaced by the "ActivateScreen" system function which can be used to call a screen with a recipe view. The data records can be edited in the recipe view. • Parameter "5 = print" No longer supported; recipe data records can be printed with the aid of a recipe report. • Parameter "7 = transfer" Replaced by the "ActivateScreen" system function which can be used to call a screen with a recipe view. The data records can be transferred to the PLC sing the recipe view.
Call data record	No longer required because data records are viewed and edited with the recipe view.
Alarms	
Clear error alarm buffer	The migration program creates the "ClearAlarmBuffer" function with error alarm parameters.
First/last error alarms	Not supported; the sequence in which error alarms are displayed can be set during configuration of the alarm view. In runtime the sequence cannot be changed.
Error alarm overflow warning	Not supported A job can be assigned to the "Alarm buffer overflow" event as a substitute for the overflow warning. The job is configured in the scheduler. A system alarm can also be output for the event.
Clear alarm buffer	The migration program creates the "ClearAlarmBuffer" function with warning alarm parameters.
Event overflow warning	Not supported A job can be assigned to the "Alarm buffer overflow" event as a substitute for the overflow warning. The job is configured in the scheduler. A system alarm can also be output for the event.
Alarm report ON/OFF	Replaced by the "SetAlarmReportMode" system function.
FM functions, MCU functions, NC functions	Replaced with the SINUMERIK option.
Password	
Password logout	Replaced by the "Logoff" system function.

OP17 project: ProTool system function	OP 177B project: WinCC flexible system functions
Print	
Print error alarm buffer	Replaced by the "PrintReport" system function which can be used in migration to print an alarm report for the "Error alarms" alarm class.
Print event buffer	Replaced by the "PrintReport" system function which can be used in migration to print an alarm report for the "Warnings" alarm class.
Print all data records	Replaced by the "PrintReport" system function which can be used in migration to print a recipe report of all records.
Print data record	Replaced by the "PrintReport" system function which can be used in migration to print a recipe report of a record with a specified record number.
Hardcopy	Replaced by the "PrintScreen" system function.
Print all screens	Not supported If you want to print all screens, copy them into a report and print the report.
Screen table of contents	No longer required.
Screens	
Screen branching	No longer required; navigation among the screens created from the screen entries during migration is implemented with the "ActivateScreen" system function.
Special screen selection	No longer required.
Display selection	Replaced by the "ActivateScreen" system function.
Screen table of contents	No longer required.
Refresh screen	No longer required.
System settings	
S7 backup	Not supported
Toggle	
S7 change operating status	Not supported
S7 change PLC	Replaced by the "ChangeConnection" system function.
Language change	Replaced by the "SetLanguage" system function.
Alarm level	Replaced by the "ActivateScreen" system function which can be used to call a screen with an alarm view.
Changing the operating mode	Replaced by the "SetDeviceMode" system function.

6.4.2 Events

Conversion

Almost all events are imported without change during migration from OP17 to OP 177B. The following events are replaced during migration:

- "Input value"

  The "Input value" event is replaced by the "Change value" event during migration.

  If you want to trigger an event with an input at the HMI device (regardless of whether the value of the tag changes), proceed as follows. Configure an additional system function of the "Bit processing" group to the <Enter> key (with which the operator completes the input) to set the bit and evaluate the set bit.

- "Output value"

  The "Output value" event is replaced by the "Change value" event during migration. This event is always triggered when the value of the tag changes, regardless of whether the change occurs in the PLC or by input at the HMI device.

6.4.3 PLC jobs

Introduction

Only certain PLC jobs of an OP17 project can be used in an OP 177B project after conversion. Because of the different concepts of ProTool and WinCC flexible some PLC jobs are no longer required or can no longer be used. Under certain circumstances, the OP 177B project and the PLC program must be edited in order to obtain the functionality of the OP17 project.

If the converted project requires editing in any case, check whether the tasks done with PLC jobs in the OP17 project can be done more efficiently with system functions in the OP 177B project.

