SIEMENS

SINUMERIK

SINUMERIK HMI engineering package WinCC flexible 2007

Configuration Manual

Avaiable on CD as a PDF

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 only to property damage 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.

Disclaimer of Liability

We have reviewed the contents of this publication to ensure consistency with the hardware and software described. Since variance cannot be precluded entirely, we cannot guarantee full consistency. However, the information in this publication is reviewed regularly and any necessary corrections are included in subsequent editions.

Siemens AG
Automation and Drives
Postfach 48 48
90327 NÜRNBERG
GERMANY

Ordernumber: Avaiable on CD as a PDF
℗ 07/2007

Preface

Preface

Note about the content of this document

Note

This document, "SINUMERIK HMI WinCC flexible Configuration Software" describes the configuration options, which are possible with SINUMERIK HMI WinCC flexible configuration software.

The default configuration options, which are part of the WinCC flexible configuration tool, are described in full in the WinCC flexible basic system, including in the online help.

Table of contents

Introduction

1

1.1 Version 2007 - Compatibility

Overview of previous versions

WinCC flexible Basic Version	SINUMERIK HMI WinCC flexible configuration software
WinCC flexible 2004 SP1	V1.0
WinCC flexible 2005	V1.1
WinCC flexible 2005 SP1	V1.1 SP1
WinCC flexible 2007	2007

The SINUMERIK packages are uniquely assigned to corresponding basic versions of WinCC flexible. WinCC flexible has the following compatibility properties:

- All previous versions can be loaded
- Projects from the preceding product, ProTool, can be migrated as from Version 6.0 SP2
- With the current release, projects can be *saved* in the format of all preceding versions (as 2005 SP1 HF7, as 2005 and as 2004 SP1)
- The runtime format can only be produced for the current version (2007)

Compatibility overview SINUMERIK HMI WinCC flexible Configuration Software SINUMERIK and SINUMERIK HMI Advanced

The WinCC flexible SINUMERIK packages are compatible with all main and currently available HMI Advanced versions. Any new HMI Advanced Versions made available with a WinCC flexible SINUMERK release are subsequently tested and released.

SINUMERIK HMI WinCC flexible configuration software	SINUMERIK HMI
V1.0	6.4
V1.1	6.4 7.1 7.2
V1.1 SP1	6.4 7.1 7.2 SP2 7.3

SINUMERIK HMI WinCC flexible configuration software	SINUMERIK HMI
V2007	7.1 7.2 SP2 7.3

1.2 WinCC flexible basic system

WinCC flexible Engineering System

The WinCC flexible Engineering System is the software for handling all your essential configuring tasks. The WinCC flexible edition determines which HMI devices in the SIMATIC HMI spectrum can be configured.

WinCC flexible Runtime

WinCC flexible Runtime is your software for process visualization. You execute the project in process mode in Runtime.

WinCC flexible options

The WinCC flexible options allow you to expand the standard functionality of WinCC flexible. A separate license is needed for each option.

Special license conditions apply to SINUMERIK (see Section 1.5.6)

With the exception of the SINUMERIK HMI devices, the WinCC flexible basic system does not support SINUMERIK.

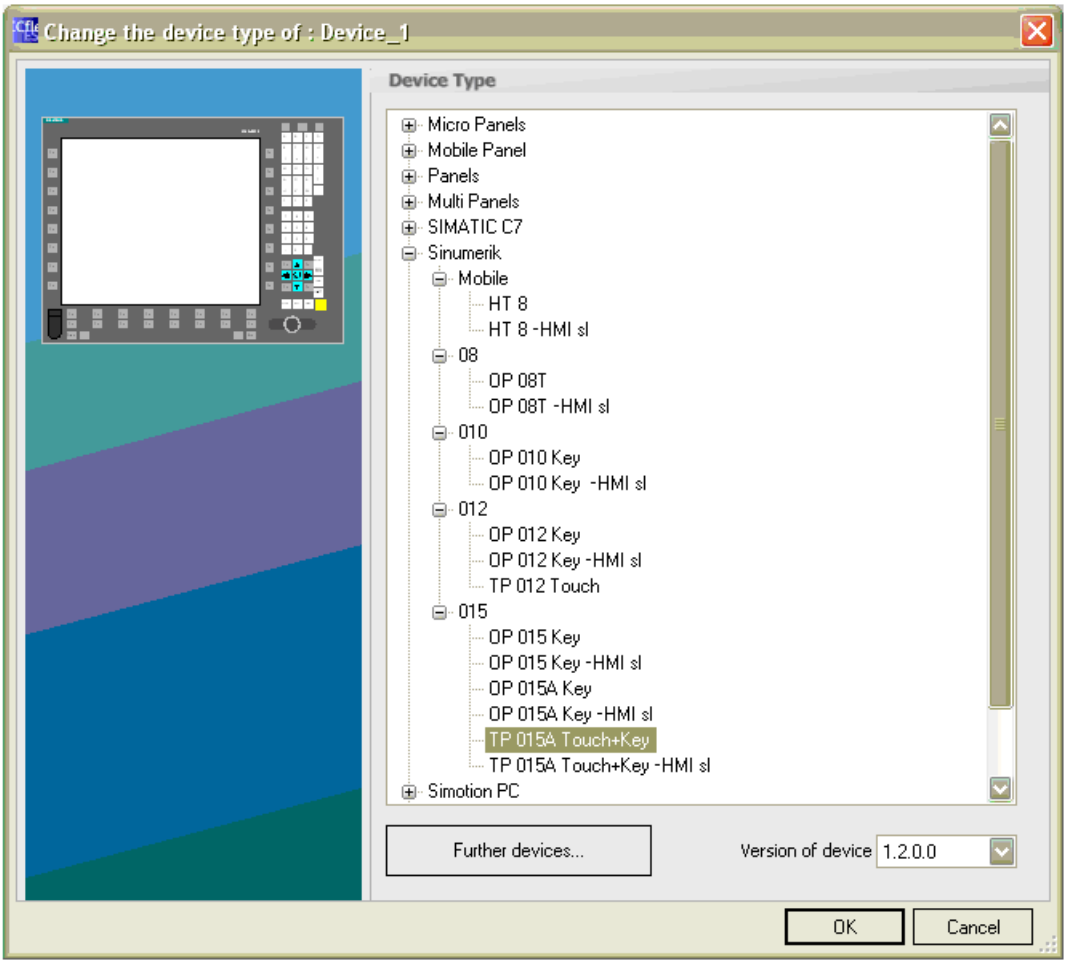
Operating devices

New SINUMERIK HMI devices are included in the selection dialog box of version WinCC flexible 2007. **HT8** and **OP08T** are now also supported.

All HMI devices are available in two versions (with the exception of TP012).

- As a PC HMI device for projects to expand HMI Advanced and standalone procedure on a PCU 50
- As an HMI device to expand the new SINUMERIK HMI sl interface. These devices take the extension -HMI sl.

Support of the new -HMI sl devices with communications channels will be activated for the SINUMERIK HMI WinCC flexible Configuration Software at a later stage using a future hotfix or service package as soon as a relevant version of the HMI sl with matching integrated runtime is available.



Besides expanding a standard SINUMERIK user interface using the PC HMI devices provided under SINUMERIK, a standalone configuration can also be established for a SINUMERIK for any additional PC HMI devices.

The following PC HMI devices are available for this purpose:

C7 636 (all variants)
PC IL 70 (all variants)
PC IL 77 (all variants)
PC 577 (all variants)
PC 670 (all variants)
PC 677 (all variants)
PC 870 (all variants)
PC 877 (all variants)
Standard PC

HMI devices with Windows CE and SINUMERIK interface

Mobile Panel

Mobile Panel 170
Mobile Panel 177 DP
Mobile Panel 177 PN
Mobile Panel 277

Touch Panel

TP 170B mono
TP 170B color
TP 177B mono DP
TP 177 B color PN/DP
TP 270 (all variants)
TP 277

Operator Panel

OP 170B mono
OP 177B mono DP
OP 177B color PN/DP
OP 270 (all variants)
OP 277

Multi Panel

MP 270 (all variants)
MP 277 (all variants)
MP 370 (all variants)
MP 377 (all variants)

1.3 Functional scope SINUMERIK HMI WinCC flexible configuration software

Functionality of PC-based HMI devices (integration in NC and PLC)

- Ethernet communication with SINUMERIK sl
- MPI communication with SINUMERIK sl
- MPI communication with SINUMERIK pl
- Visualization of BTSS tags
- Visualization of GUD tags
- Visualization of machine and setting data
- NC alarms, PLC alarms, DB2 alarms, Alarm_S
- Integration into the m:n concept of SINUMERIK HMI Advanced (up to 8 NCUs)
- Controls for HMI data management and NC editor
- Input controls of SINUMERIK HMI Advanced
- Triggering of specific PI services (for example, NC restart, setting a password)
- Triggering a general PI service (using the "General PI service" function)
- Coordinated change of language SINUMERIK HMI - WinCC flexible Runtime
- Symbolic addressing of the PLC element in STEP 7 integration
- Example projects

Functionality of CE-Panel HMI-device (integration in NC)

- Ethernet communication with SINUMERIK sl
- MPI communication with SINUMERIK sl
- MPI communication with SINUMERIK pl
- Visualization of BTSS tags
- Visualization of GUD tags
- Visualization of machine and setting data
- NC alarms and NC messages
- Control for activating part programs
- Triggering of specific PI services (for example, NC restart, setting a password)
- Triggering a general PI service (using the "General PI service" function)
- Sample Project

Languages of the runtime system of the PC HMI devices

All runtime languages that are supported by WinCC flexible 2007.

Language-specific expansions of the runtime by SINUMERIK (file viewer, editor, PI services) are available in the following languages:

- German
- English
- Italian
- French
- Spanish
- Chinese (PRC)
- Chinese (traditional)
- Japanese
- Korean

Languages of the engineering system of the PC HMI devices

- German
- English
- Italian
- French
- Spanish

Languages of the runtime system of the SIMATIC CE devices

All languages that are supported by WinCC flexible 2007.

Languages of the engineering system of the SIMATIC CE devices:

All languages that are supported by WinCC flexible 2007.

1.4 Migration of ProTool/Pro projects to WinCC flexible

1.4.1 Migration of simple Pro Tool/Pro projects

ProTool/Pro projects can be migrated to WinCC flexible as follows:

1. Select **Open ProTool Project** in the WinCC flexible project wizard.
2. Select a ProTool/Pro project.
3. Confirm by clicking **Continue** in the migration wizard.
4. The wizard migrates the project and then opens the migrated project in WinCC flexible.

For precise information about the topic of migration, please consult the WinCC online documentation.

1.4.2 Migration of STEP 7 integrated ProTool/Pro projects

1. Select **Open ProTool Project** in the WinCC flexible project wizard.
2. In the dialog box, to open a file, select the **integrated** button located to the left in the window. The existing STEP7 projects are listed.
3. Select **STEP7 projects** and the required STEP7 configuration.
4. After the migration into HW Config, define a suitable communications processor (e.g., CP5611) for the newly migrated HMI station and create and **save** a connection to the controller (communications driver) in NetPro.

Then open WinCC flexible from STEP 7 via the function "HMI Station --> Open". In the connections editor, you can now set the matching settings for the station, partner, and node for the existing SINUMERIK_PLC connection. The inserted Connection_1 can be removed.

Figure 1-1 SINUMERIK PLC

For more precise information about the topic of migration, please consult the WinCC online documentation.

1.4.3 Migration of ProTool/Pro projects with GUD tags

Introduction

The following scenarios must be considered when migrating ProTool/Pro projects with user data (GUDs):

- Projects with a connection to a controller
- Projects with n connections to a controller

Projects with a connection to a controller with GUDs

To migrate projects with a controller:

1. Copy the **globalUserData.mdb** ProTool/Pro user database to the WinCC flexible directory (.\SIMATIC WinCC flexible\WinCC flexible 2007).
   The database is located in the ProTool/Pro CS directory.
2. Migrate the ProTool/Pro project to WinCC flexible.
3. After migration, under **Connections** for the **SINUMERIK NC**, enter the name of the **globalUserData.mdb** user database in the interactive form for controllers in the **User database** field. Alternately, you can adjust the GUD again using the GUD tool and enter the resulting new database NCU840D.mdb in this field. This is compulsory if the GUDs have been changed, making adjustment necessary.

Projects with n connections to a controller with GUDs

Note

The existing **globalUserData.mdb** user database cannot be used by ProTool/Pro projects for m:n projects.

Proceed as follows when migrating:

1. Create one user database for each control using the **GUD-Tool** program.
   The controller names in netnames.ini from the ProTool project must be the same for the WinCC flexible project, otherwise the GUD tags will have to be reconnected with the relevant SINUMERIK control. The user data created must be stored in the WinCC flexible 2007 directory.
2. Migrate the ProTool/Pro project to WinCC flexible.
3. After migration, under **Connections**, enter the name of the user database for each controller in the interactive form for NC databases in the **User DB** field.

   Note

   If there are multiple controllers with GUDs in the project, the names of the individual user databases must correspond to the control names,

   e.g., controller **NCU_1** -> user database **NCU_1.mdb**,

   controller **NCU_2** -> user database **NCU_2.mdb**, etc.

1.5 Installation guide

1.5.1 Software requirements

Requirement

The software below is required prior to installation of SINUMERIK HMI WinCC flexible configuration software:

For SINUMERIK HMI devices

- WinCC flexible 2007
- For HMI-integrated work, a SINUMERIK HMI Advanced 7.1 or 7.2 SP2 or 7.3

For SIMATIC CE Panels

- WinCC flexible 2007

  The SIMATIC packages are not part of the SINUMERIK HMI WinCC flexible configuration software.

Contents of the supplied DVD:

SINUMERIK HMI configuring package	: 01.02.00.00	
with HMI Base	: **7.1**	: 07.01.00.00 (07.10.17.00)
	: **7.2 SP2**	: 07.02.01.00 (07.20.18.00)
	: 7.3	: 07.03.23.00
with SINUMERIK add-on for STEP 7	: V5.4 + SP1	: 01.04.05.00
SINUMERIK HMI Advanced (for PC/PG)	**: 7.3**	: 07.03.23.00

1.5.2 Installation

Proceed as follows

Start the program "Setup.exe" on the DVD.
The installation checks the system requirements and guides you through the program.

Install the ES(= Engineering system) system and RT(=Runtime) system (for testing) on the configuration computer.

Install the RT system on the target system.

ES system

- Windows 2000, Windows XP
- WinCC flexible 2007 ES
- SINUMERIK HMI WinCC flexible 2007
- STEP 7 Version 5.4 SP1 and higher (if STEP 7 is integrated)

RT system

- Windows XP
- WinCC flexible 2007 RT
- SINUMERIK HMI WinCC flexible 2007
- SINUMERIK HMI Advanced Version 7.1 and higher

1.5.3 Setup strategies

Introduction

The setup follows the following strategies in order to deal with differing system requirements.

- If HMI Advanced is installed, then at least 7.1, otherwise setup will fail.
- If no HMI Advanced is installed, *no* up-to-date version of HMI Advanced will be installed.
- If neither HMI Advanced nor HMI Base is installed, HMI Base 7.3 will be installed.
- If HMI Base 7.1 is installed (and no HMI Advanced), query whether to keep or upgrade to 7.2 SP2 or 7.3.
- If HMI Base 7.2 is installed (and no HMI Advanced), query whether to keep or upgrade to 7.2 SP2 or 7.3.

1.5.4 Integration into SINUMERIK HMI Advanced

Introduction

The following SINUMERIK configuration files must be adjusted to enable screens configured with WinCC flexible to be displayed correctly in the SINUMERIK HMI environment:

- Oemframe.ini
- Regie.ini
- Language\re_*.ini
- netnames.ini (only for m:n configuration)

These settings are made during setup. However, the entries in the "Regie.ini" file are only examples and must be adjusted accordingly. The configuration files are located in directory Add_On.

If a "Regie.ini" file already exists in the Add_On directory, setup will store a corresponding file as "Regie.bsp".

Entries in Oemframe.ini

```
[HmiRTm]
fSearchOnlyForTaskWindow=0
fKeepPlacement=1
fForceTaskFocus=1
nSleepTimeFL=150
nSleepTimeA=250
```

Entries in Regie.ini

```
[TaskConfiguration]
Task7=name:=OEMframe,Cmdline:="E:\\Program Files\\Siemens\\Simatic
WinCC Flexible\\WinCC flexible 2007 Runtime\\HmiRTm.exe
F:\\ADD_ON\\pro.fwx",TimeOut:=40000,HeaderOnTop:=false,Preload:=fals
e,ClassName:="HmiRTm",GimmeKeys:=0,DisableKeys:=196863
```

In other words:

- On triggering Task7 (= softkey HSK7), the WinCC flexible RT software with the created project, e.g., "pro.fwx" (freely-selectable project name), is started.
- "E:\\Program Files\\Siemens\\Simatic WinCC Flexible\\WinCC flexible 2007 Runtime\\HmiRTm.exe" is the path name in which you have installed WinCC flexible RT (this path may contain blanks).
- "F:\\ADD_ON\\pro.fwx"is the path name in which you saved the configuration (this path must **not** include any blanks).
- In "GimmeKeys := <bitmask>", the bits in the bit mask have the following effect:
  - Bit 1 set -> F11 (channel switchover) is delivered to the application directly
  - Bit 2 set -> ESC (alarm acknowledgment) is delivered to the application directly
- The DisableKeys:=196863 entry enables the left softkey range and the second horizontal softkey range.

Note

For more setting options for parameterizing the program call, please consult Appendix A.3 Description of section [TaskConfiguration].

Language\re_*.ini

```
[HsoftkeyTexts]
HSK7="application"
```

netnames.ini

If m:n is set, then the following must be observed in the file "netnames.ini" in order that alarms and NC messages function:

In the file "netnames.ini" the name of the connection and the name within the section - e.g. "param NCU_RECHTS" must be the same, e.g.:

Connection:

```
Conn_1=NCU_RECHTS
```

Section:

```
[param NCU_RECHTS]
name=NCU_RECHTS
```

See also

- M:N configuration with WinCC flexible (HMI PC devices) (Page 27)

1.5.5 WinCC flexible RT without HMI Advanced

Introduction

WinCC flexible RT can be operated without HMI Advanced.

General procedure

In this case, the servers required to enable communication with SINUMERIK are started by means of a master control subsystem.

The master control subsystem is configured in a **subsystem** INI file in the WinCC flexible RT directory. The table below shows which subsy*.ini file (depends on the SINUMERIK HMI installed) is used to start the master control subsystem:

SINUMERIK HMI	SINUMERIK powerline	SINUMERIK 840D sl
V7.1	subsy_71p.ini	subsy_71s.ini
V7.2 and higher	subsys_pl.ini	subsys_sl.ini

Entry **cp_840Di** is omitted for a communication link to a SINUMERIK powerline.

The **cp_840Di** entry is activated for a communication link to a SINUMERIK 840D sl.

If the setup finds file Loader32.exe (is additionally installed with HMI-Base with SINUMERIK HMI V7.2 SP2 and higher), the entry **Loader32** is activated!

1.5.6 Licenses

General

You will need the following licenses:

- Authorization for WinCC flexible ES
- OA copy licence per runtime

You will **not** need the following licenses:

- Powertags
- Archives and recipes

The "WinCC flexible 2007 Runtime" license for displaying the powertags is *not* required for SINUMERIK. Configure at least one "SINUMERIK_NC" or "SINUMERIK_PLC" connection. Release of this license only works for the "SINUMERIK_NC" and "SINUMERIK_PLC" connections. If an additional "S7-300/400" connection is configured, the runtime license for the Powertags will be needed.

1.5.7 Changes in Version 2007

General

Connection to HMI Advanced is implemented in the **Add_On** directory. In prior version, the **OEM** directory was used.

1.6 Example projects

General

Several example projects are supplied to elucidate the functional scope of the expansions for SINUMERIK and provide recommendations for certain procedures (for example, data management, editor, NC password, keyswitch).

- OP012_Fileman.hmi
- OP012_demo.hmi
- NCSecurity.hmi
- TP270_Sinumerik.hmi

OP012_Fileman.hmi contains a full configuration of the data management control and the NC editor control. Important aspects include the interaction between both controls across more than 2 images. A detailed description of this example is given in Section 2.3.3.

OP012_demo.hmi contains fully configured examples on the topics PI services, SINUMERIK HMI Controls and alarms.

NCSecurity.hmi shows how, by using the NC tag "Access Level", the visibility and operability of an IO field can be controlled. The most important settings are to be found in the properties dialog box of the IO field in the Animation area.

The TP270_Sinumerik.hmi example shows SINUMERIK-specific images on a Windows CE panel. For this, the following functions have been configured:

- Permanently defined PI services (e.g. NC Reset)
- Generally configurable PI service
- Read NC variables (with address multiplexing)
- Part program control (for activating part programs that have already been loaded)

The examples are to be found in installation directory Samples/Sinumerik.

See also

- Configuration example for file display plus ASCII editor (Page 54)

Working with 2

2.1 SINUMERIK connections

2.1.1 Communications principle

Overview

Internally, the SINUMERIK controller consists (in very simple terms) of

- A **program controller component** (S7-300 controller component), which, in terms of data communication, behaves in almost exactly the same way as an S7-300 controller
- A **numeric controller component** (NC controller component), which requires special tag descriptions and specific services.

All supported target systems have an S7 protocol or an expanded S7 protocol as their communication interface and can be connected via the MPI port.

Principles (see screen below)

In WinCC, the selection consists of

- HMI PC devices (e.g., OP012) for the
  - Program controller component of the **SINUMERIK PLC** communication driver
  - Numerical control section of the **SINUMERIK NC** communication driver
- HMI CE devices (e.g., mobile panel 170) for the
  - Program controller component of the **SIMATIC S7 300/400** communication driver
  - Numerical control section of the **SIMATIC NC** communication driver

It is possible to switch between **SINUMERIK PLC** and **SIMATIC S7 300/400** during project creation without losing the tag connection.