Conversion

The PLC jobs are converted as follows during migration from OP17 to OP 177B:

OP17	OP 177B	Explanation/replacement/workaround
3 "Hardcopy"	Not supported	Instead configure the "PrintScreen" system function to the "Value change" event of a tag with PLC connection.
5 "Select table of contents"	No longer required.	
7 "Print all screens"	Not supported	If you want to print all screens, copy them into a report. You can then configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection.

OP17	OP 177B	Explanation/replacement/workaround
10 "Print recipe with all data records"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. Recipe data records can be printed with the aid of a "Print recipe" object.
11 "Special screen selection"	No longer required.	To select screens use either PLC job 51 "Screen selection" or configure the "ActivateScreen" system function. You can use system function to implement other functions out of special screens, "AdjustContrast" to alter the contrast, for example.
12 "Switch alarm logging on/off"	Not supported	Instead configure the "SetAlarmReportMode" system function to the "Value change" event of a tag with PLC connection.
13 "Switch language"	Not supported	Instead configure the "SetLanguage" system function to the "Value change" event of a tag with PLC connection.
14 "Set time (BCD-coded)"	14 "Set time (BCD-coded)"	---
15 "Set date (BCD-coded)"	15 "Set date (BCD-coded)"	---
16 "Internal interface parameters (RS232)"	Not supported	Make the communication settings directly at the OP 177B with the loader menu.
17 "Module interface parameters (TTY)"	Not supported	Make the communication settings directly at the OP 177B with the loader menu.
19 "Printer parameters"	Not supported	Set the printer parameters directly at the OP 177B with the loader menu.
21 "Error alarm display type"	Not supported	In WinCC flexible, alarms are displayed with the "Alarm display" screen object or with the "Alarm window" screen object. The filter criteria for the alarms can only be set in the properties view of the screen object in the "General" group. Filtering by "Priority," "Acknowledgment group" and "Status" is not supported.
22 "Set display contrast"	Not supported	Instead configure the "AdjustContrast" system function to the "Value change" event of a tag with PLC connection. Otherwise set the preconfigured default screen "P_LANGUAGE_SCREEN." "P_LANGUAGE_SCREEN" contains three buttons, labeled accordingly. • Button 1: Set language ("SetLanguage" system function) • Button 2: Increase contrast ("AdjustContrast(1)" system function) • Button 3: Reduce contrast ("AdjustContrast(0)" system function)

OP17	OP 177B	Explanation/replacement/workaround
23 "Set password level"	23 "Set user group"	The PLC job "23 Set user group" takes the changed concept of user administration in WinCC flexible into account because it sets the user group instead of the password level. If you use PLC job 23, the PLC program must be modified accordingly.
24 "Password logout"	24 "Password logout"	---
31 "Print error alarm buffer"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. This function can be used to print an alarm report for the alarm class "Error alarms."
32 "Print event buffer"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. This function can be used to print an alarm report for the alarm class "Warning alarms."
37 "Switch overflow warning for warning alarms on/off" 38 "Switch overflow warning for error alarms on/off"	PLC job is not required.	There is no buffer overflow warning in WinCC flexible. Configure the "Alarm buffer overflow" event as a replacement for the system function "ShowSystemAlarm." You can output your own alarm text with this system function. The alarm text is restricted to one language.
41 "Transfer date/time to PLC"	41 "Transfer date/time to PLC"	---
42 "Get LED range from the PLC" 43 "Get event message range from the PLC" 44 "Get alarm message range from the PLC" 45 "Get acknowledgment range from the PLC" 47 "transfer LED range directly to the HMI device"	PLC jobs are not supported.	Instead use PLC job "46 update tag" with appropriate parameters. This PLC job reads the values of WinCC flexible tags that replace the ProTool area pointers "event messages," "alarm messages,", "PLC acknowledgment" and "LED mapping" after migration. Control which tag is read with the "Update identifier" parameter. Assign the update identifier of the desired tag to this parameter. Assignment of the parameter "update identifier": 1...8 warning alarms 9...16 error alarms 17...24 LED mapping
49 "Clear event buffer"	49 "Clear event buffer"	---
60 "Clear error alarm buffer"	60 "Clear error alarm buffer"	---
51 "Screen selection"	51 "Screen selection"	---
52 "Print out screen"	Not supported	Instead configure the "PrintScreen" system function to the "Value change" event of a tag with PLC connection.

OP17	OP 177B	Explanation/replacement/workaround
53 "Recipe selection"	Not supported	Instead configure the "ActivateScreen" system function to the "Value change" event of a tag with PLC connection with the "P_RECIPE_SCREEN" parameter. The preconfigured screen "P_RECIPE_SCREEN" contains recipe view and was created during conversion.
54 "Print out recipe"	Not supported	Instead configure the "PrintReport" system function to the "Value change" event of a tag with PLC connection. A recipe can be printed with a recipe report.
69 "Transfer recipe record from PLC to TP/OP"	69 "Read data record from PLC"	In WinCC flexible, "Recipe number" and "Record number" must be given as parameters.

If you want to transfer the recipe name or record names from the PLC to the HMI device, use the "GetDataRecordFromPLC" system function instead of the PLC job 69. |
| 70 "transfer recipe record from TP/OP to PLC" | 70 "Write record to PLC" | In WinCC flexible, "Recipe number" and "Record number" must be given as parameters.

If you want to transfer the recipe name or record name from the HMI device to the PLC, use the "WriteRecordTagToPLC" system function instead of the PLC job 70. |
71 "Partial screen update"	No longer required.	---
72 "Cursor positioning in current screen"	Not supported	Use PLC job 51 "Screen selection" instead.
73 "Cursor positioning in current special screen"	Not supported	Use PLC job 51 "Screen selection" instead.
74 "Keyboard simulation"	Not supported	---

6.4.4 Replacing job mailboxes with system functions

Introduction

Some PLC jobs that can no longer be used with WinCC flexible HMI devices can be replaced by system functions.

If the WinCC flexible project requires editing for any reason, check whether the tasks that were done with PLC jobs in the ProTool project can be done more efficiently with system functions in WinCC flexible.

Configuring

If you want the PLC to trigger the execution of a system function configured on the HMI device, configure a tag with the property "Always update" in WinCC flexible. Configure the corresponding system function for the "Change value" event on this tag.

Execution

If the value of the tag changes in the PLC, the system function is executed on the HMI device as soon as it detects the change.

Example: Trigger system function "SetAlarmReportMode" from the PLC

The following configuration can be used to replace the PLC job 12 "Switch alarm logging on/off" used in the ProTool project with a system function in the WinCC flexible project.

In WinCC flexible

1. Create an external tag of the INTEGER type.

2. Select "Tag > General" to set the "continuous cyclic" acquisition cycle.

3. Configure the "SetAlarmReportMode" system function to the event "Change value." Input the tag itself as parameter.

On the PLC

1. Set the value of the tag in the PLC program to 0 (alarm logging off) or 1 (alarm logging on).

Result

The alarm logging is switched on or off depending on the value of the tag.

6.5   Alarm system

Introduction

During migration from an OP17 to an OP 177B, alarms of all existing alarm classes are converted resulting in the following alarms classes continuing to be available:

* Warning alarms

* Error alarms

* System alarms

* Custom alarm classes

Tags and text lists that are included in the alarm text are imported during conversion.

Preconfigured screen replaces alarm level

The alarm level is not supported on the OP 177B. Instead, during migration a preconfigured screen called "P_MESSAGE_SCREEN" is created to show the alarms. This screen shows the alarms with the help of the "Advanced alarm display" screen object. The alarm display includes the system date and time. A display in the alarm text is therefore no longer required. The date/time display is removed from the alarm text during conversion. The alarm display shows the alarm number, alarm text and the abbreviation of the alarm class in a two-line view for every alarm.