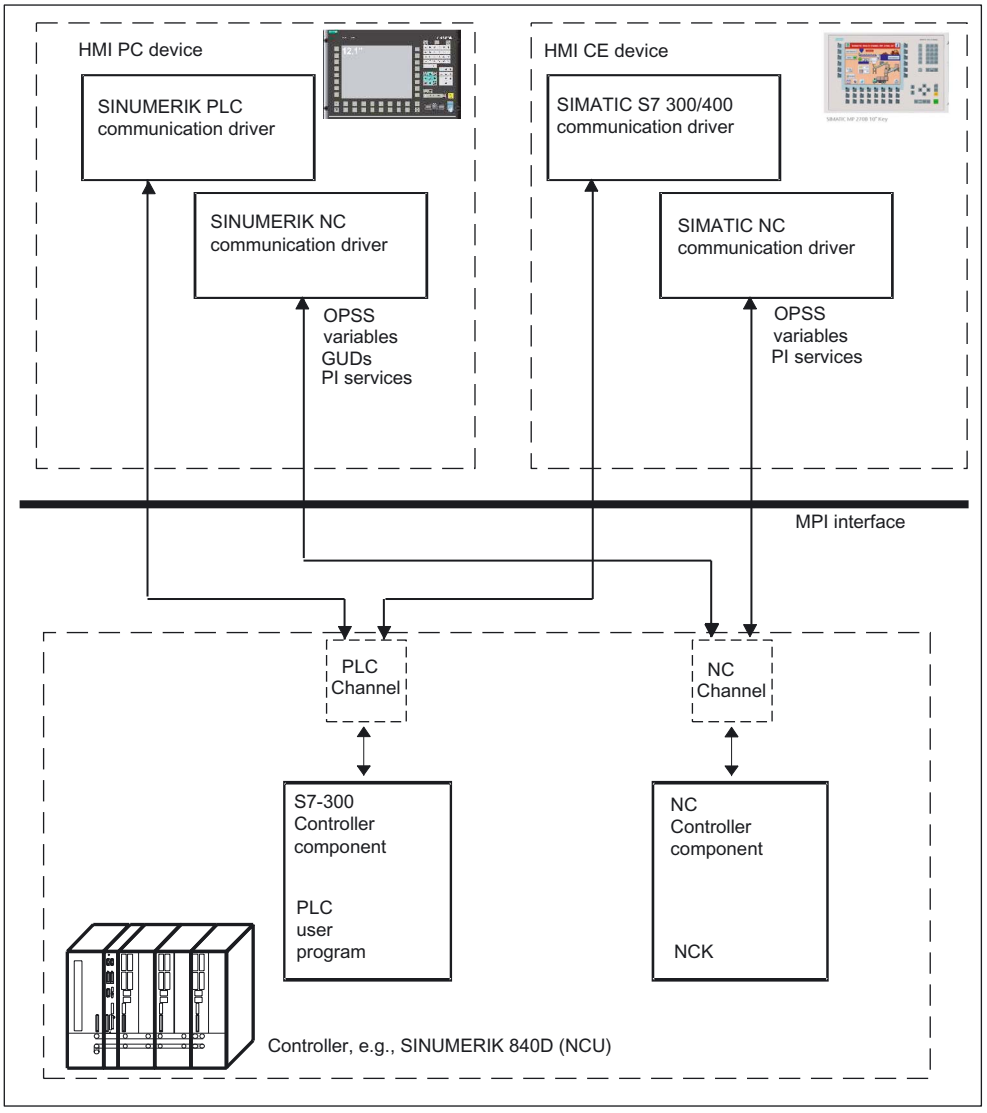


Figure 2-1 Communication principle

Note

For HMI PC devices, it is possible to read/write OPI tags and GUDs as well as to call up PI services via the **SINUMERIK NC** communication driver.

For HMI CE devices, it is possible to read/write OPI tags and GUDs as well as to call up PI services via the **SIMATIC NC** communication driver.

Several HMI devices or controllers can be operated (this corresponds to the SINUMERIK HMI m:n concept).

2.1.2 Communication to SINUMERIK controllers with HMI PC devices

General

In the case of PC HMI devices, the connection channel **Sinumerik_NC** is intended for communication with the numeric component of the SINUMERIK; connection channel **Sinumerik_PLC** is intended for communication with the PLC component.

Communication is parameterized in the configuration files of the SINUMERIK HMI Advanced interface. The installation ensures that the correct settings are made. The configuration files are located in the SINUMERIK HMI installation directory in the mmc directory and in the Add_On directory. See also Section 3.3, Description of MPI interface parameterization.

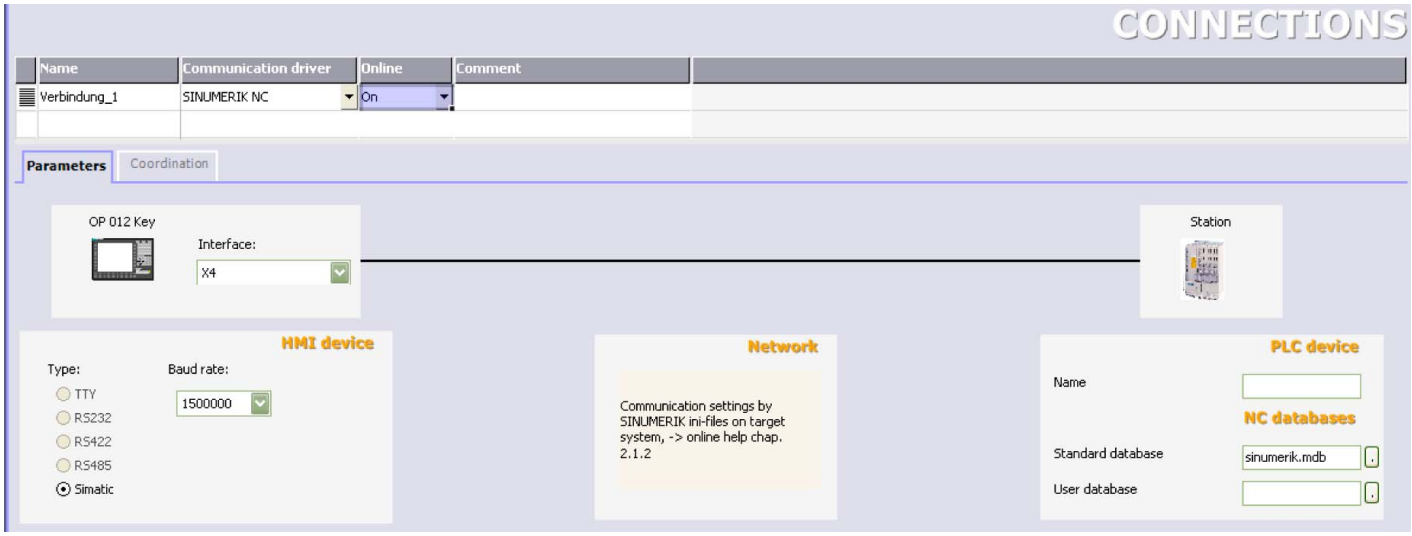


Figure 2-2 Connection parameters PC HMI devices

Usually, you will not have to change the default settings of the data for a Sinumerik_NC (Sinumerik_PLC) connection.

The "Name" parameter must be correctly set for an m:n configuration (see Section 2.1.4)

You must make an entry in parameter "User database" if you want to configure user data (GUDs).

2.1.3 Communication to SINUMERIK controllers with HMI CE devices

General

Connection channel **Simatic_NC** is intended for communication with the numeric component of the SINUMERIK for the HMI devices of the classes Mobile Panel, Panel, and Multi Panel.

Depending on the interface available, communication can be handled via the MPI/Profibus or Ethernet.

The communication parameters must be set in the engineering system.

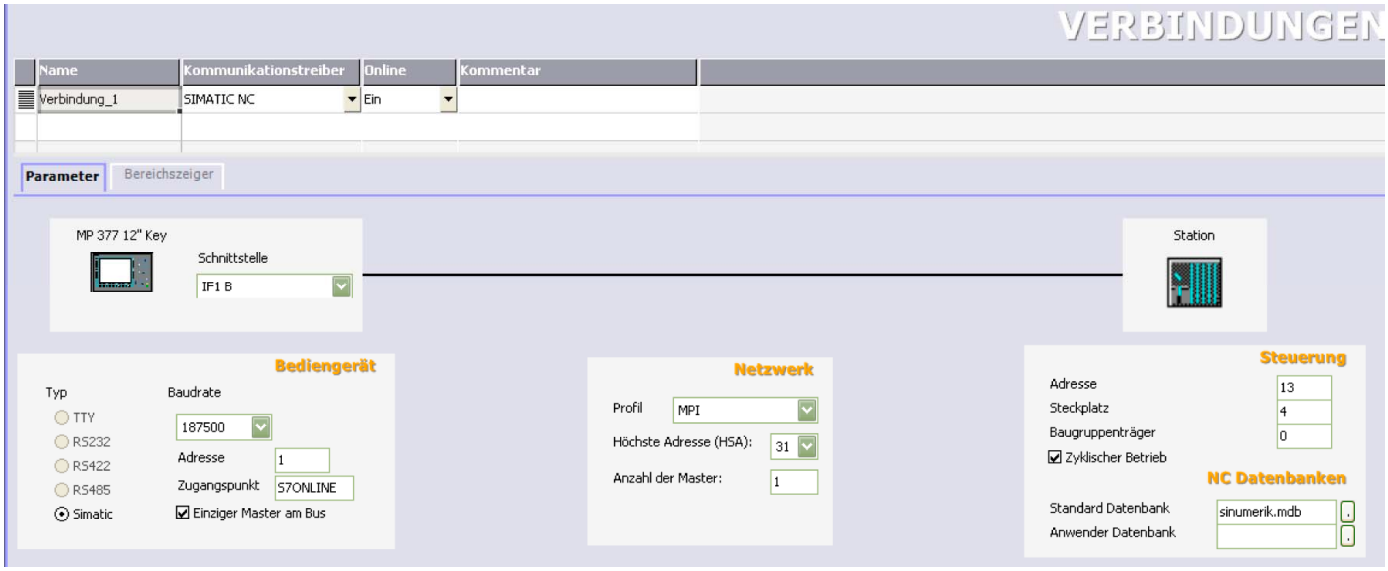


Figure 2-3 Connection parameters for CE HMI devices

You must make an entry in parameter "User database" if you want to configure the user data (GUDs).

See also

- Windows CE operator panel (Page 74)

2.1.4 M:N configuration with WinCC flexible (HMI PC devices)

Introduction

With WinCC flexible, it is possible to address a maximum of 16 connections (8 **SINUMERIK NC**, 8 **SINUMERIK PLC** -> 8 NCUs) on the HMI PC devices within a project (m:n).

General procedure

The following operating situations can be configured:

- Simultaneous access to data from one specific controller (but can be switched to all available controllers). This enables screens to be used several times. In this way, one single configuration can be used for 8 NCUs.
- Simultaneous access to data from 8 NCUs. Data from 16 connections can be displayed for each screen. This is not possible with the standard HMI user interface.
- A combination of the two operating scenarios above.

Rule

The operating situations can be configured in the following environments:

- HMI Advanced integrated procedure
  in this case, the operating procedures must match the m:n concept of the standard HMI user interface.
- Procedure without HMI Advanced (StandAlone)

See also

- Communications principle (Page 23)

HMI Advanced integrated

An operator panel with a WinCC flexible application, which offers a view, which can be switched to one specific NCU, is configured using the control name "machineswitch".

WinCC flexible projects configured in this way react to the HMI channel switchover menu, i.e., the standard HMI mechanism for channel switchover can also be used to switch the view in WinCC flexible screens. The HMI channel menu also activates active/passive switchover and displacement m:n concepts.

If you wish to simultaneously address n controllers per screen, this operator panel must match the m:n concept and be declared as a server (in the file netnames.ini). Select the following setting:
mmc_typ=0x40 (--> see also mmc2\netnames.bsp)
These n controls are then configured with the names defined in the file netnames.ini under WinCC flexible.

As there may be only one operator panel with server functionality, there may also only be one WinCC flexible operator panel with this property in a m:n system. Therefore, only one WinCC flexible operator panel can view several controllers simultaneously, all other operator panels have to work as an m:n "operator panel" via the "machineswitch" mechanism.

Note

Operating situations can also be mixed within an operator panel. One screen could run via "machineswitch" and be switched via the HMI channel menu, while other screens have a fixed connection to n controllers. It is even possible to have both operating procedures simultaneously within one screen. However, this is not recommended in the interests of a clear overview.

Note

The channel-oriented view of the standard HMI and the concept of active/passive switchover ensure that, under normal conditions, only one HMI can operate an NCU.

However, an operator panel with an active WinCC flexible application in the foreground is only switched to passive if the NCU viewed by the standard HMI is to be operated by a different HMI (case 1). If an additional and different NCU is observed on the WinCC flexible screen, and this second NCU is operated by a different HMI (case 2), the operator panel with the WinCC flexible application is not switched to passive (i.e., in certain circumstances, this second NCU can be operated by two HMIs). See screen below:

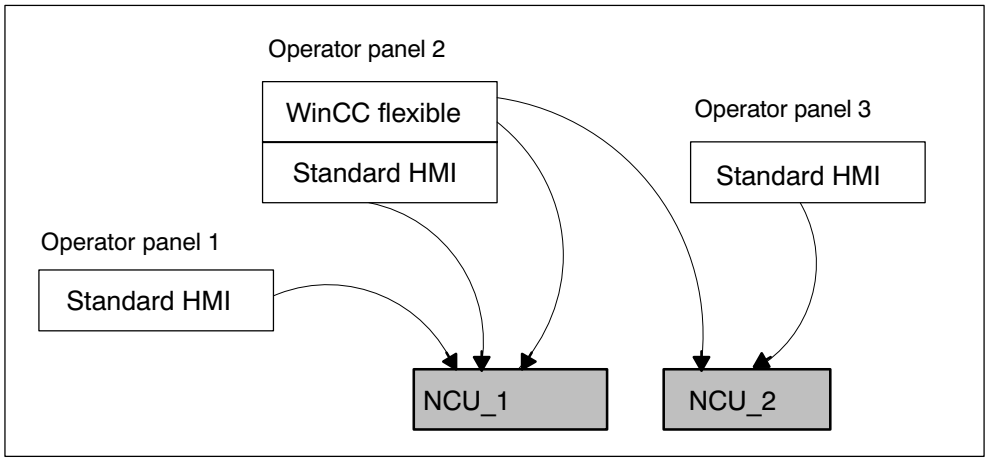


Figure 2-4 Active WinCC flexible application

Procedure in stand-alone mode

The HMI channel menu is not available in stand-alone mode. This means that the significant concepts of active/passive switchover and displacement cannot be used. It is for this reason that an m:n response is not possible.

Exceptions

An exception is operating case 1:n.

It is possible to display data from 8 NCUs/8 PLCs simultaneously on one screen. 16 controllers must also be configured with WinCC flexible ES.

A second exception is the configuration of switchable screens for a controller.

It is mandatory that the controller is then addressed via "machineswitch".

There is now an additional WinCC flexible function that enables "machineswitch" in this second case to be switched without the HMI channel menu.

The **SetNCController** function is called up to achieve this.

This function must be assigned one of the controller names defined in netnames.ini, with the prefix "net:", e.g., net:NCU_1.

This function can be used in conjunction with the configurable "Channel" key to configure a switchover menu, which corresponds to the HMI channel menu.

2.2 SINUMERIK tags

2.2.1 Control tags

Overview

Control tags of the relevant format can be defined for the communication channel set (= connection selected for tags).

The following control tags can be used for the communication channels **SINUMERIK_NC** and **SIMATIC_NC**:

- All tags of the current operator panel interface (OPI 7.2)
- All general, channel-specific and axis-specific machine and setting data
- User-defined control tags (GUD)

These tags can be configured using their symbolic name.

Any new or not yet symbolically available control tags can be configured directly with the standard address editor (for SIMATIC_NC only).

The following control tags can be used for communication channel **SINUMERIK_PLC**:

- Range of tags same as for channel S7 300/400
- For integrated working with WinCC flexible in STEP 7, you can select the PLC tags via the symbols defined in STEP 7.

Configuring OPI variables, machine and setting data

These data are available as default settings in the engineering system. They are made available via a preset database (sinumerik.mdb in connection editor to Sinumerik_NC or Simatic_NC).

2.2.2 Configuring GUD tags

2.2.2.1 GUD variable

Migrating GUDs from ProTool/Pro projects to WinCC flexible

This topic is discussed in detail in Section 1.4.3.

Making user data available

To make the user data of a particular controller available to a WinCC flexible project, it must be made available to WinCC flexible ES in the form of a database (*.mdb).

For each controller, you need:

- A user database ***.mdb** containing the user data (GUDs) at the time of configuration
- An NSK file (***.nsk**) containing the connection data during runtime (for StandAlone projects only --> Section 2.2.2.3)

These files are generated separately from WinCC flexible with the GUD tool.

2.2.2.2 Selecting the user database for dynamic user data

Introduction

The "sinumerik.mdb" database enables configuration of the standard (OPI) tags. This is pre-assigned with the **SINUMERIK NC** communication driver.

If you wish to configure additional user data, you must make them available to the configuration system using another database.

The database name must be entered in the **Controller** interactive form under **User Database** (see screen below).

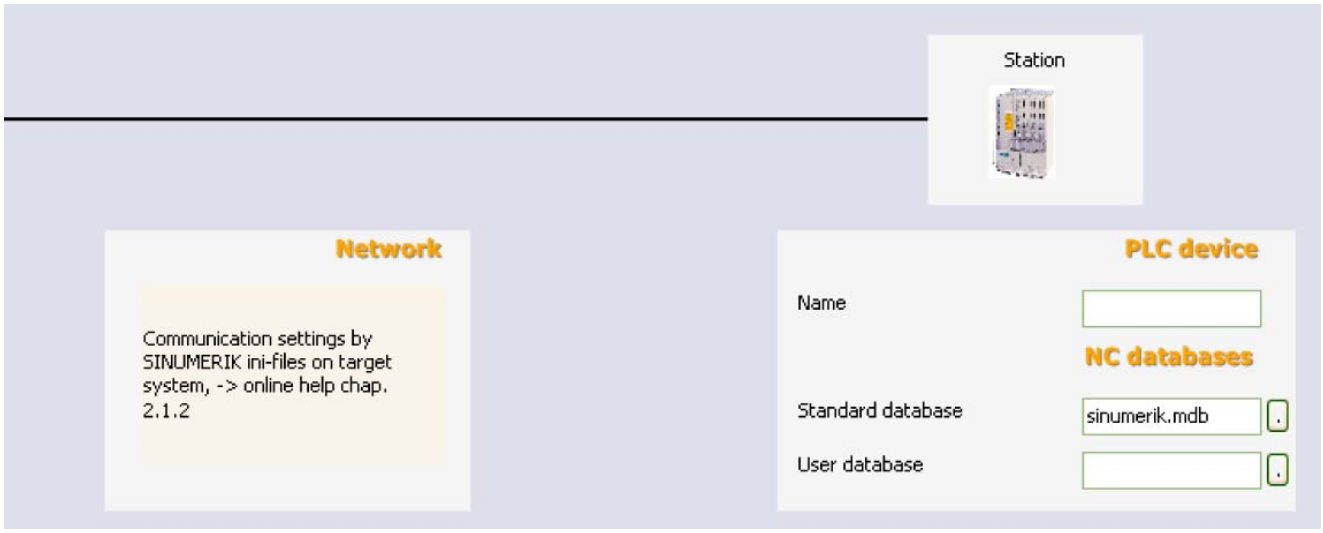


Figure 2-5 Controller

If multiple controllers require access to the user data, a control-specific user database must be entered for each communication driver.

If the file name is entered without a path, this file is searched for in the WinCC flexible installation directory.

Example

During configuration in WinCC flexible, you have created a connection to a SINUMERIK NC communication driver and you wish to configure a GUD tag for it. To do this, select the available GUD tag from the tag dialog box under the symbol editor (see screen below).

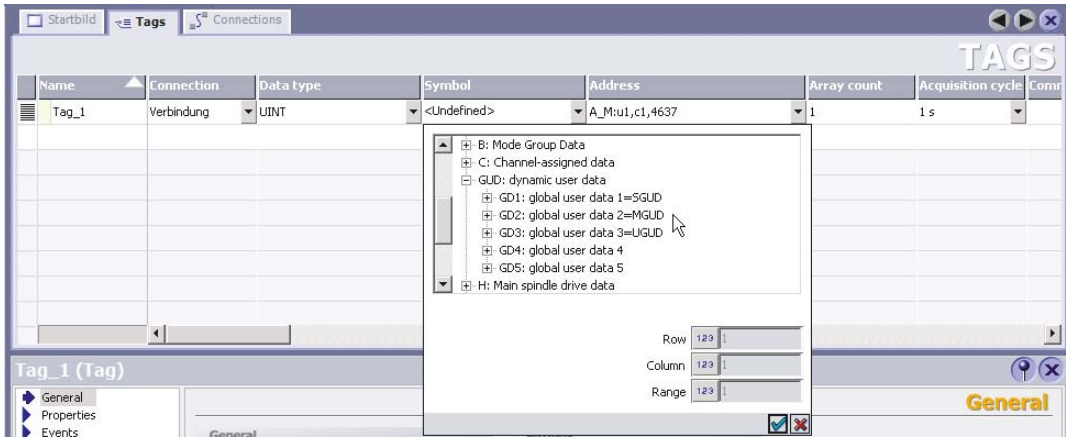


Figure 2-6 GUD tag symbol editor

2.2.2.3 Monitoring user data during runtime

1:1 connection

In HMI-integrated operation, access by WinCC flexible Runtime to the user data is always consistent. In this case, the NSK file generated by the GUD tools is not needed.

The NSK file generated by the GUD tool must be copied to the Add_On directory to ensure correct display of the user data in operation without HMI Advanced. In addition, a file user-nsk must be generated with the content:

CALL (<file name>.nsk)

If the user data of the configuration are changed after the user database is updated and the project generated (for example, by inserting new data at the beginning of a DEF file), and then loaded into the control, access by the WinCC flexible Runtime to the user database is no longer consistent. In this case, the user database must be updated with the GUD tool. Then the project must be regenerated and the NSK file copied into the Add_On directory again.