The "P_MESSAGE_SCREEN" screen is opened in runtime using the <F5> key, or the "ActivateScreen" system function.

Conversion of display variations

The "Mixed" and "Separate" display variations are not supported. The display variations are converted as follows:

- Display variation "Separate"

  The "P_MESSAGE_SCREEN" contains two alarm views: one for display of the error alarms and one for display of the warning alarms and system alarms.

- Display variation "Mixed"

  The screen that replaces the alarm level contains one alarm view to display system alarms, error alarms and warning alarms together. They are displayed in chronological order of receipt. Sorting the alarms can be set during configuration and cannot be changed in runtime.

  The screen with the alarm display is opened with a function key or with the "ActivateScreen" system function after conversion, depending on the former implementation of calling the alarm level.

Rest alarm

The rest alarm is not supported.

Priority of alarms

In the OP 177B priorities can only be assigned for alarm classes, not for single alarms. Priorities that were assigned to various alarms in the OP17 are discarded after conversion.

Conversion of alarm area pointers

During conversion of an OP17 project to an OP 177B project, array tags with a 16-bit array element are created to replace the area pointers "event messages" and "alarm messages." Corresponding names are given to the array tags. Following conversion, each bit of the OP17 area pointer corresponds with one bit in the array element of the new array tags. The alarms are assigned to the bit numbers in increasing order of alarm numbers.

As soon as the PLC sets the value of a bit in the array element to "TRUE," the HMI device detects the assigned process event as "received." In reverse, the alarm is interpreted as "sent" after the bit in the array element is reset to "FALSE" on the PLC.

Alarm buffer overflow

The "Alarm buffer overflow" event is triggered when the alarm buffer is completely filled. The overflow of the alarm buffer cannot be queried separately for system alarms and warning alarms. The "ClearAlarmBuffer" can be used to delete alarms of specific classes from the alarm buffer and thus empty the alarm buffer when it overflows.

Alarm logging

The alarm log with printing of alarms on receipt and transmission is retained after conversion.

Printing headers and footers with the alarm log is not supported.

6.6 User authorizations

6.6.1 Security concept in WinCC flexible

New user authorization concept in WinCC flexible

The concept of user authorization in WinCC flexible differs from that in ProTool.

In WinCC flexible, each users is assigned only to one user group. You assign those user groups characteristic access rights, namely the authorizations. User groups are not organized in a hierarchical structure.

A default "Administrator" user group is created in each project. Users assigned to this group in Runtime have all rights, and may also set up new users.

You determine what rights the user requires for efficient operation of the various control elements.

In Runtime, all users are identified by their user name and password.

Migration of the ProTool password level

During conversion of a ProTool project to a WinCC flexible project, the program creates up to nine user groups. One user group is created for each password level used in the ProTool project.

The "Administrators" user group is always created. All user authorizations for which password level 9 was valid in ProTool are assigned to this "Administrator" group. All other user authorizations that were assigned the same password level in ProTool are also assigned to the same user group in the WinCC flexible project. The user groups are assigned the authorizations as in the password levels of the ProTool project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Advanced options in WinCC flexible

In WinCC flexible, users may be assigned to any user group, without making any allowances for hierarchical structures. During conversion, check the project to see whether to create new user groups or not.

6.6.2   Adapting user authorizations

Introduction

During conversion of an OP17 project to an OP 177B project user groups are created that have the same authorization that were assigned to the password levels in the OP17 project.

In addition, the various control elements will be assigned the same access protection as in ProTool during migration.

Password list is not migrated

The password list is not migrated because it is saved on the specific HMI device and is not in the project data.

After migration, the users must be created again. There are two options for this on the OP 177B:

- New users, user groups and passwords are created during configuration with the runtime user administration.

- A screen with a user view is configured. The user view shows existing users, passwords and user groups to an administrator. An administrator can also create new users, user groups and passwords with the user view.