If when changes have been made, new user data are merely appended to the end of the DEF file and these GUDs are not intended for use in the WinCC flexible project, the NSK file does not have to be transferred again. The user database does not have to be updated nor the project regenerated.

Note

As from SINUMERIK HMI V7.2 SP2, access by WinCC flexible Runtime to the dynamic user data is always consistent even in operation without HMI Advanced. The NSK file generated by the GUD tool is therefore no longer needed. The user database does not have to be updated nor the project regenerated after the dynamic user data have been changed.

M:N connection

To display the user data correctly, the NSK files generated by the GUD tool (one for each control) must be copied to the Add_On directory. A file called user.nsk must also be generated in which all NSK files are listed as follows:

CALL (<Control1>.nsk)
CALL (<Control2>.nsk)
...
CALL (<ControlN>.nsk)

If the user data of the configuration are changed after the user database is updated and the project generated (for example, by inserteing new data at the beginning of a DEF file) and then loaded into the controls, access by the WinCC flexible Runtime to the user data is no longer consistent. In this case, the user data must be updated with the GUD tool. Finally, regenerate the project and copy the NSK files into the Add_On directory again.

Check Consistency

You can check user data consistency with the runtime system. The following entry must be made in the hmirtm.ini file (WinCC flexible RT directory):

All GUD blocks are included in the check

```
[GUD]
ConsistensCheck=ON
```

A selection (ON entries) of GUD blocks is included in the check

```
[GUD]
ConsistensCheck=ON
SGUD=OFF
MGUD=OFF
UGUD=OFF
GUD4=ON
GUD5=OFF
GUD6=OFF
GUD7=OFF
GUD8=OFF
GUD9=ON
```

2.2.2.4 WinCC flexible configuration of GUD arrays

Definition

GUD arrays consist of rows and columns:

Row 1/Column 1	Row 1/Column 2..	..	Row 1/Column n
Row 2/Column 1	Row 2/Column 2..	..	Row 2/Column n
..	..	..	..
Row n/Column 1	Row n/Column 2..	..	Row n/Column n

The table below shows the definition of the GUD arrays in the controller:

GUD definition	Example
DEF {NCK|CHAN} type name[rows, columns]	DEF NCK REAL REALNGUD[2,3]
	DEF CHAN REAL REALCGUD[2,3]
DEF {NCK|CHAN} STRING[bytes] name[rows]	DEF NCK STRING[8] NTEXT[5]

Where:

NCK	=	Global controller tag
CHAN	=	Channel-specific tag
Type	=	Tag type (BOOL, INT, REAL) Special case, type= STRING, comprised of bytes. It is not possible to define a field of STRINGs.

Name	=	Variable name
Rows	=	Number of rows
Columns	=	Number of columns

The table below shows the addressing of the array elements. Please note that addressing starts at 0 in the controller and at 1 in WinCC flexible.

GUD definition
Syntax: DEF NCK|CHAN <type> <tag>[<rows>,<columns]
Example: DEF CHAN REAL REALGUD[2,3]

Example

Controller	WinCC flexible configuration
<tag>[<row>,<column>]	C_GD4:u1,c<column>,<row>
REALGUD[0,0]	C_GD4:u1,c1,1
REALGUD[0,1]	C_GD4:u1,c2,1
REALGUD[0,2]	C_GD4:u1,c3,1
REALGUD[1,0]	C_GD4:u1,c1,2
REALGUD[1,1]	C_GD4:u1,c2,2
REALGUD[1,2]	C_GD4:u1,c3,2

2.2.3 Internal Tags

Introduction

Internal tags do not have any connection to the PLC.

Principle

Internal tags are stored in the memory of the HMI device. Therefore, only this HMI device has read and write access to the internal tags. You create internal tags, for example, in order to execute local calculations.

You can use all basic data types for internal tags. A detailed list of the data types can be found under "Basic data types."

2.2.4 Connecting a tag to another PLC

Introduction

In WinCC flexible, you can change the connection of a tag to a PLC at any time. This is, for example, necessary when you change the configuration of your system.

Depending on the PLC selected, you may need to modify the configuration of the tag. The tag properties which must be changed will be highlighted in color.

Requirements

The external tag, whose connection you wish to change, must already exist.

The connection to the PLC must already exist.

The Property view for this tag is open.

Procedure

1. Select the new "Connection" in the property view in the "General" group.

   The tag properties which you must change will be highlighted in color in the tag editor and in the property view.

2. Change all highlighted properties of the tag to suit the requirements of the new PLC.

Result

The external tag will now be saved in the new PLC.

2.2.5 Configuring a tag for address multiplexing

Introduction

Using address multiplexing, you can use a single tag to access a multitude of memory locations within the PLC's address range. You read and write to the addresses without defining a tag for each individual address. This is an efficient method for handling a large amount of data.

Requirements

A tag has been configured with one of the communication drivers **SINUMERK NC**, **SINUMERIK PLC** or **SIMATIC NC**.

The Property view for this tag is open.

Proceed as follows

1. Click on "Addressing" in the "Properties group" in the properties view.
2. Select a "Range". Depending on the range you selected, the system will display one or two additional selection lists.
3. In the first selection list, click on the arrow on the left edge and select the entry "multiplexed." The selection list has an arrow on the right edge to open the object list for tags.
4. Click on the selection list's right arrow and, in the object list, select the tag in which the memory location's address should be written in runtime.
5. If a second selection list has been displayed, repeat steps 3 and 4 for the second selection list.

Result

During runtime, the multiplex tag is used to access the memory location corresponding to the address currently found in the tag.

Note

The value in the memory location will only be read at the next update cycle for the addressed tag.

If, for example, you use a multiplex tag in a script, do not attempt to access contents of the memory location directly after changing it.

2.2.6 Configuring SINUMERIK tags

External tags

External tags can be used with the following SINUMERIK controller communication drivers:

- SINUMERIK NC
- SINUMERIK PLC

The tags are stored in a database (standard database). You define the database as a parameter when configuring the connection.

Note

The communication drivers must be configured under **Connections**.

Editors

If you have configured a connection to the **SINUMERIK NC** for the tag and move your cursor to the **Symbol** field, you can select the corresponding tag, e.g., **Active operating mode**, from those available in the database by opening the field (see screen below).

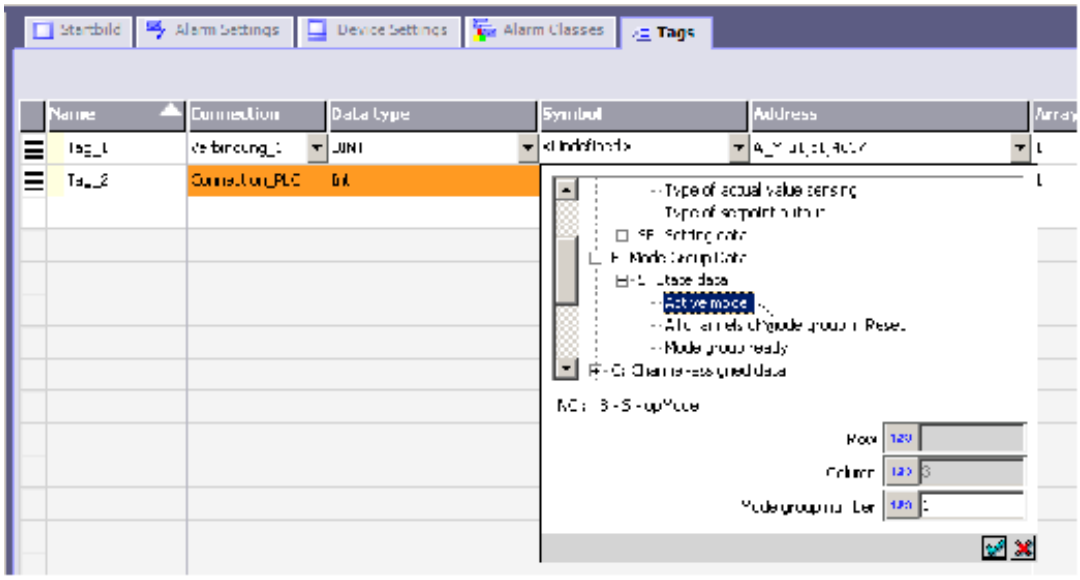


Figure 2-7 Symbol editor

If you have selected a symbolic tag, the address is displayed in the **Address** field.

The address can also be edited in the **Address** field.

The example shown (see screen below) relates to the **Actual value in the basic coordinate system** address, which you have selected from the database.

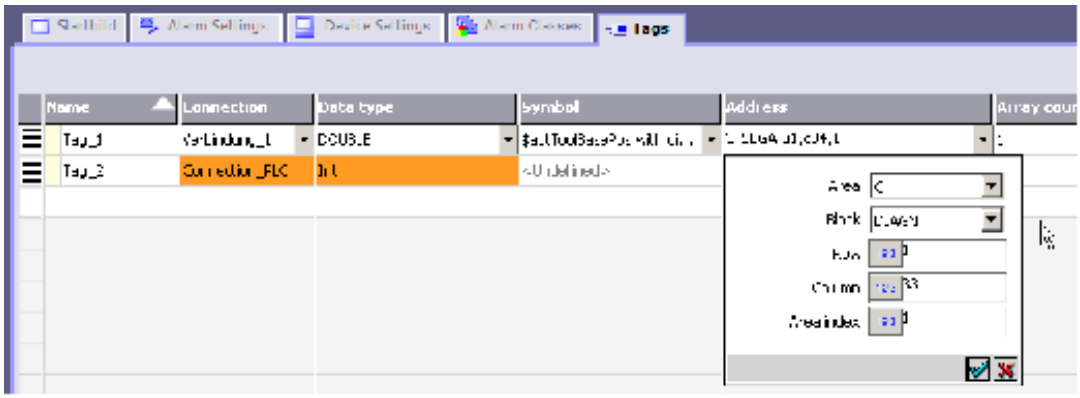


Figure 2-8 Address editor

Note

If you change the address, the symbolic address is adjusted in the **Symbol** field after the symbol editor has been activated.

Online help

If you select a SINUMERIK tag and hover the cursor over the selection box of the **tag** for a certain length of time, a context-sensitive help box for tags appears. Click on to expand the help box and access a detailed description of SINUMERIK tags.

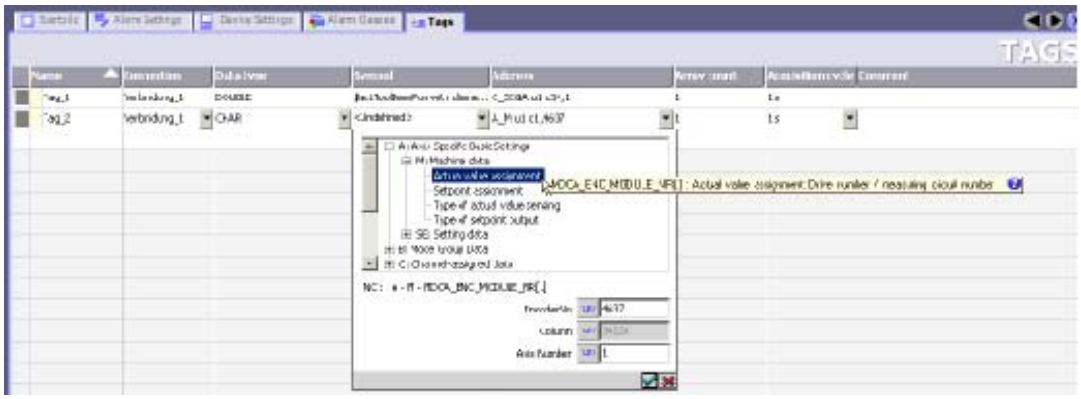


Figure 2-9 Context-sensitive help

2.3 SINUMERIK display objects

Introduction

Your configuration consists of objects, e.g., screens or tags. Screens are configuration templates for the user interface of your HMI device.

You can use the screen editor to configure display objects (e.g., input field or command button) on this configuration template.

Overview

With the WinCC flexible SINUMERIK HMI configuring package, you can configure the following SINUMERIK display objects in addition to the display objects offered by WinCC flexible:

- ASCII editor
- File display
- Input/output field
- Radio button
- Check box
- Combo box
- Toggle
- Scroll bar

The screen below shows the SINUMERIK display objects in WinCC flexible:

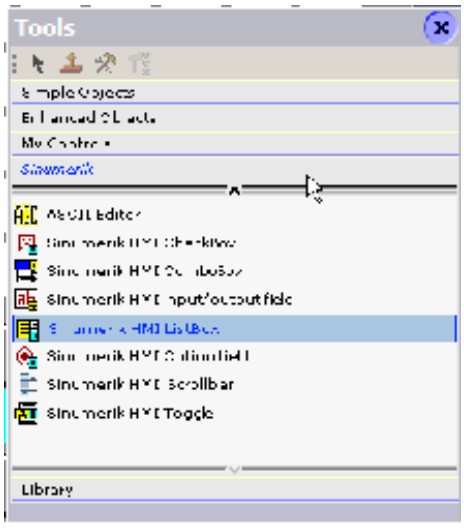


Figure 2-10 SINUMERIK display objects

See also

- SINUMERIK HMI ASCII editor (Page 41)
- SINUMERIK HMI input/output field (Page 56)
- SINUMERIK HMI radio button (Page 60)
- SINUMERIK HMI check box (Page 62)
- SINUMERIK HMI selection list (Page 64)
- SINUMERIK HMI toggle (Page 67)
- SINUMERIK HMI scroll bar (Page 69)
- SINUMERIK HMI list box (Page 70)

2.3.1 SINUMERIK HMI ASCII editor

Purpose

The **SINUMERIK HMI ASCII-Editor** display object allows you to edit part programs or subroutines, for example, during runtime.

The editor is merely a display control. To ensure this control is linked to a typical editor functionality, the editor functions must be configured on the relevant softkeys or function keys.

The following functions are available:

Table 2-1 SINUMERIK NC functions

Functions	Description	Parameters	Example
EditorCutBlock	The selected block is copied to a buffer and then deleted from the editor.	Display element name	Editor_1
EditorInsertBlock	The block on the clipboard is inserted at the cursor position.	Display element name	Editor_1
EditorCopyBlock	The selected block is copied to a clipboard.	Display element name	Editor_1
EditorSelectBlock	Switch editor to select mode (automatic selection when the cursor is moved).	Display element name	Editor_1
EditorPointSettings	The start number and increment for the "Renumber" function can be set here.	Display element name	Editor_1
EditorMoveToBlockNumber	Cursor is positioned on the specified block.	Display element name	Editor_1
EditorRenumber	The whole text is renumbered. The start number and increment can be changed using the Settings function.	Display element name	Editor_1
EditorOpen	The specified file is opened in the editor.	Display element name File	Editor_1 TEIL1.MPF

Functions	Description	Parameters	Example
EditorClose	Closes the editor.	Display element name	Editor_1
EditorSave	The current file is saved in the editor.	Display element name	Editor_1
EditorFind	Finds text in the editor	Display element name	Editor_1
EditorFindReplace	Finds and replaces text in the editor	Display element name	Editor_1
EditorChangeInsertMode	This function switches between Overwrite and Insert input modes.	Display element name	Editor_1
EditorFindNext	Find next instance of previous search text in the editor.	Display element name	Editor_1

See also

- Configuration example for file display plus ASCII editor (Page 54)

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI ASCII editor** display object:

General

Under **General**, you can define:

- If blocks should be numbered automatically
  - If so, a start value and an increment must be specified (e.g., "5" for 5, 10, 15, etc.).
- If scroll bars should be displayed in the view
- If character-by-character or block-by-block selection should be used when editing text

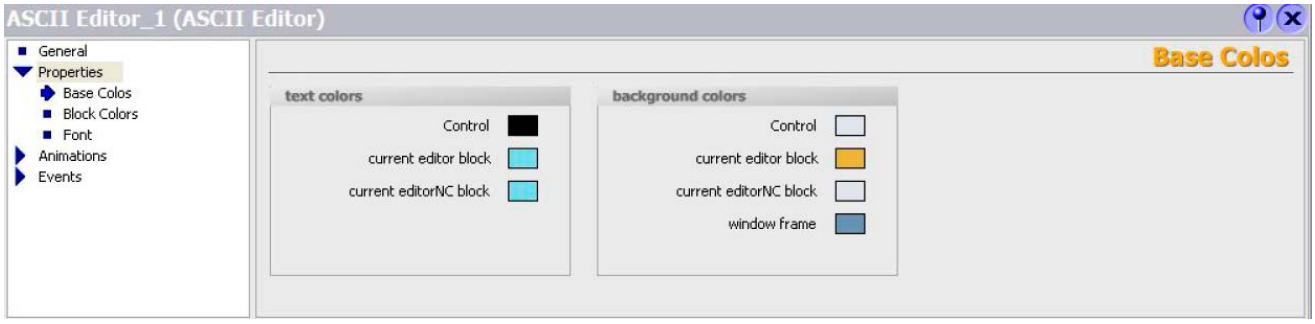
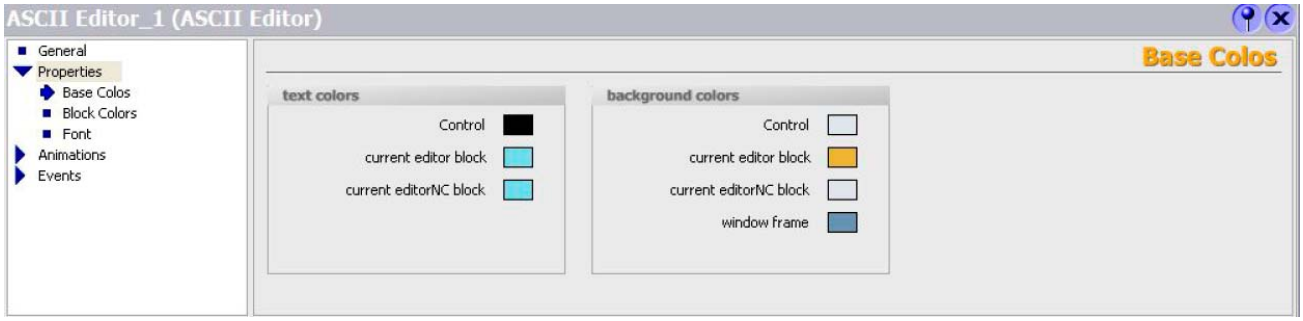
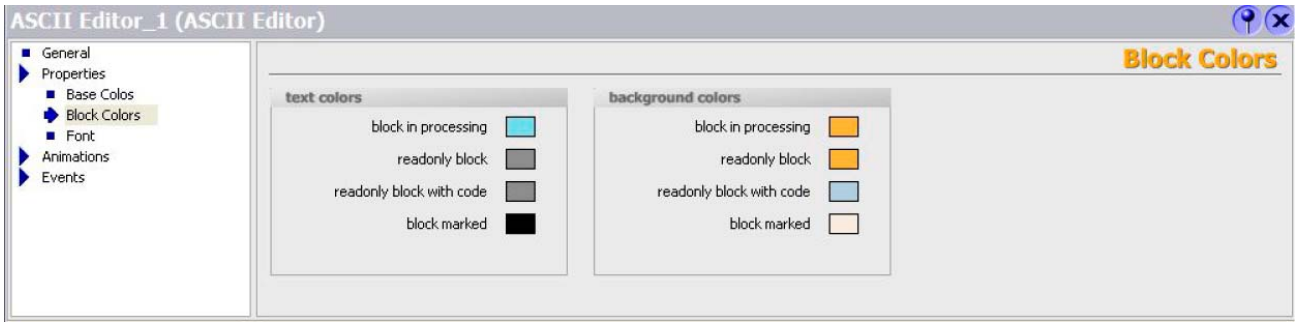


Figure 2-11 SINUMERIK ASCII editor general

Properties

You can define colors for different states under Properties --> Base Colors and Block Colors.





Events

Under **Events**, you can react to ASCII editor events with functions.

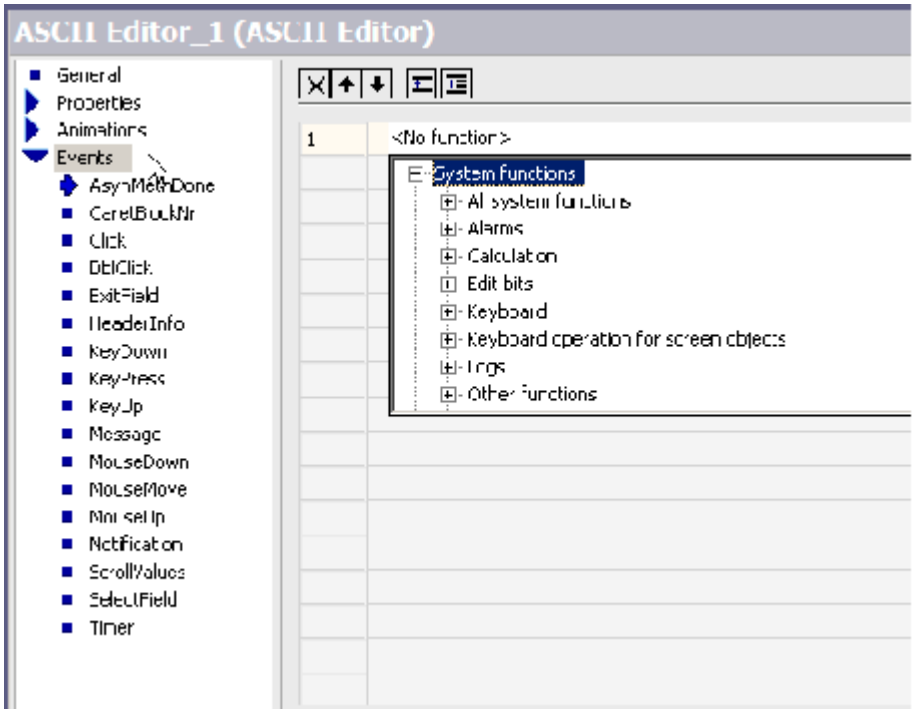


Figure 2-12 SINUMERIK ASCII editor events

Table 2-2 Events for the ASCII editor display object

Events	Description
Asynchronous method terminated	Do not use
Block number with cursor	Do not use
Click	The field was clicked with the mouse or a new entry was selected
Double-click	The field was double-clicked with the mouse
Exit field	Is triggered, if the editor loses the focus
Header	Do not use
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Notification	Do not use
Mouse key pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field
Mouse key released	A mouse key was released
Message	Do not use
Scroll bar value	Do not use
Field selected	Is triggered, if the editor retains the focus
Watchdogs	Do not use

Special features

The following table describes the effects of certain shortcuts in the configured ASCII editor during runtime:

Table 2-3 Shortcut

Shortcut	Function in ASCII editor
Ctrl+C	Copy block
Ctrl+V	Paste block
Ctrl+X	Delete block
Insert	Toggles insert mode
Toggle key (number 5 on numeric keypad)	Toggles selection mode

2.3.2 SINUMERIK HMI file display

Purpose

With image object **SINUMERIK HMI File Display**, you can configure a file display on your configuration template that resembles the Microsoft Explorer or the HMI Advanced file display.