Conversion of default screen for password editing

The default screens for working with passwords are converted as follows:

Default screen of OP17	Replacement on OP 177B
Password processing - login	Replaced by the "ShowLogonDialog" system function call.
Password processing - logout	Replaced by calling the "Logoff" system function.
Password processing - edit	Replaced by the preconfigured screen "P_USER_SCREEN". "P_USER_SCREEN" contains a user display.

Advanced options of OP 177B

On the OP 177B users can be assigned to a user group as desired without worrying about hierarchies. During conversion, check the project to see whether new user groups should be created. In addition, appropriate names for the user groups created during migration can be assigned in the "Groups" editor in the "User administration runtime" group.

6.7 Print

Introduction

The printing of project data in WinCC flexible is controlled by reports. Reports contain objects that enable screens, the alarm buffer and recipes to be printed.

The user of an OP 177B can only print the alarm log and the currently displayed screen (hardcopy) without requiring a report.

Headers and footers are not included in the migration from OP17 to OP 177B and thus not printed.

Alarm logging

During migration of OP17 to OP 177B the settings for the alarm logging are retained 1:1. During migration the corresponding OP17 system function is replaced by the OP 177B "SetAlarmReportMode" system function.

Printing on alarm buffer overflow

If "Logging on overflow" is configured in the OP17 project, use the following configuration to set the same behavior in the OP 177B project:

1. Create a report with the "Print alarm" object.

2. In the properties view of the alarm report select "Alarm events" as "Source for alarms."

3. In the scheduler create a job that will be triggered by the "Alarm buffer overflow" system event.

4. Assign the "PrintReport" system function to the job with the report and alarm log to be printed.

The alarm log is printed when the alarm buffer overflows.

Printing the alarm buffer

On the OP17, system functions control the printing of the event buffer and the error alarm buffer. During migration from an OP17 to an OP 177B, this function is converted as follows:

Two reports are generated with a "Print report" object ("Alarm log"). The reports are configured so that the warning alarms or error alarms are printed from the alarm buffer. The corresponding system functions of the OP17 project are replaced by the "PrintReport" system function for printing the desired report.

The alarms can be printed in ascending or descending chronological order. They cannot be sorted by alarm number.

Printing the displayed screen (hardcopy)

When an OP17 project that has the option of printing the displayed screen (hardcopy) is converted, the OP 177B project retains this function. During migration the corresponding OP17 system function is replaced by the OP 77B "PrintScreen" system function.

Printing screens

Printing all screens in one print job is not supported on the OP 177B. Use the project documentation to document screens in a project. The project documentation used to output selected or all configuration data of a project.

If you want to allow the user to print screens in runtime even if they are not displayed, proceed as follows:

1. Create a report for every screen by configuring all objects from the screen. Objects that cannot be configured can also not be printed, e.g. bars.

2. Configure an IO field so the user can select one of the screens from a text list. Configure a tag for the IO field to save the result of the selection.

3. Configure the "PrintReport" system function to a button. Send the value of the tag as a parameter.

In runtime the user can select a screen from the text list and print with the button.

If you have used printouts of screens for the project documentation, call the menu command "File > Print Project Documentation".

Printing recipes

On the OP17, printing of all or one specific data record of a recipe is controlled by system functions. During migration from an OP17 to an OP 177B, this function is converted as follows:

Two reports are generated with a "Print recipe" object ("Recipe log"). The recipe reports are configured so that all data records are printed with one recipe report and the currently selected data record is printed with the other recipe report. The corresponding system functions of the OP17 project are replaced by the "PrintReport" system function for printing the desired report.

Printer setting

Printers connected to OP 177B are configured in the Control Panel. Printer configuration by means of system functions is not supported. Detailed information on the loader menu can be found in the OP 177B user manual.

6.8 Recipes

Introduction

During migration from an O1P7 project to an OP 177B project, the recipes are also converted.