The file display is merely a display control. To ensure this control is linked to a typical functionality, the data display functions must be configured on the relevant softkeys or function keys.

The following functions are provided:

Table 2-4 SINUMERIK NC functions

Functions	Description	Parameters	Example
FileViewInsert	The files previously selected using the "Copy file/directory" function are now inserted at the desired position.	Display element name	Filev_1
FileViewUnload	All files and directories selected in the file display are unloaded from the active NC.	Display element name	Filev_1
FileViewLoad	All files and directories selected in the file display are loaded into the active NC.	Display element name	Filev_1
FileViewDelete	All files and directories selected in the file display are deleted. Prompt = 0 -> No prompt Prompt = 1 -> With prompt	Display element name Prompt	Filev_1 0
FileViewNew	A new file or directory is created in the current directory.	Display element name	Filev_1
FileViewRename	The selected file/directory can be renamed. It is also possible to change the file type.	Display element name	Filev_1

Functions	Description	Parameters	Example
FileViewSetAttributes	The current attributes for the selected file are displayed.	Display element name	Filev_1
FileViewSelectProgram	The selected program is selected for execution. If the selection is to be made for the current channel, a tag with the content "-akt. Kanal" (current channel) must be configured. This tag must be transferred to the function as a parameter.	Display element name	Filev_1
FileViewSaveState	The directories opened and the current selection are saved and made available for the "Restore state" function. The file name of the selected file is written to the specified tag and made available for the "Open file" function.	Display element name	Filev_1
FileViewRestoreState	The file display state is updated.	Display element name	Filev_1
FileViewCopy	The current selection is temporarily stored. The copy procedure is started with the FileViewInsert function.	Display element name	Filev_1

Note

The following functions are only permitted within the data management function:
FileViewUnload
FileviewLoad
FileviewNew
FileViewRename
FileViewSetAttribute

See also

- Configuration example for file display plus ASCII editor (Page 54)

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI ASCII file display** display object:

General

Under **General**, you can define:

- HMI Advanced or Windows display (see table below)

  Recommendation: When viewing the NC data management (default setting), display mode "HMI Advanced" should be selected.
- Maximum directory depth (only in HMI Advanced display)
- Width ratio percentage of the display in the main window (Windows display only)
- Columns, by selecting the column descriptions to be displayed and defining their column description (column display), e.g., width and alignment.
  Under **Display**, you can set the columns to an HMI Advanced display or a Windows display. If you selected HMI Advanced display, for example, this is displayed as a digit string in the access authorization column or as a DOS file with extension in the file name column.
- Sorting according to various columns. Click with the mouse on the column headers to resort files dynamically during runtime.

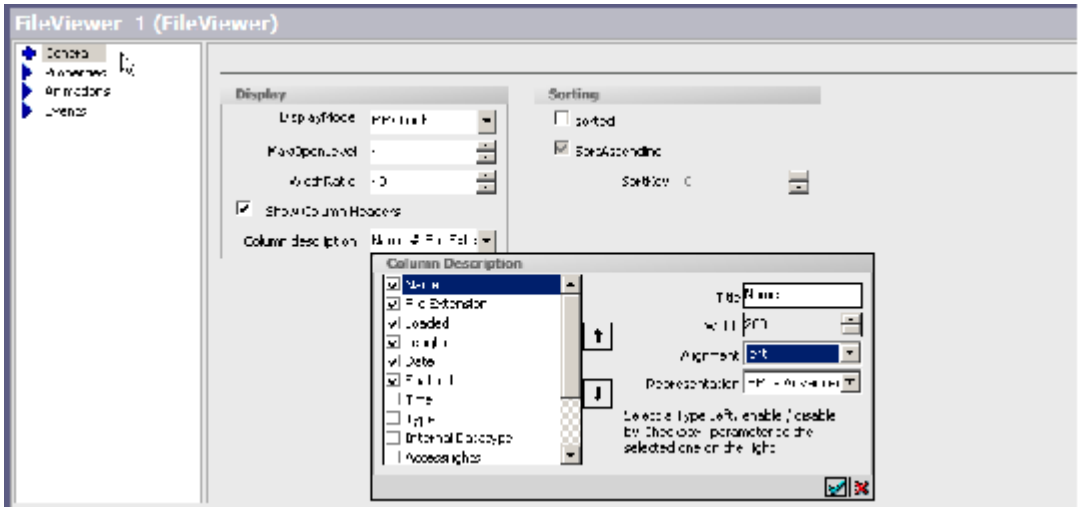
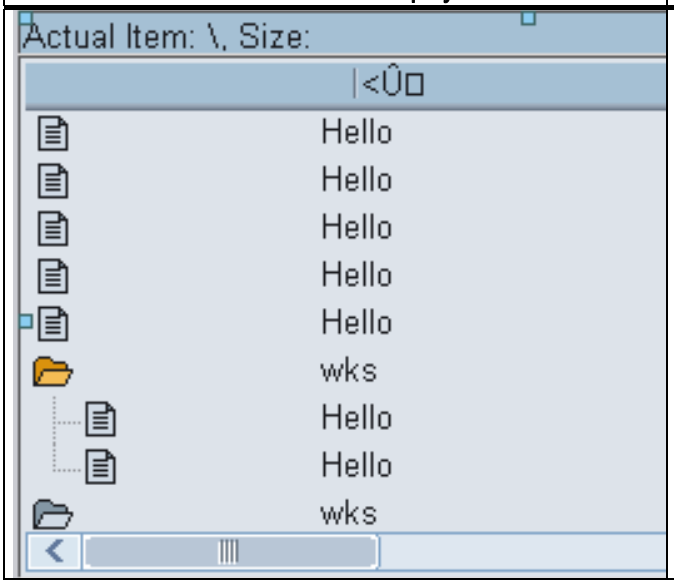
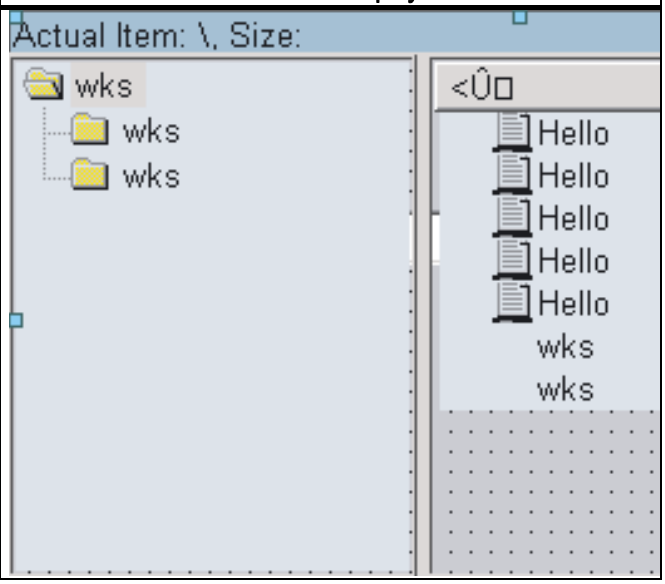


Figure 2-13 SINUMERIK file display, general

The following table shows the different file displays (HMI Advanced on the left and Windows on the right):

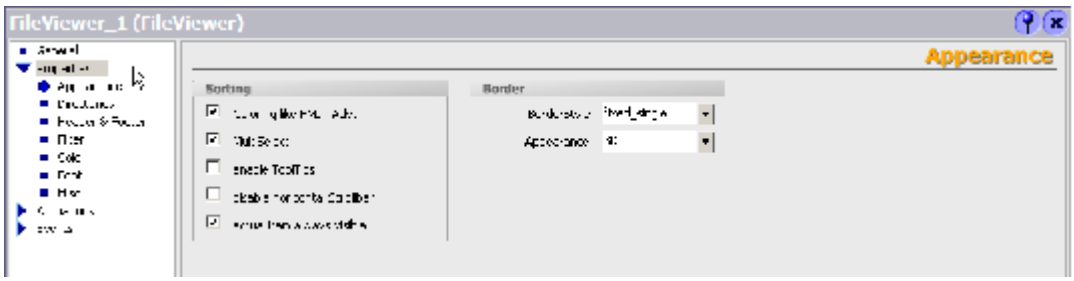
Table 2-5 Display

HMI Advanced display	Windows display
	

Properties

Under **Properties** , you can make global settings for the appearance of the SINUMERIK HMI file-display display object:

- **Appearance**
  Under **Appearance**, you can select either a 3D display or a surrounding frame. **Coloring as HMI Advanced** means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.



SINUMERIK file display, properties

- **Directories**
  Under **Directories**, you can define possible access to drives for the directories to be displayed.
  In the **Start Directory** value input field, a tag can be made available, to which the start directory or a parameter can then be written as a character string. For example, for SINUMERIK HMI Advanced, the start directory is:
  - "MC://domain/C:" on the hard disk or
  - "MC://domain/hd–nc" in the data management.

- **Password system**
  Under Password System, you configure the password levels for the relevant data carriers, whose data can be viewed during runtime. These carriers are the data management tree, the local hard drives or the connected networks.

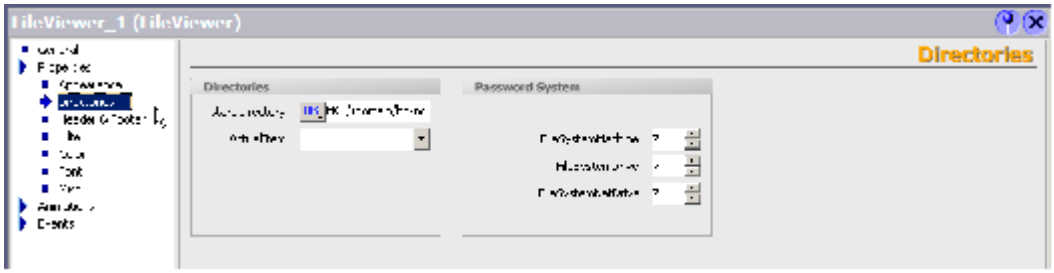


Figure 2-14 SINUMERIK file display, directories

- **Header/footer**
  You can define a header and a footer for the file display (see screen below).
  In addition to the parameters, which you can assign to the header and footer, you can also add any text you wish. Everything marked with "&[...]" refers to a wildcard, which points to text replaced during runtime.

  – Example:
    &[directory]/&[file] free memory:&[memory]
    becomes: MC://domain/hd–nc/mpf.dir/hallo.mpf free memory:
    Hard disk: 805 MB NCU: 1.25 MB

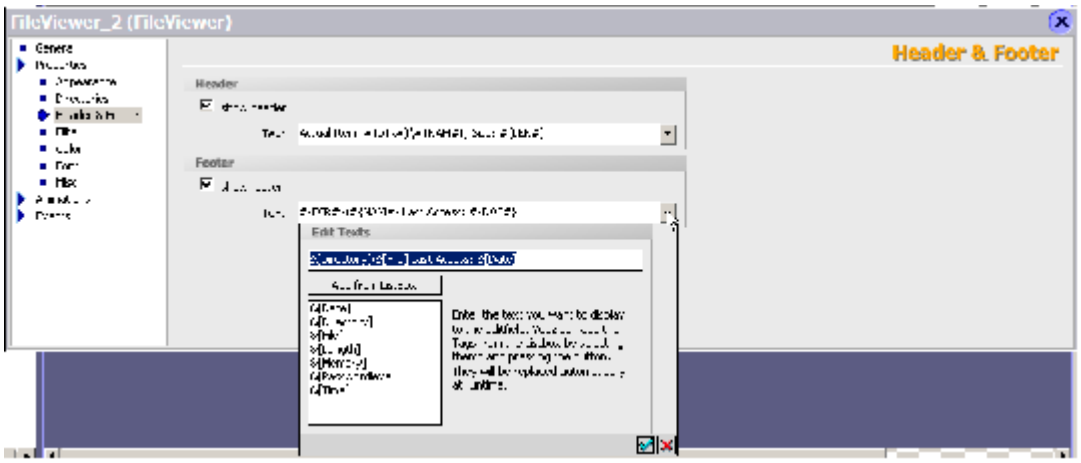


Figure 2-15 SINUMERIK file display, header and footer

- **Filter**
  Under **Filter**, you can configure various display filters. The display filters (e.g., "following file applications only") define the files/directories to be displayed in the file display (e.g., Windows) during runtime.

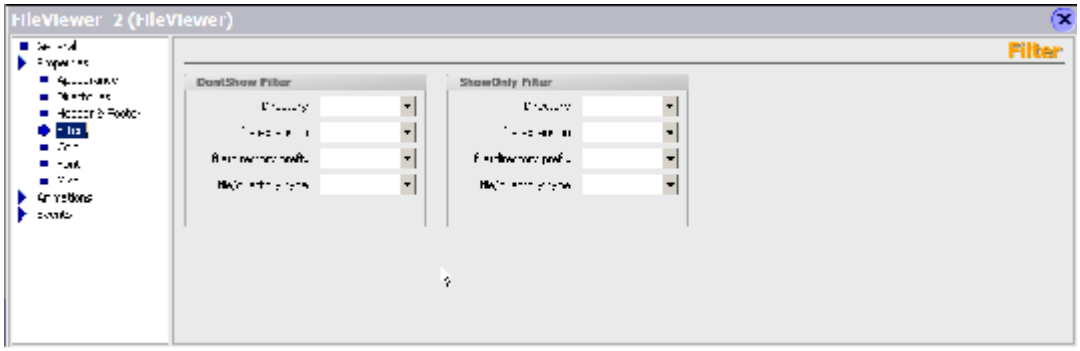


Figure 2-16 SINUMERIK file display, filter

The following negative filters are available for selection:

Table 2-6 Negative filter

Filter	Functions	Content of filter entry (examples and explanation)
"All apart from directory"	Dynamic suppression of directories	• MPF • SPF • Winnt • ... Hide directories with names "MPF", "SPF" and "Winnt".
"All apart from entries beginning as follows"	Dynamic suppression of certain file prefixes	• Operate • Form • ... Hide all files and directories of the form "Operate*.*" and "Form*.*".
"All apart from type"	Dynamic suppression of certain file types	• *.MPF • *.DOS • ... Hide all files of file type "*.MPF" and "*.DOS".
"All apart from file extension"	Dynamic suppression of certain file extensions	• MPF • TXT • DIR • ... Do not show files with extensions "MPF", "TXT", and "DIR".

The following positive filters are available for selection:

Table 2-7 Positive filter

Filter	Functions	Content of filter entry (examples and explanation)
"Following directories only"	Explorer only shows the directories that are transferred.	• MPF • SPF • Winnt • .... Only show directories with names "MPF", "SPF", and "Winnt".
"Entries beginning as follows only"	Explorer only shows elements with the relevant file prefixes that are transferred.	• Operate • Form Only show files and directories of the form "Operate*.*" and "Form*.*".
"Following types only"	Explorer only shows elements with the relevant file types that are transferred.	• *.MPF • *.DOS • ... Only show files with file type "*.MPF" and "*.DOS".
"Following file extensions only"	Explorer only shows elements with the relevant file extensions that are transferred.	• MPF • TXT • DIR • ... Only show files with extensions "MPF", "TXT", and "DIR". Note: Elements with no extension (e.g., directories outside of data management, typically) are ignored by this filter, so are still shown. If a directory has an extension (e.g., Winnt.400), it is not ignored by this filter.

Events

Under **Events**, you can react to file display events with functions.

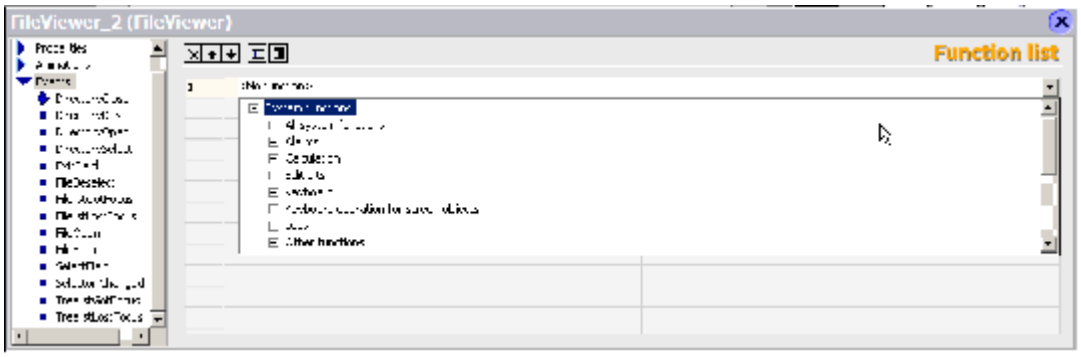


Figure 2-17 SINUMERIK file display, events

Table 2-8 Events for file-display display object

Event	Description
Close directory	Expanded representation of a directory closes up, directory name in "current entry". Close a directory by pressing the Return/Enter key, clicking with the mouse click or pressing the left cursor control key.
Deselect directory	Cursor exits directory
Open directory	Directory is expanded, directory name in "current entry". Open a directory for display by pressing the Return/Enter key, clicking with the mouse click or pressing the right cursor control key.
Select_directory	Cursor moves to a directory, directory name is in "current entry".
Exit field	The FileViewer – ActiveX is deactivated, i.e., it is no longer the active control on this page (it loses the focus).
Deselect file	Cursor exits file
Select ListCtrl	ListCtrl is activated (by changing from TreeCtrl or activating the control).
Deselect ListCtrl	ListCtrl is deactivated (by changing to TreeCtrl or deactivating the control).
Activate file	Activate a file by pressing the Return/Enter key or double-clicking with the mouse.
Select file	Cursor moves to a file, file name is in "current entry"
Select field	The FileViewer – ActiveX is activated, i.e., it becomes the active control on the current page (it takes on the focus).
Change selection	Triggered if the current element changes its selection state.
Select TreeCtrl	TreeCtrl is activated (by changing from ListCtrl or activating the control).
Deselect TreeCtrl	TreeCtrl is deactivated (by changing to ListCtrl or deactivating the control).

Special features

If you are using a PC keyboard or a mouse, certain special shortcut keys or mouse shortcuts have been designed to have the same effect as on the OP010, OP012, or OP015 (operator panel controller).

The tables below describe how entering PC keyboard codes or clicking with the mouse affects the configured file display:

For example, if you press "!" on the PC keyboard, the operator panel interprets this code as "Shift+CursorUp".

Table 2-9 Keyboard code

Keyboard code	Function in NC file management
Left arrow	Jump to higher-level directory or Close current directory.
Right arrow	Jump to first lower-level directory or Open current directory.
Up arrow	Select a higher-level file/directory

Keyboard code	Function in NC file management
Down arrow	Select a lower-level file/directory
Home	Jump to first element
End	Jump to last element
Page Up	Scroll up one page
Page Down	Scroll down one page
Tab / Shift–Tab	Change from tree to selection list and vice versa, step enable of focus in tab order with external elements
Shift + up/down arrow	Multiple selection (selection expansion)
Shift + Home	Select up to first element
Shift + End	Select up to last element
Space bar	Same as clicking with the left mouse button
Return	Open a selected file/directory
Select (ASCII code 12)	Toggle selection for the current element
$	Same as Shift + CursorDown (MMC operator panel)
!	Same as Shift + CursorUp (MMC operator panel)
Insert	Deselect all selected elements, end selection expansion
Backspace	Go back one sub-level
Letter	Jump from current position to next element with this initial letter
Minus	Close the current directory (MMC file management)

Table 2-10 Mouse click

Action	Function in NC file management
Left-click tree +/-	Directory/drive expanded or contracted in tree, no change to selection list
Left-click tree entry	Directory/drive expanded in tree, content of directory displayed in the selection list
Left-click selection list	Element is selected
Left-click HMI	Element is selected
Double-click tree	Directory/drive expanded or contracted in tree, content of directory displayed in selection list
Double-click selection list	Selects file to be edited
Double-click HMI	
Right-click tree	Display popup menu (not first version)
Right-click selection list	Display popup menu (not first version)
Right-click HMI	
Shift + left-click tree	Same as left-click tree
Shift + click selection list	Range between previous selection and current element is selected

Action	Function in NC file management
Shift + click HMI	
Ctrl + left-click tree	Same as left-click tree
Ctrl + left-click selection list	Current element is also selected
Ctrl + left-click HMI	

2.3.3 Configuration example for file display plus ASCII editor

Overview

Configuration examples are supplied with the SINUMERIK HMI WinCC flexible configuration software. After installation, these examples are stored in the ".\samples\SINUMERIK" WinCC flexible installation directory.

The "OP012_FileMan.hmi" example contains the configuration of a file display with an ASCII editor.

The following screens were created during configuration:

- "screen_viewer" for file display and
- "screen_editor" for ASCII editor

The examples in the following sections describe a practical way of interconnecting the file display and ASCII editor components.

See also

- SINUMERIK HMI file display (Page 45)
- SINUMERIK HMI ASCII editor (Page 41)

File display - "screen_viewer"

The "screen_viewer" is the configuration start screen.

A SINUMERIK HMI file-display display object has been created on the configuration template in the "screen_viewer" screen.

The display object name is then the display element name, which is required during configuration for NC file display functions.

In the example, the display element name for the SINUMERIK HMI is "viewer".