Converting the recipes

In detail the recipes are converted as follows:

- The recipe header becomes the localized recipe name.

- The recipe number is imported unchanged.

- The table below shows how to convert recipe entries and the corresponding tags.

  The names of the recipe entries are formed from the total text of the recipe entry. The tag value on the OP 177B will always be displayed after the name of the recipe entry.

OP17	OP 177B
Dual-line, 80 characters overall	Single-line, 40 characters overall
	Note: In WinCC flexible, longer texts must be truncated.

Example (display of the tag value is shown by <...>):

Recipe entry on OP17	Recipe entry on OP 177B
Sugar <1.5> kg	Sugar (kg) <1.5>
Water <5> l	Water (l) <5>

WinCC flexible does not support password protection for separate recipe entries. Instead, configure access control by assigning a password when you call the P_RECIPE_SCREEN screen.

Conversion of data records

The data records of the recipes cannot be converted. They must be input again. The data records can be created in the "Recipes" editor on the configuration computer and then sent to the OP 177B.

Display of recipes

The OP 177B has the "Recipe display" screen object for displaying the recipes. The data records of a recipe can be displayed, edited and sent to the PLC with this display on the HMI device.

During migration a screen with a recipe view is created instead of the recipe table of contents and the individual recipes. The recipe view is automatically configured in accordance with the recipes existing in the OP17 project.

The "Flashing" property is not supported for the names of the recipe elements.

Area pointer for transfer of data records

The area pointer used for transferring data records of a recipe depends on the PLC to which the OP 177B is linked.

- Connection to the S7

  On connection to the S7 the "Data record" area pointer is used to transfer data records of a recipe.

  After migration from OP17 to OP 177B, the length of the data record remains the same, and the structure is converted as follows:

  1. Word: Recipe number (unchanged)

  2. Word: Record number (previously 4th word)

  3. Word: Reserved

  4. Word: Record acknowledgment (previously byte n+3 of the interface range)

  5. Word: Free

- Connection to the S5

  On connection to the S5 the transfer of the data records is converted from indirect to direct transfer. The recipe number box, the recipe box and the recipe subsequent box are discarded.

6.9 Migration of the configured PLC drivers

1:1 conversion

Most PLC drivers available for OP17 are also available for OP 177B. They are automatically converted during conversion. This applies to the following PLC drivers:

PLC driver	Supported by migration	RS232 coupling via adapter 1)
SIMATIC S7-300/400	✓	
SIMATIC S7-200 (one PLC)	✓	
SIMATIC S7-NC	— 2)	—
SIMATIC S5 DP	✓	
SIMATIC 500/505 DP	✓	
Allen-Bradley DF1 PLC-5	✓	✓
Allen-Bradley DF1 SLC 500	✓	✓
Allen-Bradley DH485	✓	
GE Fanuc SNP	✓	✓
MITSUBISHI FX	✓	
MITSUBISHI Protocol 4	✓	✓
Modicon Modbus	✓	✓
Omron Host-Link/Multilink	✓	✓
Telemecanique Adjust	— 2)	—
1) For a RS232 coupling, an adapter is necessary (MLFB no. 6ES5 ...)		
2) If you convert a project that has a connection to either the "Telemecanique Adjust" or "SIMATIC S7-NC" PLC, a warning is shown during conversion. The "SIMATIC S7-300/400" PLC driver is set on the OP 177B project as a substitute. All tags will be separated from the PLC.		

Project with connection to SIMATIC S5-FAP / AS511

If the SIMATIC S5 FAP PLC was used in the OP17 project, a SIMATIC S5 AS511 must be selected in ProTool before migration. After migration, you can change to a SIMATIC S5 DP in WinCC flexible.

Procedure:

1. Open the OP17 project in ProTool.

2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 FAP" protocol is now shown on the right.