When configuring this "viewer" display object, please note:

- Functions for generating the file display
  The **Generate screen** event triggers configuration of the **Restore state** NC file management function.
  This is achieved by positioning the cursor in an empty area of the "screen_viewer" screen configuration template, using **Events** to select the **Generate screen** event in the display object editor and assigning the **Restore state** function to this event.
  This function is required in order to reset the current selection and the previously opened

directories in the file display following a screen change. The display element name is then "viewer".

- Functions for exiting the file display
  The **Activate file** event generates configuration of the following functions:
  - **Save state** and
  - **Screen change** to "screen_editor" screen
    The **Activate file** event is triggered when you press Return/Enter or double-click with the mouse in the file management during runtime.
    The **Save state** function saves the opened directories and the current file display selection.
    The name of the selected file is written to the "Cursor" tag and can be accessed using the **Open file** function (described in "ASCII editor - "screen_editor"").
    The **Change screen** function is used to select the "screen_editor".

ASCII editor - "screen_editor"

A SINUMERIK HMI ASCII editor display object has been created using the configuration template in the "screen_editor" screen.

The display object name is then the display element name, which is required during configuration for functions.

In the example, the display element name for the SINUMERIK HMI ASCII editor is "editor".

When configuring this "editor" display object, please note:

- Functions for generating the ASCII editor screen
  The **Generate screen** event triggers configuration of the **Open NC editor** function.
  This is configured by moving the cursor to the "screen_editor" screen configuration template, using **Events** to select the **Generate screen** event in the display object editor and assigning the **Open NC editor** function to this event.
  The display element name is then "editor" and the tag name is "cursor", which contains the name of the file to be opened.
- Functions for exiting the ASCII editor
  In the example, a **Close** command button was configured for exiting the ASCII editor.
  The **Close NC editor** function of the "editor" display element and the **Screen change** function of the "screen_viewer" screen are assigned to this key.
  This ensures that you can return to the file display from the ASCII editor.

2.3.4 SINUMERIK HMI input/output field

Purpose

You can use display object **SINUMERIK HMI input/output field** during runtime to

- Enter values on the HMI device, which are transferred to the controller or
- Display current controller values on the HMI device.

Values can be input/output numerically, alphanumerically or symbolically.

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI input/output field** display object:

General

Under **General** , you can make global settings for the appearance of the SINUMERIK HMI input/output field display object:

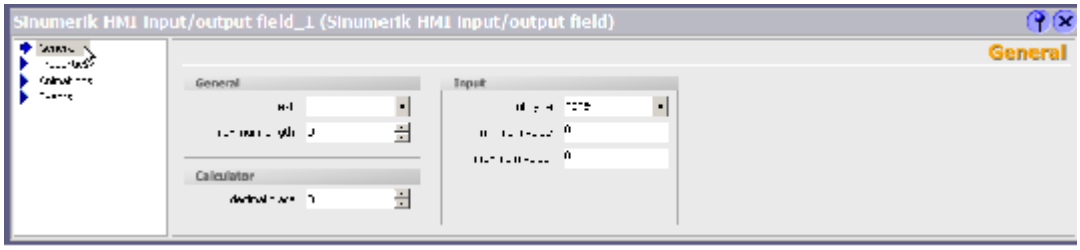


Figure 2-18 SINUMERIK input/output field, general

- **General**
  Among other things, you can enter the text, which identifies the input/output field during runtime.
  A text length defined as "0" is not restricted.
- **Entry**
  You can select the character of a value to be input/output during runtime in the **Entry Type** selection list.

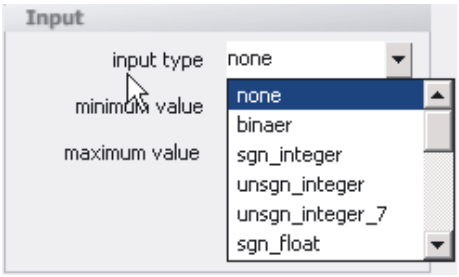


Figure 2-19 SINUMERIK input/output field, entry type

You can choose from the following character types:

Table 2-11 Types in entry selection list

Type	Letter/number	Pocket calculator
0 - none	All characters permitted (default)	
1 = binary	0 1	Yes
2 = sgn integer	0 - 9 signed	Yes
3 - unsgn int	0 - 9	Yes
4 - unsgn int 7	0 - 7	Yes
5 = sgn float	0 - 9 <period> signed	Yes
6 - unsgn float	0 - 9 <period>	Yes
7 = unsgn isxdigit	A - F a-f 0 - 9	Yes
8 = alpha	A - Z a - z _	
9 - alpanum	A - Z a - z 0 - 9	
10 - csym	A - Z a - z 0 - 9 _	
11 - csym & spec	A - Z a - z 0 - 9 _ = ( ) $ <period> * /	
12 - ProgSpec	1. and 2nd position: A <to> z a <to> z Remaining positions: A <to> z a <to> z 0 <to> 9 _ or 1st position: L Remaining positions: A <to> z a <to> z 0 <to> 9 _	
13 - ProgLab	1. and 2nd position: A <to> z a <to> z Remaining positions: A <to> z a <to> z 0 <to> 9 _ or 1st position: _ Remaining positions: A <to> z a <to> z 0 <to> 9 _	
14 - VarSpec	1. and 2nd position: A <to> z a <to> z Remaining positions: A <to> z a <to> z 0 <to> 9 _ = ( ) $ <period> * /)	

Type	Letter/number	Pocket calculator
15 - BlockNo	1. Position: N <or>: Remaining positions 0 <to> 9	
16 - float ext	Floating point with exponential display 0-9 <period> - e E	Yes

Properties **minimum value** and **maximum value** are verified when the value is accepted (e.g. with Return). Any errors are output via the event **Communicate**. Error handling must be configured in WinCC flexible. The incorrect value continues to be displayed in the image object until it is changed by the configuration.

- **Pocket calculator**
  You can enter the number of decimal places (max. 9) for the pocket calculator in the **Decimal places** input field. The pocket calculator can be called up during runtime with "=". Depending on the corresponding mode, calculations are made in either binary, integer, floating point or hexadecimal format.
  The + - / * ( ) operators are supported, e.g., "34-30/2+5".
  Completing the entry with "m" or "i" converts the value to a metric or U.S. customary value. (the value is divided or multiplied by 25.4).
  Entry of an integer or floating number is restricted to a maximum of ten characters (integer places). With binary input, up to 24 characters are permitted. For floating numbers with an exponent, the number of characters is restricted to the maximum number of characters, which can be entered in the calculation window.
  The arithmetic expression can contain up to 70 characters.

Options

Under **Options** , you can make global settings for the appearance of the SINUMERIK HMI input/output field display object:

Figure 2-20 SINUMERIK input/output field

This is, for example **Show screens**.

Click the **Show screens** field to load the bitmaps for that display object during runtime.

A bitmap displaying the key symbol, which can be used, for example, to make entries in the selected field, then appears.

If you have selected the **Show screens** field, you are still able to configure whether the bitmap should be displayed as a text symbol.

The bitmaps used are searched for in the SINUMERIK subdirectories:

- ..\user
- ..\oem
- ..\add_on
- <in the directory, in which the Control is currently located>
- The first bitmap to be found is used for the display.

The table below shows the possible keyboard image bitmaps:

Table 2-12 Bitmap

Key	Standard keyboard	Scr obj	Text
Edit key	Insert key		
Input key	Enter key		

The text symbols can be changed by the user.

Edit key: McEdit_t.bmp
input key: Mccben_t.bmp

HMI coloring means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.

Events

Under **Events**, you can react to events with functions (see screen below).

Figure 2-21 SINUMERIK input/output field, events

Table 2-13 Events for the SINUMERIK HMI input/output field display object

Event	Description
Calculator off	Calculator disappears
Calculator on	Calculator appears
Modification	Editing box or entry has changed

Event	Description
Click	The field was clicked with the mouse or a new entry was selected
Communicating	Do not use
Double-click	The field was double-clicked with the mouse
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Mouse key pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field
Mouse key released	A mouse key was released
Select field	Field is selected

2.3.5 SINUMERIK HMI radio button

Purpose

You can use the **SINUMERIK HMI radio button** display object to activate a field during runtime.

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI radio button** display object:

General

Under **General**, you can make global settings for the appearance of the SINUMERIK HMI radio button display object:

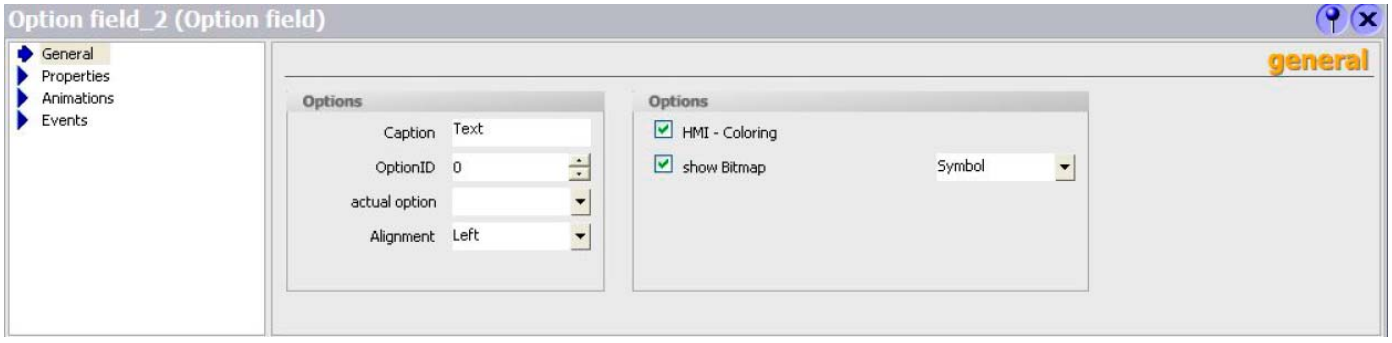


Figure 2-22 SINUMERIK radio button, general

- **General**
  The value specified for the tag for the current option is shown in the field **OptionID**, on condition that the field **actual option** (current option) is activated.
  The tag of a group is shown in **actual option**. All option fields that indicate the same tag belong to this group.
  In this case the value that is written in the target tag when the corresponding option field is activated is shown in **OptionID**.
- **Options**
  For example, **Show screens**.
  Click the **Show screens** field to load the bitmaps for the display object during runtime.
  A bitmap displaying the key symbol, which can be used, for example, to make entries in the selected field, then appears.
  If you have selected the **Show screens** field, you are still able to configure whether the bitmap should be displayed as a text symbol.
  The bitmaps used are searched for in the SINUMERIK subdirectories:
  - ..\user
  - ..\oem
  - ..\add_on
  - <in the directory, in which the Control is currently located>
  - The first bitmap to be found is used for the display.

The table below shows the possible keyboard image bitmaps:

Table 2-14 Bitmap

Key	Standard keyboard	Scr obj	Text
Select key	Number 5 on numeric keypad		

The text symbol can be changed by the user.

Select key: Mctogl_t.bmp

HMI coloring means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.

Events

Under **Events**, you can react to events with functions.

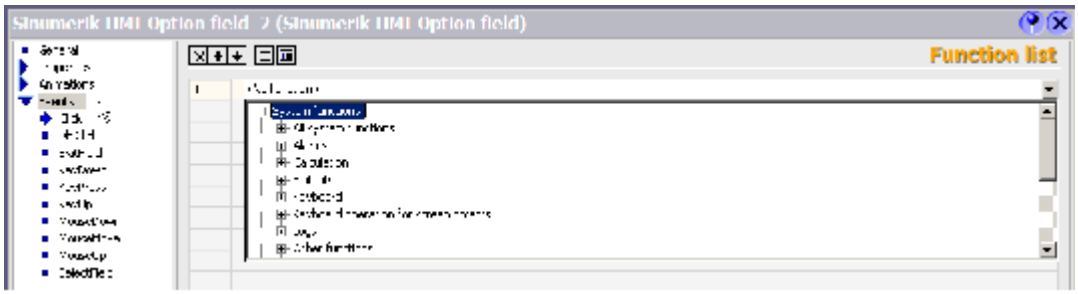


Figure 2-23 SINUMERIK radio button, events

Table 2-15 Events for the SINUMERIK HMI radio button display object

Event	Description
Click	The field was clicked with the mouse or a new entry was selected
Double-click	The field was double-clicked with the mouse
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Mouse key pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field
Mouse key released	A mouse key was released
Field selected	Field is selected

2.3.6 SINUMERIK HMI check box

Purpose

You can use the **SINUMERIK HMI check box** display object to activate/deactivate a field during runtime.

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI check box** display object:

General

Under **General**, you can make global settings for the appearance of the SINUMERIK HMI check box display object:

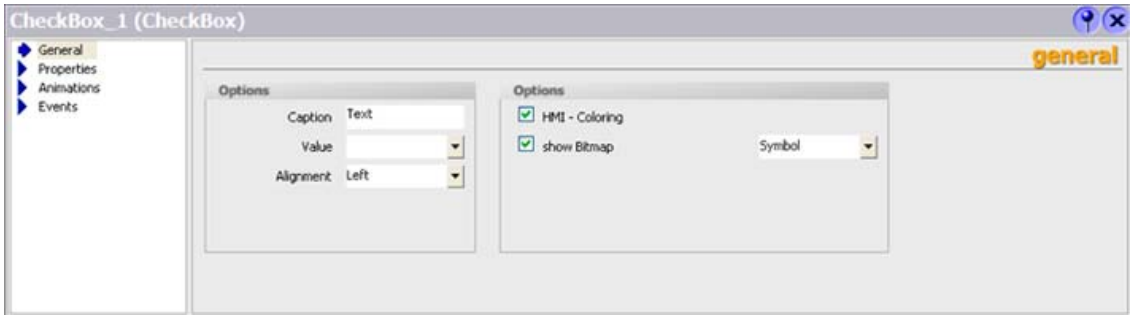


Figure 2-24 SINUMERIK check box, general

- **General**
  If you enter a value of "-1" in the Value input field, this option field is then active during runtime.
- **Options**
  For example, **Show screens**.
  Click the **Show screens** field to load the bitmaps for the display object during runtime. A bitmap displaying the key symbol, which can be used, for example, to make entries in the selected field, then appears.
  If you have selected the **Show screens** field, you are still able to configure whether the bitmap should be displayed as a text symbol.
  The bitmaps used are searched for in the SINUMERIK subdirectories:
  - ..\user
  - ..\oem
  - ..\add_on
  - <in the directory, in which the Control is currently located>
  - The first bitmap found is used for the display.
    The table below shows the possible keyboard image bitmaps:

Table 2-16 Bitmap

Key	Standard keyboard	Scr obj	Text
Select key	Number 5 on numeric keypad		

The text symbol can be changed by the user.

Select key: Mctogl_t.bmp

HMI coloring means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.

Events

Under **Events**, you can react to events with functions.

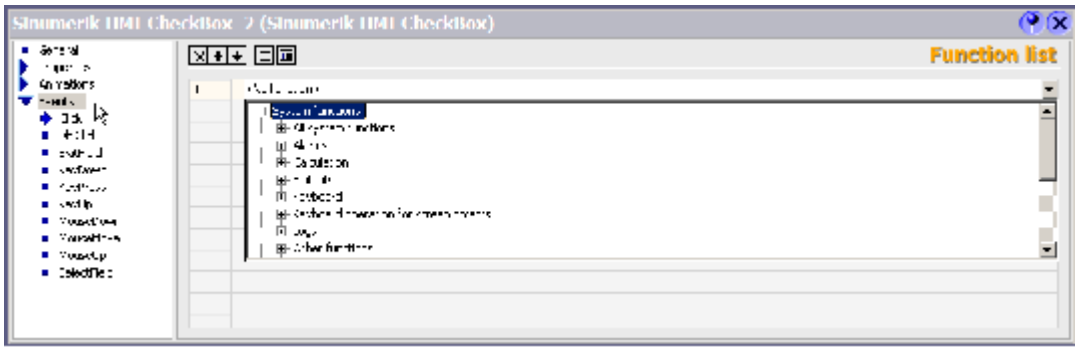


Figure 2-25 SINUMERIK check box, events

Table 2-17 Events for the SINUMERIK HMI check box display object

Event	Description
Click	The field was clicked with the mouse or a new entry was selected
Double-click	The field was double-clicked with the mouse
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Mouse key pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field
Mouse key released	A mouse key was released
Field selected	Field is selected

2.3.7 SINUMERIK HMI selection list

Purpose

You can use the **SINUMERIK HMI selection list** display object during runtime to

- Select from various values, which you defined during the configuration period (combo box) or
- Enter values, which are transferred to the controller.

Values can be input/output numerically, alphanumerically or symbolically.

See also

- SINUMERIK HMI selection list (Page 64)

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI selection list** display object:

General

Under **General**, you can make global settings for the appearance of the SINUMERIK HMI selection list display object:

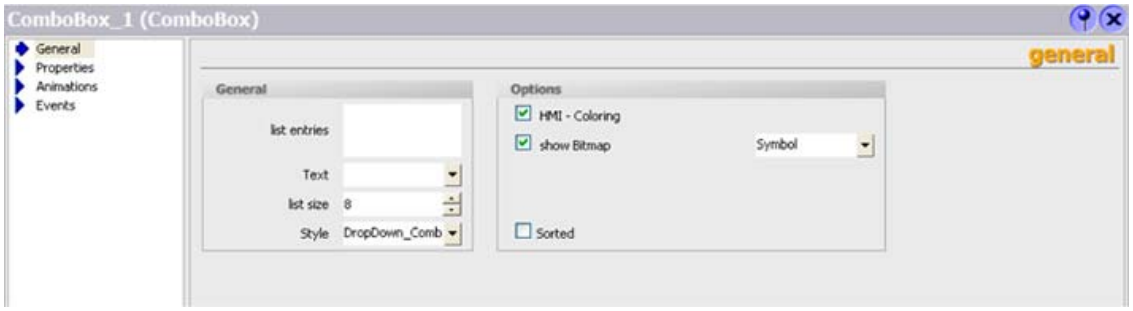


Figure 2-26 SINUMERIK selection list, general

- **General**
  This includes, for example, **List entries** (e.g., "first value") that you wish to make available for selection in the SINUMERIK HMI selection list.
  In the **Type** field, you can choose between these configurations (see screen below):
  - **Drop_DownCombo**,
    a combined input/selection field.
    This allows you to either select an existing value from the list or to enter a new value.
  - **DropDown_List**,
    selection field only.
    A value can be selected from the list.

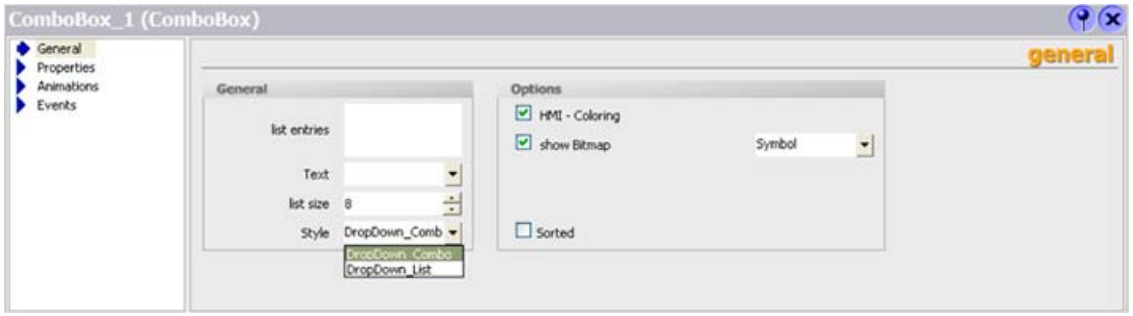


Figure 2-27 SINUMERIK selection list, type

- **Options**
  For example, **Show screens**
  . Click the **Show screens** field to load the bitmaps for the display object during runtime.
  A bitmap displaying the key symbol, which can be used, for example, to make entries in the selected field, then appears.
  If you have selected the **Show screens** field, you are still able to configure whether the bitmap should be displayed as a text symbol.
  The bitmaps used are searched for in the SINUMERIK subdirectories:
  - ..\user
  - ..\oem
  - ..\add_on
  - <in the directory, in which the Control is currently located>
  - The first bitmap found is used for the display.
    The table below shows the possible keyboard image bitmaps:

Table 2-18 Bitmap

Key	Standard keyboard	Scr obj	Text
Edit key	Insert key		
Input key	Enter key		

The text symbols can be changed by the user:

Edit key: McEdit_t.bmp
input key: Mccben_t.bmp

HMI coloring means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.

Events

Under **Events**, you can react to events with functions.

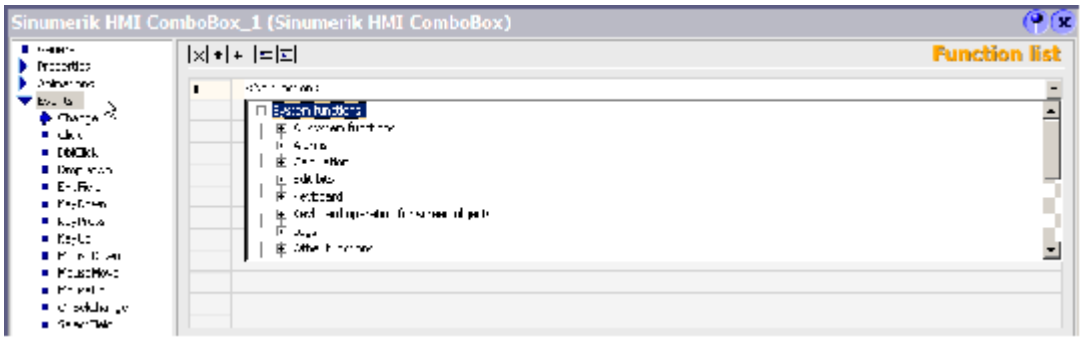


Figure 2-28 SINUMERIK selection list, events

Table 2-19 Events for the SINUMERIK HMI selection list display object

Event	Description
Modification	Do not use
Click	The field was clicked with the mouse or a new entry was selected
Double-click	
Open	List has been opened
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Mouse pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field.

Event	Description
Mouse released	A mouse key was released
Change selection	Selection made from list
Field selected	Field is selected

2.3.8 SINUMERIK HMI toggle

Purpose

You can use the **SINUMERIK HMI toggle** display object during runtime you can choose between various values (a maximum of five is recommended, to retain a clear overview), which you defined during the configuration period (toggle box).

See also

- SINUMERIK HMI toggle (Page 67)

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI toggle** display object:

General

Under **General**, you can make global settings for the appearance of the SINUMERIK HMI toggle display object:

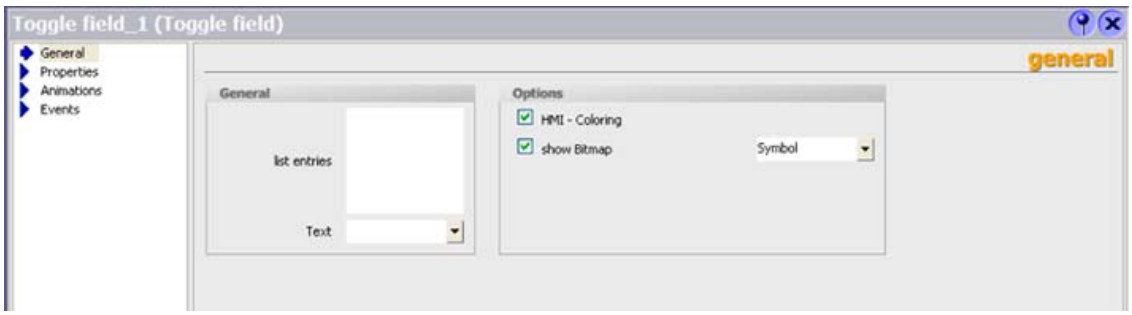


Figure 2-29 SINUMERIK toggle, general

- **General**
  For example, entries, which you wish to make available for selection in SINUMERIK HMI toggle.
- **Options**
  For example, **Show screens**
  . Click the **Show screens** field to load the bitmaps for the display object during runtime. A bitmap displaying the key symbol, which can be used, for example, to make entries in

the selected field, then appears.
If you have selected the **Show screens** field, you are still able to configure whether the bitmap should be displayed as a text symbol.
The bitmaps used are searched for in the SINUMERIK subdirectories:

– ..\user
– ..\oem
– ..\add_on
– <in the directory, in which the Control is currently located>
– The first bitmap found is used for the display.
The table below shows the possible keyboard image bitmaps:

Table 2-20 Bitmap

Key	Standard keyboard	Scr obj	Text
Select key	Number 5 on numeric keypad		

The text symbol can be changed by the user.

Select key: Mctogl_t.bmp

HMI coloring means that the display object fetches the colors for the foreground and background from the system and the transferred colors are not used.

Events

Under **Events**, you can react to events with functions.

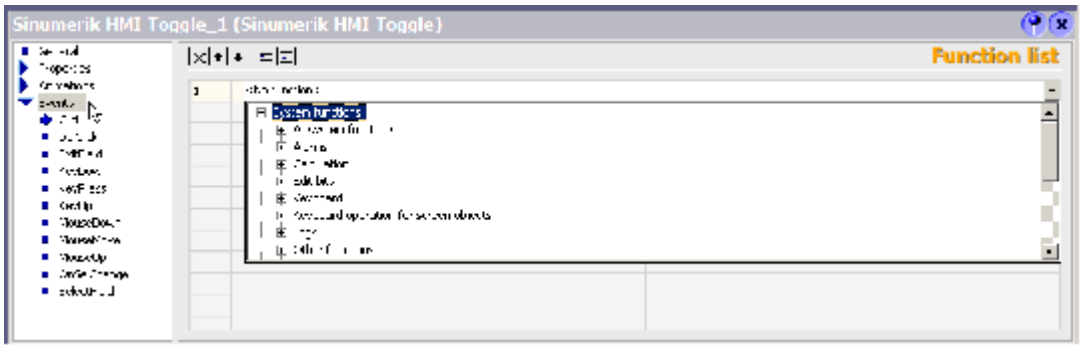


Figure 2-30 SINUMERIK toggle, events

Table 2-21 Events for the SINUMERIK HMI toggle display object

Event	Description
Click	The field was clicked with the mouse or a new entry was selected
Double-click	The field was double-clicked with the mouse
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard

Event	Description
Key released	A key was released
Mouse key pressed	A mouse key was pressed
Mouse moved	The mouse was moved over the field
Mouse key released	A mouse key was released
Change selection	Selection made from list
Field selected	Field is selected

2.3.9 SINUMERIK HMI scroll bar

Purpose

You can use the **SINUMERIK HMI scroll bar** display object to steplessly count up a tag during runtime.

See also

- SINUMERIK HMI scroll bar (Page 69)

Screen editor

The screen editor provides the following options for configuring the **SINUMERIK HMI scroll bar** display object:

General

Under **General**, you can make global settings for the appearance of the SINUMERIK HMI scroll bar display object:

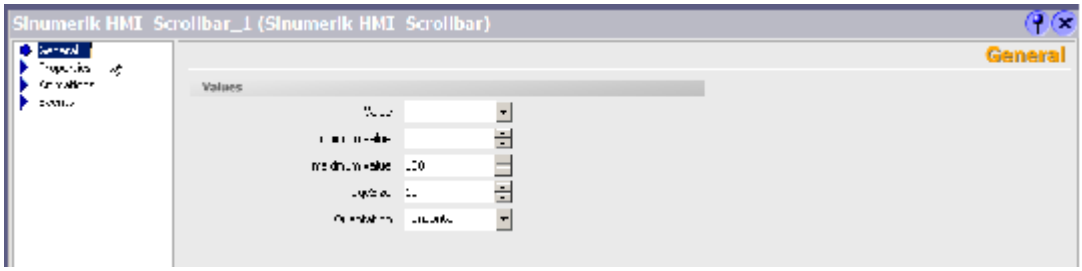


Figure 2-31 SINUMERIK scroll bar, general

Events

Under **Events**, you can react to events with functions.

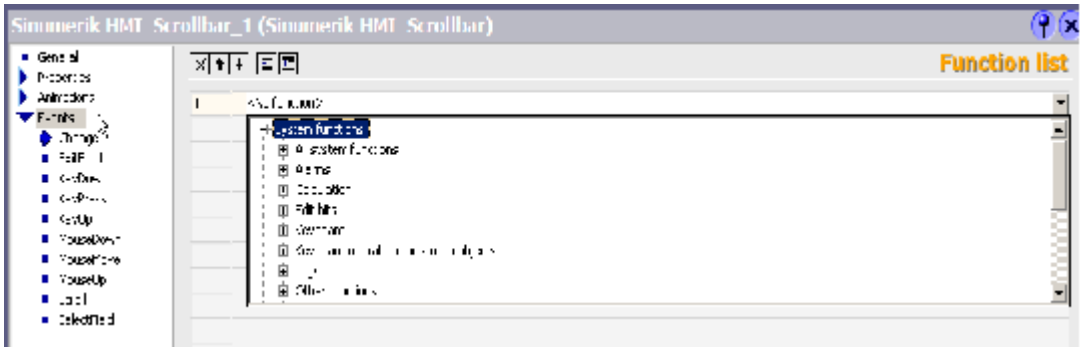


Figure 2-32 SINUMERIK scroll bar, events

Table 2-22 Events for the SINUMERIK HMI scroll bar display object

Event	Description
Change	Do not use
Exit field	Field is exited
Key pressed	A key was pressed
Key value	A character was entered by the keyboard
Key released	A key was released
Mouse key pressed	
Mouse moved	
Mouse key released	
Scrolling	The middle scroll bar has been clicked
Field selected	Field is selected

2.3.10 SINUMERIK HMI list box

Purpose

The **SINUMERIK HMI list box** display object enables you to select from a list of defined values during runtime. These values can then be transferred to another field.

Note

The **list box** configuration largely corresponds to that of the **selection list**.

See also

- SINUMERIK HMI selection list (Page 64)

2.4 PI services as configurable functions

Purpose

In WinCC flexible, events (e.g., "Key pressed") can be linked with defined functions. If an event should occur during operation, the function triggers a certain action on the HMI device or in the controller.

General procedure

In general, the functions can be used to

- Generate the configuration with reference to a specific process (e.g., branching from one screen to another).
- Control the process (e.g., setting control parameters).
- Carry out online system settings on the HMI device (e.g., changing interface parameters).

As a rule, functions and events are linked to a certain object. For example, a function with event "Press key" is not triggered by every key, but by one specific key.

Objects, which can be linked to functions, include:

- Function keys and command buttons
- Fields
- Figures
- Variables

The number of functions offered is, therefore, dependent on the HMI device and on the situation.

When using the SINUMERIK HMI WinCC flexible configuration software, SINUMERIK-specific and WinCC flexible functions are offered.

These additional functions for SINUMERIK controllers are combined under the generic term "SINUMERIK NC".

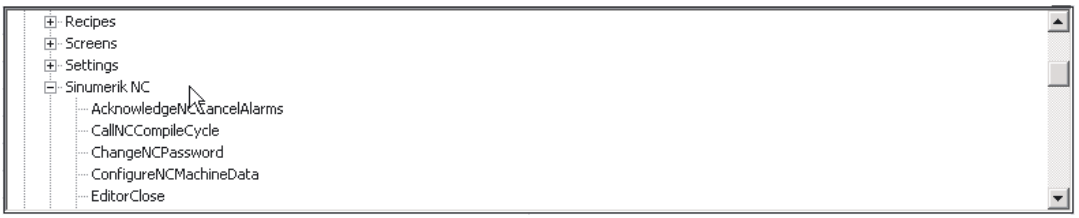


Figure 2-33 SINUMERIK NC functions

The following PI services are available as a configurable function:

- ResetNC
- LogoffNC
- LogonNC
- ChangeNCPassword
- AcknowledgeNCCancelAlarm
- ConfigureNCMachineData
- CallNCCompileCycle
- SetNCController
- SetNCUserFrame
- StartNCPIService (can be parameterized with any string)

Table 2-23 SINUMERIK NC functions

Functions	Description	Parameters	Example
StartNCPIServic	Any PI service can be sent to the NC using this function. This must be described by a parameter string in accordance with the OPI documentation.	PI parameter string	PI_START (0d0d, 201, I_N_MPF_DIR/_N_BSP_MPF,_N_SELECT)
AcknowledNCCancelAlarms	All pending cancel alarms are acknowledged.	None	
CallupNCCompileCycle	This function enables you to parameterize (max. 5 parameters) and call up a compile cycle.	Channel No. of compile cycle Parameter 1 Parameter 2 Parameter 3 Parameter 4 Parameter 5	1 2 PAR1 PAR2
ConfigureNCMachineData	This function activates all machine data of the classification NEW_CONF. The CLASS parameter allows for a more exact classification. Currently, however, only value 1 is supported.	Class	1
ChangeNCPassword	A new NC password can be defined using this function. You are prompted to enter the new password and the password level in a dialog box.	None	
LogoffNC	This function resets the NC password. Following this, the keylock switch is in control.	None	

Functions	Description	Parameters	Example
LogonNC	A new NC password is entered using this function. The password is either entered using a dialog box or transferred as a fixed parameter. Dialog box= 1: A dialog box appears during runtime Dialog box= 0: No dialog box appears during runtime -> The password is defined during the configuration period.	NC password	XYZ
SetNCUserFrame	A previously defined zero offset is activated in the NC using this function.	Channel	1
SetNCController	A switchover menu that corresponds to the HMI channel menu can be configured using this function and the configurable Channel key.		

2.5 Windows CE operator panel

Overview Windows CE operator panel

An HMI CE device communicates with a SINUMERIK controller via the SIMATIC NC communication driver and SIMATIC S7 300/400.

SIMATIC NC

The following configuration options are available with the HMI SINUMERIK WinCC flexible configuration software for **SIMATIC NC**:

- Reading/writing of OPI and GUD tags
- **SINUMERIK NC part program**
  You can use the **SINUMERIK NC part program** display object to display an overview of part programs and select them.
- Functions
  - For the **SINUMERIK NC part program** display object (e.g., navigating and selecting in the list directory)
  - For a selection of PI services, to be executed in the NCK
  - For the current block display
- SINUMERIK NC messages for NCK alarms and messages.

SIMATIC S7 300/400

The additional configuration options below are available with the HMI SINUMERIK WinCC flexible configuration software for **SIMATIC S7 300/400**:

- Start PI service and stop PLC

2.5.1 Read/write tags (OPI)

Introduction

You can configure SINUMERIK OPI tags for a Windows CE operator panel in conjunction with SINUMERIK controls.

General procedure

OPI tags can be selected in the symbol editor (see screen below).

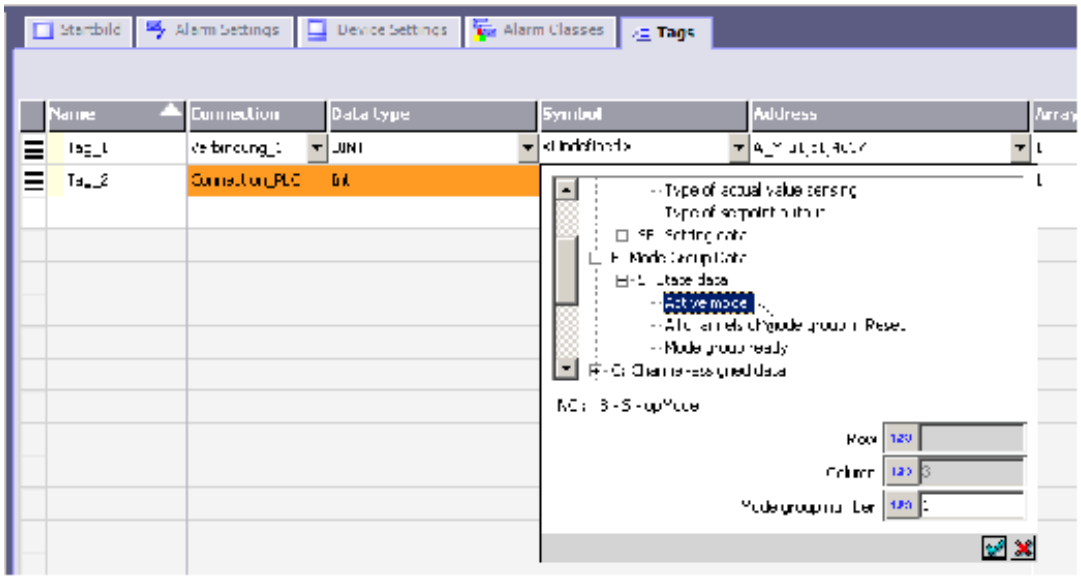


Figure 2-34 Symbol editor

Possible configurations

If you have configured a **SINUMERIK NC part program** display object, the following additional configurations are recommended to provide a clearer overview.

Example

Configure an additional output field, in which the tag will show the path on the controller where the program is stored. To do this, you must select the following addresses in the symbol editor from the OPI file:

- The **workpiece name** (path) is obtained via address C_SPARPP:u1, c4, 1
- The **program name** is found in address C_SPARPP:u1, c5, 1

2.5.2 SINUMERIK NC part program display object

Introduction

You can use the **SINUMERIK NC part program** display object to display an overview of part programs or workpieces during runtime and to select them.

During configuration, you can specify, for example, which font the image object will have.

SINUMERIK NC part program

The screen below shows the **SINUMERIK NC part program** display object during configuration.

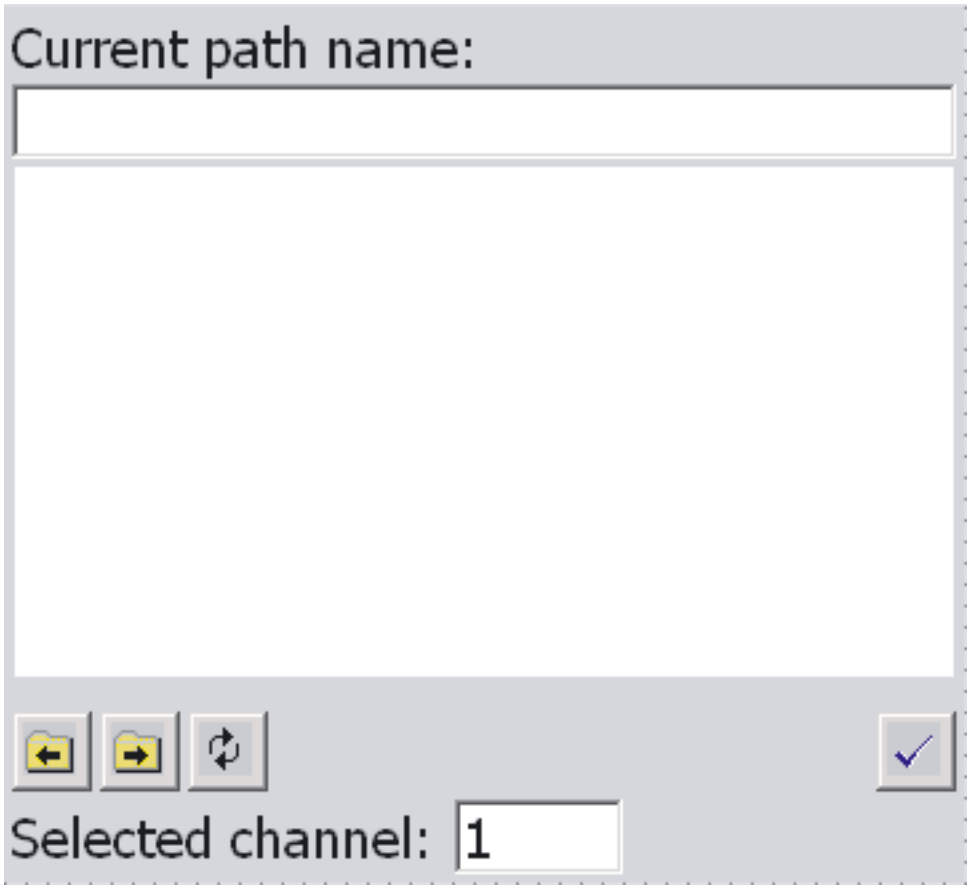


Figure 2-35 SINUMERIK NC part program

User interface

An alphanumeric input/output field shows the current path name and allows data from the relevant directory (e.g., MPF, SPF, etc.) to be selected and displayed in the list below it, by changing the path.

Note

If you select a path directly from SINUMERIK during runtime, i.e., not from the **SINUMERIK NC part program** display object, this path is not shown in the display object.

The part program names or the directory structure are displayed line-by-line in the list.

During runtime, command buttons can be used to: .

Table 2-24 Command buttons for the display object

Button	Functions
	Navigate and select in the list directory
	The selected part program/workpiece is selected on the NC for editing.

Button	Functions
	Update directories in the list box
Click/touch	When the directory name is selected, the value is transferred to the current path name field and the corresponding subdirectory is displayed in the list.

To make the part program overview functionally complete, you can configure function keys or configured command buttons with NC part program functions (e.g., **Navigate and select in the list directory**).

Screen editor

The screen editor provides the following options for configuring the *SINUMERIK NC part program* display object:

General

Under **General**, you can make the following settings for part program selection:

- **Connections**
  You select the controller using its symbolic name in the **Connections** selection list.
- **Path**
  A defined path name (max. 29 characters) can be set in the **Path** input field.
- **Path Tag**
  You select the connected path tag from the **Path tag** selection list.
- **Channel**
  A defined channel (1 thru 10) can be set in the **Channel** input field.
- **Channel tag**
  You select the connected channel tag in the **Channel tag** selection list.
- **Editable**
  The paths or channels can be set to editable/non-editable by selecting the "Editable" field.

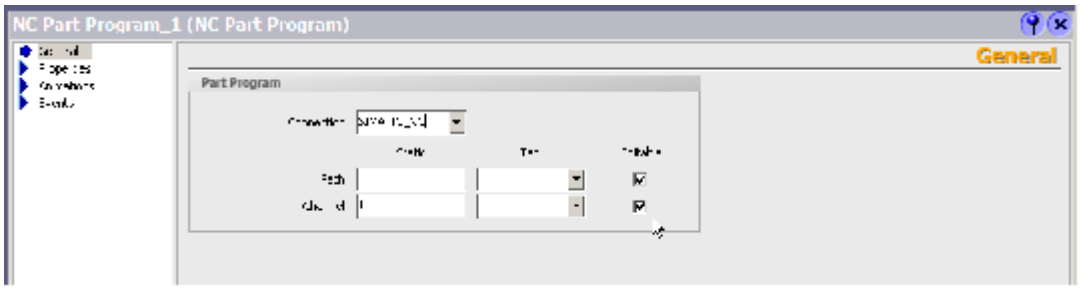


Figure 2-36 General

Properties

Under **Properties** , you can make global settings for the appearance of the SINUMERIK NC part program display object:

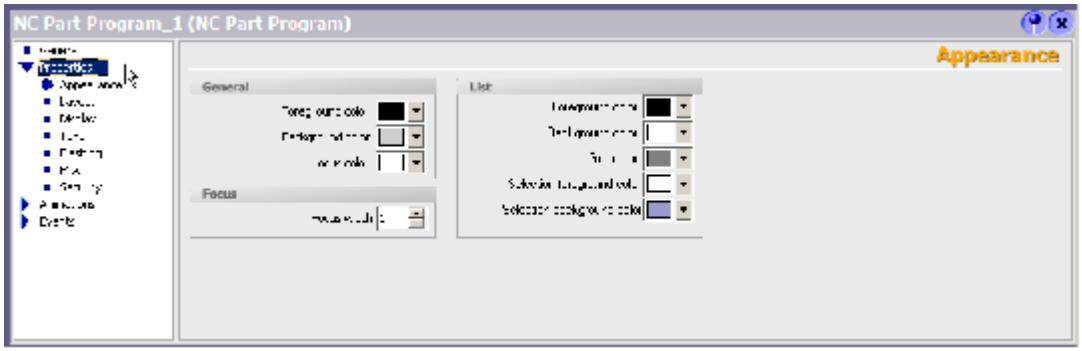


Figure 2-37 Properties

- **Display**
  Under **Display**, you can select the command buttons, etc., to be shown in the display object.

Note

In order to also be able to operate the **NC part program** externally or, depending on the functionality, to allocate an appropriate authorization, command buttons can be hidden in the display object during the configuration period and the required functions can be triggered via softkeys with varying authorizations.