3. Double-click on the input PLC to change the PLC protocol.

4. Select the SIMATIC S5 AS511 PLC protocol from the "PLC" dialog. Confirm with "OK".

5. Save the project.

6. Close ProTool.

7. Open the OP17 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."

8. Select the reference language.

The project is automatically converted to an OP 177B project and then opened in WinCC flexible.

It is now connected to the SIMATIC S5 AS511.

To change to a SIMATIC S5 DP:

1. Double-click the "Connections" entry in the project view for the OP 177B. The "Connections" editor opens.

2. Select "SIMATIC S5 DP" as the connection.

3. Replace the default FB (FB 52 or FB 53) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

Project with connection to the SIMATIC S5-L2-DP

If the SIMATIC S5-L2-DP PLC was used in the OP17 project, the SIMATIC S5 DP PLC must be used on the OP 177B. Before migration, switch to a SIMATIC S5 AS511 in ProTool. After migration, change to a SIMATIC S5 DP in WinCC flexible.

Procedure:

1. Open the OP17 project in ProTool.

2. From the project view select the "PLC" object. The PLC with the "SIMATIC S5 L2 DP" protocol is now shown on the right.

3. Double-click on the input PLC to change the PLC protocol.

4. Select SIMATIC S5 AS511 as PLC protocol in the "PLC" dialog and confirm with "OK".

5. Save the project.

6. Close ProTool.

7. Open the OP17 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."

8. Select the reference language.

   The project is automatically converted to an OP 177B project and then opened in WinCC flexible.

   It is now connected to the SIMATIC S5 AS511.

9. Double-click the "Connections" entry in the project view for the OP 177B. The "Connections" editor opens.

10. Select "SIMATIC S5 DP" as the connection.

11. Replace the default FB (FB 58) with the FB 158 in the PLC program. Otherwise errors may occur during communication between PLC and HMI device.

Project with two SIMATIC S7-200 PLCs

With OP17 projects it is possible to configure multiple PLCs with the SIMATIC S7 200 PLC driver protocol when the PPI profile is used.

On the OP 177B, only two PLCs can be coupled with the MPI communication network. The PLC profile must be changed in the OP17 project before conversion to ensure a successful conversion.

Procedure:

1. Open the OP17 project in ProTool.

2. From the project view select the "PLC" object. The two PLCs with the SIMATIC S7-200 protocol are now displayed on the right.

3. Double-click on the first PLC to open the "PLC" dialog box.

4. Click on "Parameter" and select the entry "MPI" as "Profile." Confirm the setting with "OK."

5. Save the project.

6. Close ProTool.

7. Open the OP17 project on the start page of WinCC flexible by selecting "Options > Open ProTool project."

8. Select the reference language.

   The project is automatically converted to an OP 177B project and then opened in WinCC flexible.

Project with connection to the Omron Host-Link/Multilink

If the Omron Host-Link/Multilink PLC was used in the OP17 project, the station address is set to 0 during conversion to an OP 177B project. After conversion, the correct station address is set in the OP 177B project as follows:

1. Click in the project window on "Communication"> "Connections."

2. Select the entry "Omron Host-Link/Multilink" in the "Communications driver" column.

3. Select the "Parameter" entry in the properties view.

4. Enter the correct station address.

6.10 Timer

Converting timers to jobs

"Timers" of a ProTool project become jobs that are controlled by a time event after conversion to WinCC flexible.

During conversion, the ProTool alarm times are converted to a time event. The function list configured on the alarm is input into the function list of the event.

In ProTool projects a date/time field can be configured that can be used to assist the user to set the starting time of a job in runtime. If such a date/time field exists in a ProTool project, the migration program automatically generates a relevant field and the associated "PT_AlarmTag" tag in the WinCC flexible project.

Note

WinCC flexible only supports daily or annual tasks.

In WinCC flexible, the start of jobs can be configured not only by time events but also by system events, such as "Alarm buffer overflow" or "Change Screen". The project should be checked during conversion to see if it is more appropriate to start jobs by system events or time events.

Index