The functions can then be selected from **SIMATIC NC**.

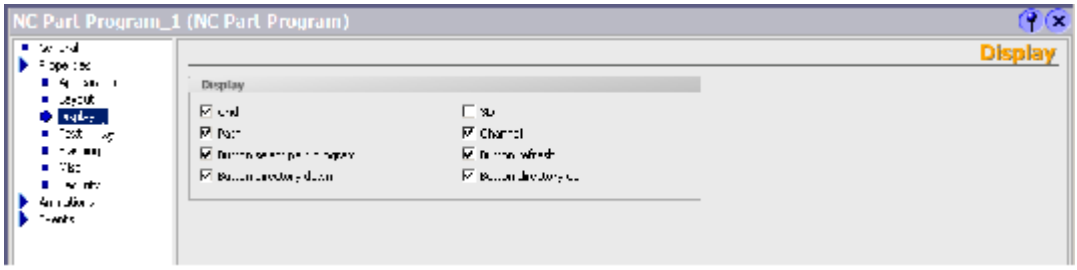


Figure 2-38 Display

- **Security**
  Under **Security**, you can configure the group authorized to select the part program/workpiece for editing on the NC.

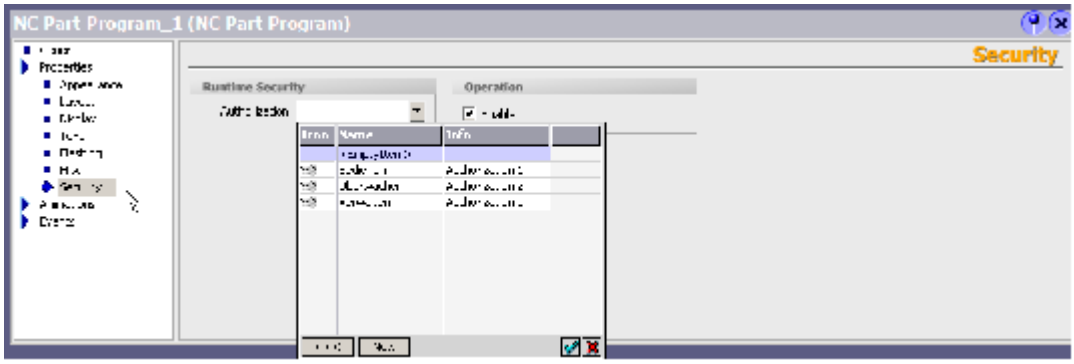


Figure 2-39 Security

Events

Under Events, you can react to events with functions.

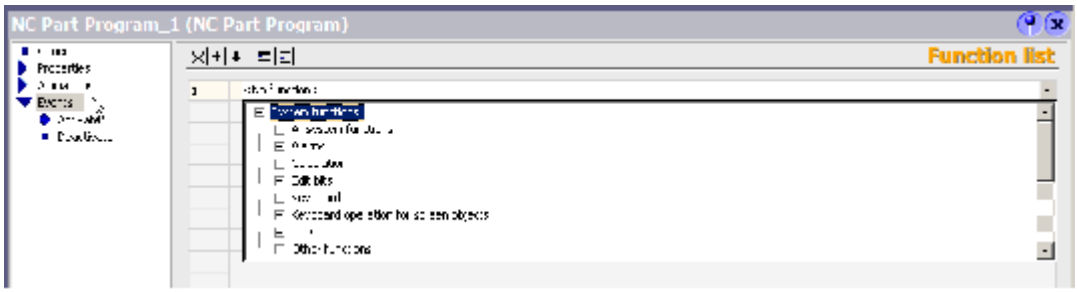


Figure 2-40 Events

Table 2-25 Events for the NC part program display object

Event	Significance
Enable	On selection, the display object is activated.
Deactivation	On deselection, the display object is deactivated.

Changing the display object using script

You can even change the properties of a display element during runtime using scripts. This provides a means, for example, of changing the colors or the part program path dynamically.

Script example

```
Dim view
Set view = HmiRuntime.Screens("Screen_1").ScreenItems("NC Part
Program_1")
view.Channel=1
view.BackColor=RGB(0, 255, 0)
view.GridlineColor=RGB(0, 0, 255)
view.Path="MPF_DIR"
```

2.5.3 Functions

Introduction

As described in the SINUMERIK general functions, events (e.g., "Key pressed") can be linked to defined functions in WinCC flexible. If an event should occur during operation, this function triggers a certain action on the HMI device or in the controller.

Principles

With HMI CE devices, you can use HMI CE device functions specifically for the SINUMERIK HMI WinCC flexible configuration software.

These are the following functions:

- Functions for the **NC part program** display object
- Functions for execution in the NCK
- Functions for the current block display
- Functions to start and stop the PLC

See also

- SINUMERIK functions for HMI CE devices (Page 129)
- PI services as configurable functions (Page 71)

2.5.3.1 Functions for the NC part program display object

Introduction

In order to also be able to operate the **NC part program** externally or, depending on the functionality, to allocate an appropriate authorization, command buttons can be hidden in the display object during the configuration period and the required functions can be triggered via softkeys with varying authorizations.

Functions

The following functions affect the NC part program display object:

Table 2-26 Functions for the NC part program display object

Function	Meaning
ScreenObjectCursorDown	
ScreenObjectCursorUp	

Function	Meaning
ScreenObjectPageUp	
ScreenObjectPageDown	
PartProgramListDirDown	Navigate and select in the list directory
PartProgramListDirUp	Navigate and select in the list directory
PartProgrammListRefresh	Update directories in the list box
PartProgrammListSelectNCPartProgramm	The selected part program/workpiece is selected on the NC for editing.

These functions can be selected under

- **Hotkey for display objects**
- **SIMATIC NC**

2.5.3.2 Functions for execution in the NCK

Introduction

An HMI CE device communicates with a SINUMERIK controller via the **SIMATIC NC** communication driver and enables the following calls to execute PI services in the NC controller component:

General procedure

Select the PI services in the screen editor under **Functions > SIMATIC NC**.

Functions

The following functions are available:

- **AcknowledgeNCCancelAlarms (NC: Acknowledge cancel alarms)**
  This function acknowledges all pending cancel alarms on the specified NC at once.

Table 2-27 Alarm acknowledgment parameters

Parameters	Significance
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.

Note

Assign the NC: General PI service function to a key a tag. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the function parameter.

- **ChangeNCPassword (NC: New password)**
  A password is transferred to the NC. The existing password for this NC level is overwritten.

Table 2-28 New password parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	The currently valid NC password is overwritten by the new password. The user must be logged on to the NC with a suitable level.
Password	New controller password (max. 8 characters) (constant or tag)	
Level	Password level of the NC (constant or tag): 0 = system 1 = manufacturer 2 = service 3 = user	

Note

Assign the NC: General PI service function to a key a tag. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the first function parameter. Enter the new password and its NC level as further parameters.

The password and level parameters can be configured as local tags or as constants.

- **ConfigureNCMachineData (NC: Configure machine data)**
  This function activates all machine data of the NEW_CONF classification.
  The CLASS parameter allows for a more exact classification. Currently, however, only value 1 is supported.

Table 2-29 Configure machine data parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	A corresponding PI service is transmitted to the NC from the currently valid parameters.
Class	Classification of the machine data being activated (tag or constant): 1 - 3	

Note

Assign the NC: General PI service function to a key a tag. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the first function parameter. Specify the class of the machine data to be activated in the NC as a further parameter. The parameters can be configured as tags or constants.

- **LogoffNC (NC: Logout)**
  The NC password is cleared with this function. Following this, the keylock switch is in control.

Table 2-30 Logout parameters

Parameters	Significance
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.

Note

Assign the NC: General PI service function to a key a tag. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the function parameter.

- **LogonNC (NC: Login)**
  A password is transferred to the NC. The passwords for the HMI device and the NC are independent of one another.

Table 2-31 Login parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	After activating the function, the specified password is used to attempt to log on to the NC.
Password	Password authorizing NC logon (tag or constant)	

Note

Assign the NC: General PI service function to a key a tag. Then select the symbolic name of the NC from the controller list as the first function parameter. Specify the password to be transferred as the second function parameter.

The password parameter can be configured as a local tag or a constant.

- **SetNCUserFrame (NC: Set user frame (activate zero offset)**
  Data for zero offsets are activated for a channel. This affects all NC tags of the block type FU. On entry, these tags are initially stored in a temporary memory on the NC. The new values entered are activated entirely via the NC: Set user frame function and can subsequently be read.

Table 2-32 Activate zero offset parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	After the HMI device user has entered the data for the zero offsets into the HMI device and pressed the configured key, the zero offset data is activated for the specified channel.
Channel	Channel to be activated (constant or tag): 1 - max 10	

Note

Assign the NC: Cold restart function to Set user frame function to a key. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the first function parameter. Enter the channel for the zero offset data as the second parameter.

The channel parameter can be configured as a tag or a constant.

- **SelectNCPartProgram (NC: Select part program)**
  As well as using the part program overview display element for operation, additional functions for the selection of the NC part programs are implemented, which can be selected for all part programs from the SINUMERIK controller using domain services. For example, special screens can be created for permanently-defined part programs. These screens can then support a simplified selection process, using command buttons.

Table 2-33 Select part program parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	The specified part program is selected on the NC for editing.
Channel	Channel, for which the program is selected: 1 – max. 10 (tag or constant)	
Path name	Absolute path specification of the program directory (tag or constant); without prefix _N_ compatible with OP7/17	

Parameters	Significance	Procedure on HMI device
Program name	Name of the program or the workpiece to be processed (tag or constant) without prefix _N_ compatible with OP7/17	

Note

Together, the path name and program name parameters must indicate the part program path.

Assign the **NC: Select part program** function to a screen key and specify the corresponding NC, the channel, path names, and program names as parameters.

Enter only the absolute program or path name.

Path name examples:

- MPF_DIR for main directory

- SPF_DIR for subdirectory

- WKS_DIR for workpiece program directory

The path name and program name parameters can be configured as tags or constants.

- **StartNCPIService (NC: General PI service)**
  This function can be used to send any PI service to the NC. This must be described by a parameter string in accordance with the OPI documentation. This parameter string must include the full command and not merely the parameters of the PI Service, for example,

  NCK reset
  Parameter string: PI_START(/NC,K00,0,_N_IBN_SS)

Table 2-34 General PI service parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	A corresponding PI service is transmitted to the NC from the currently valid PI parameter string.
PI parameter string	Specification of the PI service to be executed (constant or tag)	
Result	When a function is activated, a return value is reported in this parameter asynchronously after termination.	

Note

Assign the NC: General PI service function to a key or a tag. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the first function parameter. Specify the PI parameter string as a further parameter.

The PI parameter string parameter can be configured as a tag or a string type constant.

- **ResetNC (NC: Cold restart (NCK reset)**
  This function triggers a cold restart on the NC.

Table 2-35 NCK reset parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	A cold restart is executed on the NC.

Note

Assign the NC: Cold restart function to a key. Then select the symbolic name of the NC, for which the function is to be executed, from the controller list as the function parameter.

2.5.3.3 Function for the current block display

Introduction

An HMI CE device communicates with a SINUMERIK controller via tags and enables the following call for the current block display:

General procedure

Select the function using **Open Screen** in the **Properties Editor** under **Events->Generated->Functions** in the **SIMATIC NC directory.**

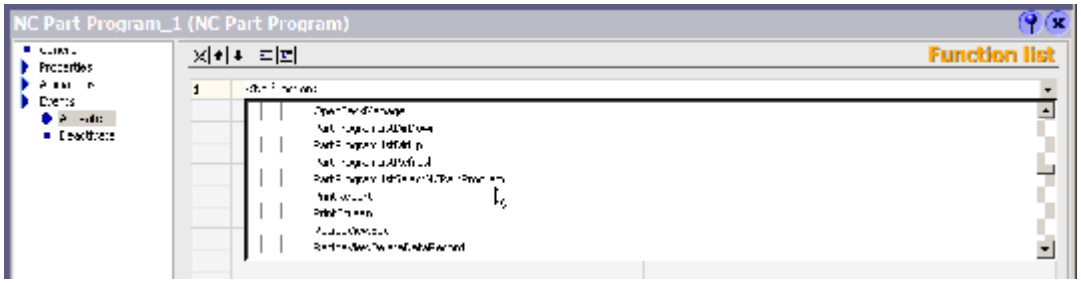


Figure 2-41 Current block display

- **PrepareActualNCBlockForDisplay**
  This function takes the current, previous and subsequent traversing block of an NC part program, which is currently being processed in a channel, from the NC and stores them in local tags.

Table 2-36 Display part program block parameters

Parameters	Significance	Procedure on HMI device
PLC name	Symbolic name of the NC, which you entered under the "Properties" of the controller.	Destination: You wish to monitor execution of the traversing blocks of an NC traversing program on the control device. The Display NC part program block function is loaded when opening screens. A non-configured OPI tag block is activated with the configured cycle. On each cyclic reading operation of the OPI tag block, the tag block is divided into three parts and stored in the local tags of the previous, current and subsequent blocks. The local tags can be displayed on the screen as outputs.
Channel	Channel, from which the program is to be read (tag or constant) 1 - max. 10	
Cycle	Update time as a multiple of 100 ms (constant only): 1 – 600 (equal to 100 milliseconds – 1 minute)	
Previous block	Local tag for the traversing block, which precedes the current traversing block (STRING data type)	

Parameters	Significance	Procedure on HMI device
Current block	Local tag for the current traversing block (STRING data type)	
Subsequent block	Local tag for the block following the current traversing block (STRING data type)	

Note

Specify three local tags of type String of length 66. Within a screen, create three output fields each with one of these tags. As an open screen trigger, now configure the NC: Display part program block function for this screen. Next, specify the corresponding NC, the channel from which the current block is to be read, the cycle, and the three local STRING tags as parameters for this function.

The channel parameter can be configured as a tag or a constant.

2.5.3.4 Function to start and stop the PLC

Introduction

An HMI CE HMI device communicates with a SINUMERIK controller via the **SIMATIC S7 300/400** communication driver and enables the following calls to execute a PI service in the S7-300 controller component:

General procedure

Select the PI service from the **Settings functions.**

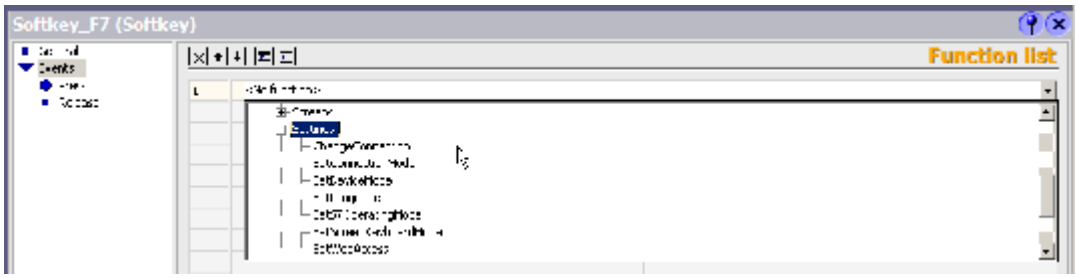


Figure 2-42 S7 change operating status

- **SetS7OperatingMode (Change S7 operating state)**
  This function is used to select the operating state of a S7 CPU, from: RUN or STOP.

Table 2-37 Change S7 operating state parameters

Parameters	Significance
PLC name	Symbolic name of the S7-CPU, which you entered under the "Properties" of the controller.
Operating state	Operating state of the S7 CPU (tag or constant): 0 = RUN 1 = STOP

Note

Assign the Change S7 operating state function to a screen key and specify the corresponding controller and the operating state as parameters.

The operating state parameter can be configured as a tag or a constant.

2.5.4 Alarms and Messages

2.5.4.1 SINUMERIK NC messages

Introduction

The following SINUMERIK NC messages can be configured for HMI CE devices:

- NCK system messages with fixed texts (alarms)
- Comments on part programs allocated by the user (messages)

General procedure

If you have configured an HMI CE device with the **SIMATIC NC** communication driver as the connection for a SINUMERIK controller, you can also configure the SINUMERIK NC messages as a messaging process using **Messages > Settings > Message Settings** (see screen below).

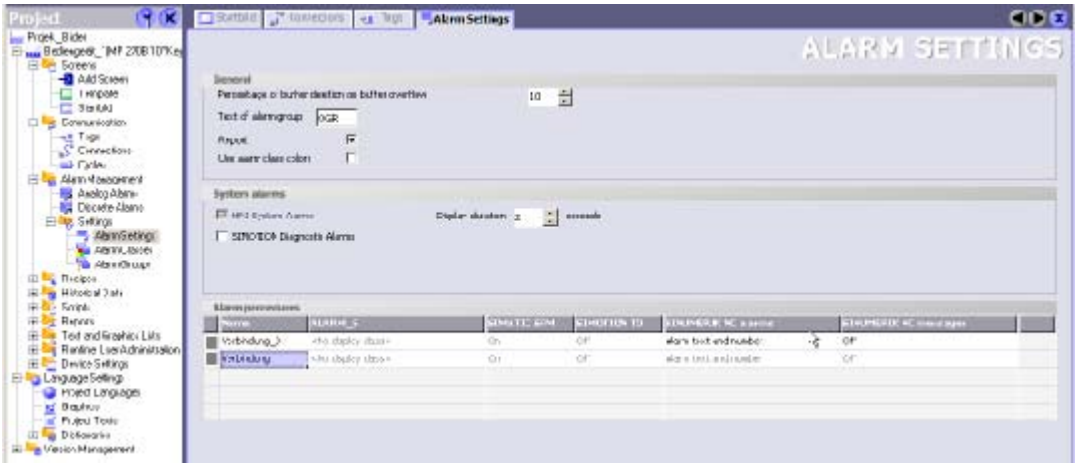


Figure 2-43 Alarm settings

For the display of NC alarms with fixed texts, you can choose between:

- Message text and message number
- Message number only

Note

If necessary, the display of message texts can be deactivated to increase the available memory capacity.

2.5.4.2 Alarm and message directories

Introduction

NCK alarm and message texts cannot be configured in WinCC flexible ES.

NCK alarm texts are stored in multiple languages in a permanently-defined directory on installation of WinCC flexible ES.

General procedure

As a rule, when generating the alarm folder for a configuration (e.g., ProjectNC.hmi), a search is carried out in the following order:

1. Device-specific and project-specific in the configuration directory, e.g., **ProjectNC.Device_1.ES2RT**
2. Project-specific in the configuration directory, e.g., **ProjectNC.ES2RT**
3. Configuration-specific in the installation directory, e.g., **..\WinCC flexible 2007\configs\ES2RT**

Alarms are defined in the following order in the alarm folder and subdirectory **SimaticNC**:

- User

- Oem
- Siemens

Note

If you wish to configure alarms project-/device-specifically, these must be stored in a new alarm folder. This alarm folder must be located in the same directory as the project file.

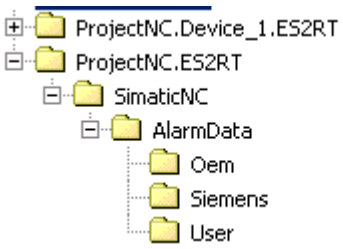
If you change the alarms, a complete generation process must be carried out for the project in question. First delete the temporary files with menu item: Delete Options/Temporary Files.

An alarm text must be always generated and attached in a directory in all languages. A new language, e.g. "Polish", must always be installed in the correct directory and the appropriate adjustment made in the ini file where necessary.

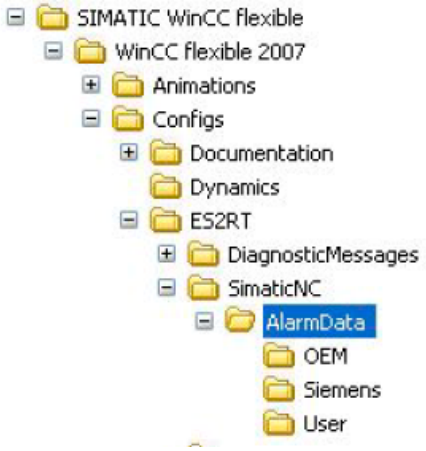
Subdirectories

During the configuration period and for SINUMERIK controller alarms, you have the following options for saving in subdirectories:

- **Project/device-specific**
  The screen below shows an alarm directory structure. In this case, we are dealing with a project-specific directory, e.g., **ProjectNC.ES2RT\SimaticNC\AlarmData**:



- **Configuration-specific** (see screen below)
  This directory **..\WinCC flexible 2007\Configs\es2rt\SimaticNC\AlarmData** (see screen below) is created by default when the HMI SINUMERIK WinCC flexible configuration software is installed and contains the following subdirectories:
  - Oem
  - Siemens
  - User
    The screen below shows the directory after installation:



Directory names

The directories must be named as follows, for example, if you have created a **ProjectNC.hmi** and a **Device_1** device:

- **Device-specific**
  , e.g., **ProjectNC.Device_1.ES2RT**
- **Project-specific**
  , e.g., **ProjectNC.ES2RT**
- **Configuration-specific**
  , e.g., **ES2RT**

Note

Make a copy of directory ..\SimaticNC\AlarmsData.. from the installed (configuration-specific) directory and save this in the appropriate folder (device-/project-specific). Only change the alarms in this location.

These directories must be transferred with every project.

Alarm initialization file "WinCC_NCK.ini"

A WinCC_NCK.ini file must be installed in the alarm data directory. This file adapts SINUMERIK language-dependent alarm files to WinCC flexible syntax.

By default, this file is structured as follows:

```
[Files]
TextFilePrefixes=ALMC_,ALSC_,ALC_,ALZ_,ALN_

[LocaleMap]
UK=9
GR=7
TW=1028
CZ=5
DK=6
```

```
SP=10
FI=11
FR=12
HU=14
IT=16
JP=17
KO=18
NL=19
NO=20
PL=21
PO=22
RU=25
SK=27
SW=29
TR=31
CH=2052
```

Files

The [Files] section contains the **TextFilePrefixes** entry. This entry contains the prefixes of the alarm files currently in use. The files are ordered according to priority.

Some files contain the same message numbers, so a message number that is defined in numerous files is taken from the first file, in which it appears.

Locale Map

The [LocaleMap] section contains language code entries (e.g., GR for German), which are used by SINUMERIK.

WinCC codes (e.g., = 7 for German) are then allocated to these language codes.

Generating a project with SINUMERIK NC messages

If you generate a project after configuring it with SINUMERIK NC messages, WinCC flexible will check the alarm files for correct file names, syntax, message numbers, etc., and output any errors (e.g., "File "file name" not found").

2.6 Alarm handling

2.6.1 Alarm handling with SINUMERIK controllers

Introduction

With the SINUMERIK HMI configuration software, WinCC flexible displays SINUMERIK NC and PLC alarms.

These alarms are assigned internally to the **Operation** message class.

Note

Should you wish to change the background color, you can change the color for the **Operation** message class.

This is configured by selecting the corresponding setting (see screen below **Messages > Settings > Message Classes**).

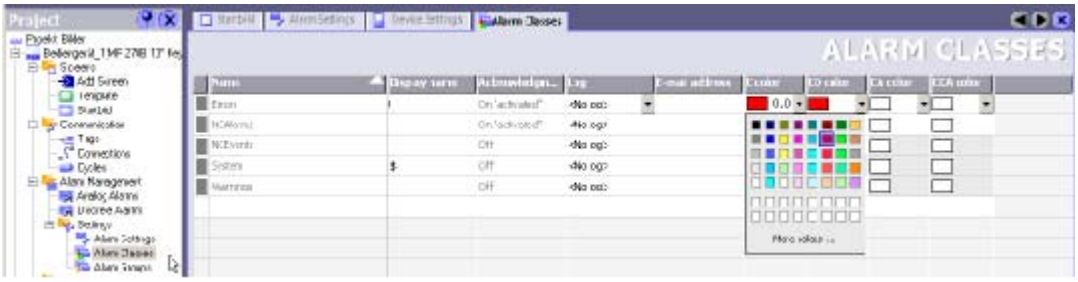


Figure 2-44 Message class settings

Note

The **Use message class** field must be checked under **Messages > Settings > Message Settings**.

2.6.2 ALARM_S

Introduction

Alarms of the alarm_S message class can be displayed in projects integrated into STEP 7.

Settings:

- General message settings
- Message view display object

General message settings

Activate the alarm_S message class under **Messages -> Settings -> Message Settings** by checking the **All display classes** field (see screen below).

Settings

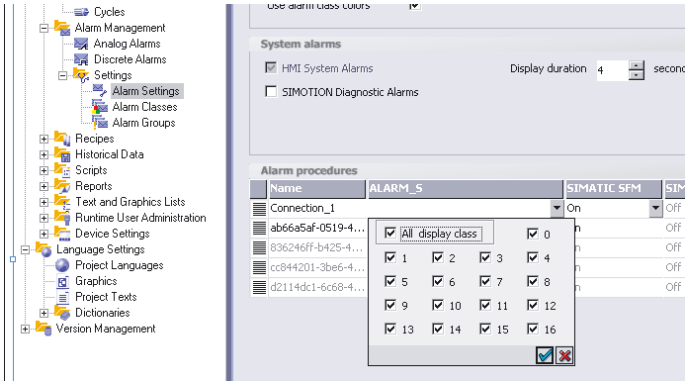


Figure 2-45 Alarm_S settings

Message view display object

To display ALARM_S messages in the configured **Message view** display object, check the **S7 Message** field in the display object under **General -> Message Classes** (see screen below).

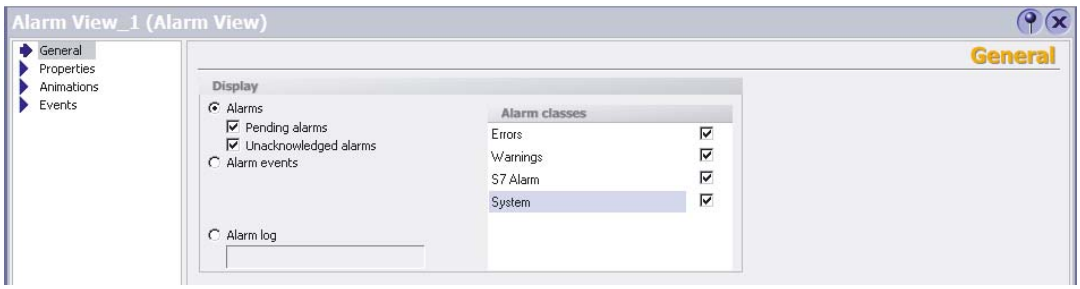


Figure 2-46 Activate_alarm_S

Special points to be noted for SINUMERIK

COM files

During the WinCC flexible ES generation process, the configured alarm_S messages are also stored in *.com files. SINUMERIK diagnostics need these files in order to display alarm_S messages.

The files are stored in the project directory of the particular HMI project (configuration calculator).

Note

You must save the current *.com files to the MB.DIR directory of the SINUMERIK data management on the target hardware.

You must also specify these files in file mbdde.ini (possibly by creating a new file of this name in the directory) with the following entry, e.g., for Project_1:

```
[Textfiles]
UserPLC_PMC = F:\DH\MB.DIR\PROJECT_1.OP012DEMO_
```

Message number allocation for the entire project

With STEP 7, it is possible to allocate unique message numbers for the entire project or for the entire controller.

Note

Message numbers must be allocated for the entire project for the SINUMERIK alarm server.

Ini entries

The following ini entries are required in order for SINUMERIK MESSAGE LOGGING to process alarm_S messages correctly:

mmc.ini ->

```
[Add_on_Features]
PDIAG = 1
```

mbdde.ini ->

```
[Alarms]
ConnectToPTPAlarmServer = 1
```

This entry controls the SINUMERIK diagnostics attempt to obtain additional alarm information from WinCC flexible RT.

This procedure is required, for example, if you also wish to display erroneous sequencer names from S7GRAPH programs on the SINUMERIK diagnostics screens.

2.6.3 DB2 alarms

Introduction

DB2 alarms are displayed on HMI-PC and HMI-CE operator devices with different communication drivers.

HMI PC devices

For the HMI-PC operator devices, DB2 alarms are displayed in the alarm overview if, for the appropriate PLC, a communications driver **SINUMERIK PLC** was configured.

The message texts that are displayed come from the appropriate SINUMERIK alarm text files.

HMI CE devices

For HMI-CE operator devices, DB2 alarms are not automatically displayed.

In order to display DB2 alarms, a communications driver **S7300/400** must be configured for the PLC. For this connection, DB2 alarms must be subsequently configured as bit messages.

Messages 1..32767 are possible as message numbers for bit messages; this means that the DB2 alarm numbers must be mapped on these.

2.6.4 Fault location, an additional message parameter

Introduction

The alarm indication features an additional column containing information about the fault location. The name of the faulty controller appears here. This information is of particular importance as regards m:n configurations. The controller name will only be displayed if the symbolic controller name in the m:n configuration file, netnames.ini, matches the addressing name.

Example (from netnames.ini)

```
[own]
owner=MMC
```

```
[conn MMC]
conn_1=SIN840D
conn_2=CPU315

[param SIN840D]
...
name=SIN840D

[param CPU315]
...
name=CPU315
```

Both these controllers must be addressed using net:SIN840D and net:CPU315 in the Parameters dialog for WinCC flexible. Entries are case-sensitive.

2.7 Integration of WinCC flexible in STEP 7

2.7.1 Basic principles of integration in STEP 7

Introduction

If you are using a SINUMERIK control and have installed the STEP 7 configuration software on your system, you can integrate WinCC flexible in STEP 7.

Advantages of integrating in STEP 7

During integrated configuration, you access STEP 7 configuration data that you created when you configured the PLC with STEP 7. This gives you the following advantages:

- You can use the SIMATIC Manager as a central point for creating, processing and managing SINUMERIK controllers and WinCC flexible projects.
- The communication parameters of the PLC are preassigned when the WinCC flexible project is created. When a change takes place in STEP 7, the communication parameters are updated in WinCC flexible.

- Connection parameters created by the system during STEP 7 integration: Network parameters and partner parameters are preassigned
- When configuring tags and area pointers, you can access the STEP 7 symbols directly in WinCC flexible. In WinCC flexible, simply select the STEP 7 symbol to which you would like to link a tag. Symbol changes made in STEP 7 are updated in WinCC flexible.
- You assign a symbolic name once in STEP 7 and use it in STEP 7 and WinCC flexible.
- ALARM_S and ALARM_D alarms configured in STEP 7 are supported in WinCC flexible and can be output on the HMI device.
- You can create a WinCC flexible project without integration in STEP 7 and integrate it in STEP 7 at a later time.
- You can remove an integrated project from STEP 7 and use it as a standalone project.
- In a STEP 7 multiproject, communication connections can be configured across projects.

Installation requirements

A specific installation sequence must be followed to integrate WinCC flexible in STEP 7. You must first install the STEP 7 software, and then WinCC flexible. When installing WinCC flexible, it detects an existing STEP 7 installation and automatically installs the support for integration in STEP 7.

For user-guided installation, the "Integration in STEP 7" option must be activated.

If WinCC flexible is already installed and STEP 7 is installed subsequently, WinCC flexible must be uninstalled and reinstalled once the STEP 7 installation is complete.

Note

For more information (e.g., about symbolic addressing), refer to the STEP 7 documentation.

Reference

The TOOL-Box software is supplied with the SINUMERIK HMI WinCC flexible Configuration Software to allow you to select SINUMERIK controls (for example, 810D/840D with PLC315) in the HW config of STEP7.

See also

- TOOLBOX (Page 127)

2.7.2 Working with the SIMATIC Manager

Introduction

When you are working with WinCC flexible integrated in STEP 7, you can use the SIMATIC Manager for your WinCC flexible projects. In STEP 7 projects, the SIMATIC Manager is the central point for managing your projects, including your WinCC flexible projects. The SIMATIC Manager enables you to access the configuration of your automation systems and the configuration of the operator control and monitoring layer.

Requirement

WinCC flexible is integrated in SIMATIC STEP 7.

Working with the SIMATIC Manager

In integrated projects, the SIMATIC Manager provides the following options:

- Create an HMI or PC station with WinCC flexible Runtime
- Insert WinCC flexible objects
- Create WinCC flexible folders

- Open WinCC flexible projects
- Compile and transfer WinCC flexible projects
- Export and import texts for translation
- Specify language settings
- Copy or move WinCC flexible projects
- Archive and retrieve WinCC flexible projects within the framework of STEP 7 projects

2.7.3 Working with HW Config

Introduction

The HW Config editor is provided in STEP 7 for configuring and assigning parameters to the hardware. Use drag-and-drop operations to assemble the required hardware. A catalog is provided for selecting the hardware components. During configuration, a configuration table with the address parameters is automatically created. During subsequent editing in STEP 7 or WinCC flexible, the system accesses this configuration table and accepts the prepared parameters.

Using HW Config

You use HW Config to create the hardware configuration for new stations or to add required modules to existing stations. HW Config provides a catalog with the available modules and preconfigured components and stations. HW Config checks the usability of the objects you wish to insert. Thus, unusable or illegal objects cannot be inserted. Edit the properties of an inserted object directly in HW Config. Open the context menu of the object and select "Object Properties". Edit the object properties directly in the dialog that appears.

For example, you can create a PC station in the SIMATIC Manager. Open the station for configuration in HW Config. Insert a WinCC flexible runtime application. Select a communication interface and insert it. Edit the settings for the communication interface in HW Config. The WinCC flexible runtime application will not open via HW Config. To open it, use SIMATIC Manager.

For more information, consult the HW Config documentation.

2.7.4 Configuring connections

Introduction

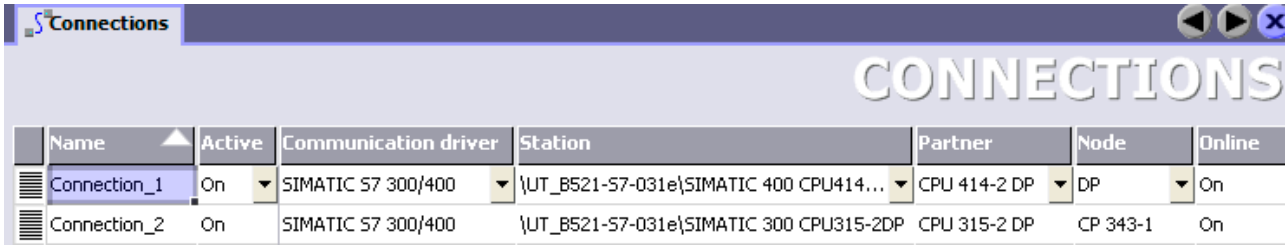
Data exchange between WinCC flexible and the automation layer requires connections for communication to take place. In integrated projects, you can create connections with the following applications:

- WinCC flexible
- NetPro

This configuration can be made with either WinCC flexible or NetPro. This does not affect the Sinumerik_NC and Simatic_NC connections. These connections must be configured in WinCC flexible and not in NetPro.

Using WinCC flexible

You can create new connections or edit existing ones. In integrated projects, the "Station," "Partner," and "Nodes" columns are also provided in the editor for connection configuration.



When creating a connection, select the station, partner, and connection node from selection lists. The required connection parameters are automatically accepted in STEP 7. The project has to be saved after configuration has been completed. Connections, which you configure in WinCC flexible, are not transferred to NetPro and can only be edited with WinCC flexible.

Using NetPro

NetPro is recommended for use with larger projects. In NetPro, you configure the connections on a graphically supported interface. When you start up NetPro, the devices and subnets in the STEP 7 project will be displayed. NetPro has a catalog of network objects that you can use to insert additional devices or subnets. In integrated projects, this catalog also includes the SIMATIC HMI station object. You insert objects from the catalog in the work area of NetPro using a drag-and-drop operation. Drag and drop individual stations to connect them to the subnets. Use Properties dialog boxes to configure the connection parameters of the nodes and subnets. You then save the configuration in NetPro to update the data management in the WinCC flexible project. Connections, which you configure with NetPro, can only be read in WinCC flexible. In WinCC flexible you can only rename the connection, enter a comment for the connection and set the connection "online". Editing of the connection itself is carried out exclusively with NetPro.

Note

Subnet properties, such as the data transmission rate, are set automatically in STEP 7 for all nodes in a subnet. If you create or modify the subnet properties yourself, you must ensure that these settings are applied for each node in the subnet. You can find more information on this topic in the NetPro documentation.

Note

If a new HMI station is set up in STEP 7, the MPI/DP nodes are set to MPI and Address 1 by the system. If the HMI station is not networked and the HMI station should be networked via a different substation type, the connection parameters must be changed in NetPro or in the HW configuration.

2.7.5 Working with objects

Introduction

Perform the following steps to create an integrated WinCC flexible project:

- Create a new HMI station in the SIMATIC Manager
- Integrate a WinCC flexible project in STEP 7.

Creating an HMI station

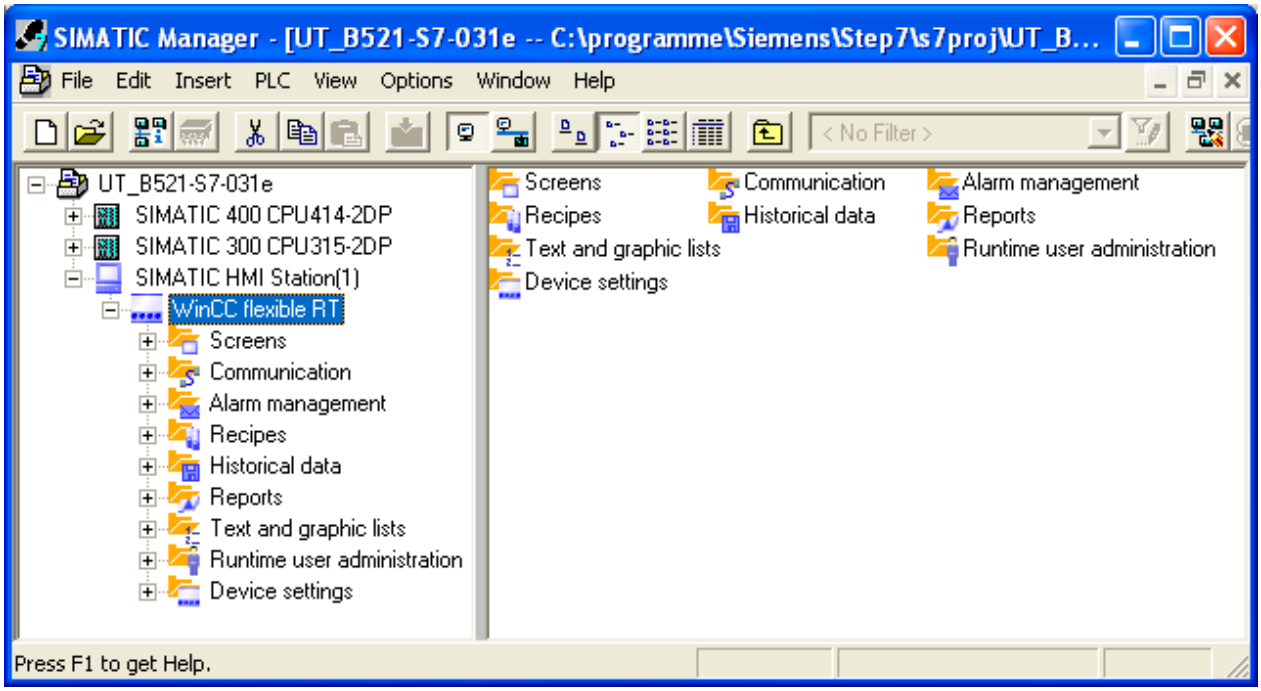
Creating an HMI station in the SIMATIC Manager basically creates a new WinCC flexible project.

Inserting multiple HMI devices in a WinCC flexible project

If you require multiple HMI devices in a WinCC flexible project, you must insert the HMI devices in the project in WinCC flexible.

Inserting WinCC flexible objects

Once a WinCC flexible project has been integrated in STEP 7, the project is displayed in the project window of the SIMATIC Manager. A WinCC flexible project is displayed in the project window of the SIMATIC Manager in the same way as in the project window of WinCC flexible. If you select a WinCC flexible element in the project window, the objects of the WinCC flexible project are displayed in the work area.



From here, you can open existing projects or create new ones. If you create or open a WinCC flexible object in the SIMATIC Manager, WinCC flexible is automatically started for editing the object.

Select, for example, the "Screens" element and create a WinCC flexible screen directly in the SIMATIC Manager. The new screen will be created and opened immediately for editing in WinCC flexible. Or create a log of changes in which all the changes of the project are documented in a continuous table. Changes of STEP 7 are not explicitly identified in the log of changes.

Note

Working with integrated projects depends on the rights of the Windows user who is logged on. You need at least write access for this. If the Windows user opens an integrated project with read-only access, the user cannot make any changes. Data cannot be synchronized between STEP 7 and WinCC flexible with read access.

Representing WinCC flexible objects

Global project elements that cannot be edited in the SIMATIC Manager are not displayed; examples of such elements include version management and language settings.

Data that you edit in WinCC flexible with a table editor are displayed as symbols in the SIMATIC Manager. Opening such symbols via the SIMATIC Manager causes WinCC flexible to open for editing the data. For example, if you select the "Tag" element, a symbol for all WinCC flexible tags will be displayed in the work area of the SIMATIC Manager. The individual WinCC flexible tags are not displayed in the SIMATIC Manager. If you create a new tag in the SIMATIC Manager, it will be created in WinCC flexible and opened for editing in WinCC flexible.

For more information about STEP 7, consult the SIMATIC Manager documentation.

Changing the names of STEP 7 data blocks

If you change the name of a STEP 7 data block, it can happen that the old name is displayed in the object list of WinCC flexible. To make sure that the changed name of the STEP 7 data block is displayed in the object list, open the list again in the WinCC flexible project integrated into STEP 7.

Renaming a WinCC flexible project

The renaming of an integrated WinCC flexible project in SIMATIC Manager is only of temporary nature. After having opened and renamed the project in WinCC flexible, you changes will be lost if do not save the project before you close it again.

2.7.6 STEP7 integration with SINUMERIK hardware

Prerequisite

The SINUMERIK HMI WinCC flexible configuration software is installed. The STEP 7 hardware catalog now contains the SINUMERIK hardware.

Procedure for creating an integrated WinCC flexible project with SINUMERIK hardware.

1. Select the appropriate SINUMERIK hardware (e.g., 840D) in HW Config.
2. Insert the HMI station (WinCC flexible project is created).
3. Insert a communications processor for the HMI station (specify MPI card).
4. Connect the HMI station to the control (e.g., via NetPro).
5. Save the project.
   By following these steps, a communication driver **SIMATIC S7 300/400** will automatically be set up as the connection to the configured SINUMERIK hardware and two nodes (programming device interface -> PLC315-2DP M/S, 2AF03 OPI interface -> S7 FM-NCU) will be made available (see screen below).

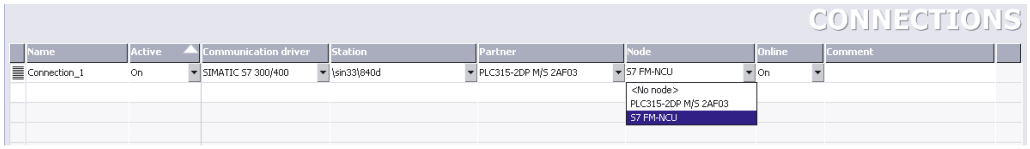


Figure 2-47 SIMATIC S7 300/400

6. Switch from the **SIMATIC S7 300/400** communication driver to the **SINUMERIK PLC** communication driver to access the OPI interface (in WinCC flexible **nodeS7 FM-NCU**) (see screen below).

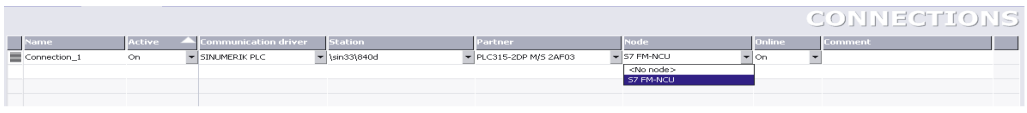


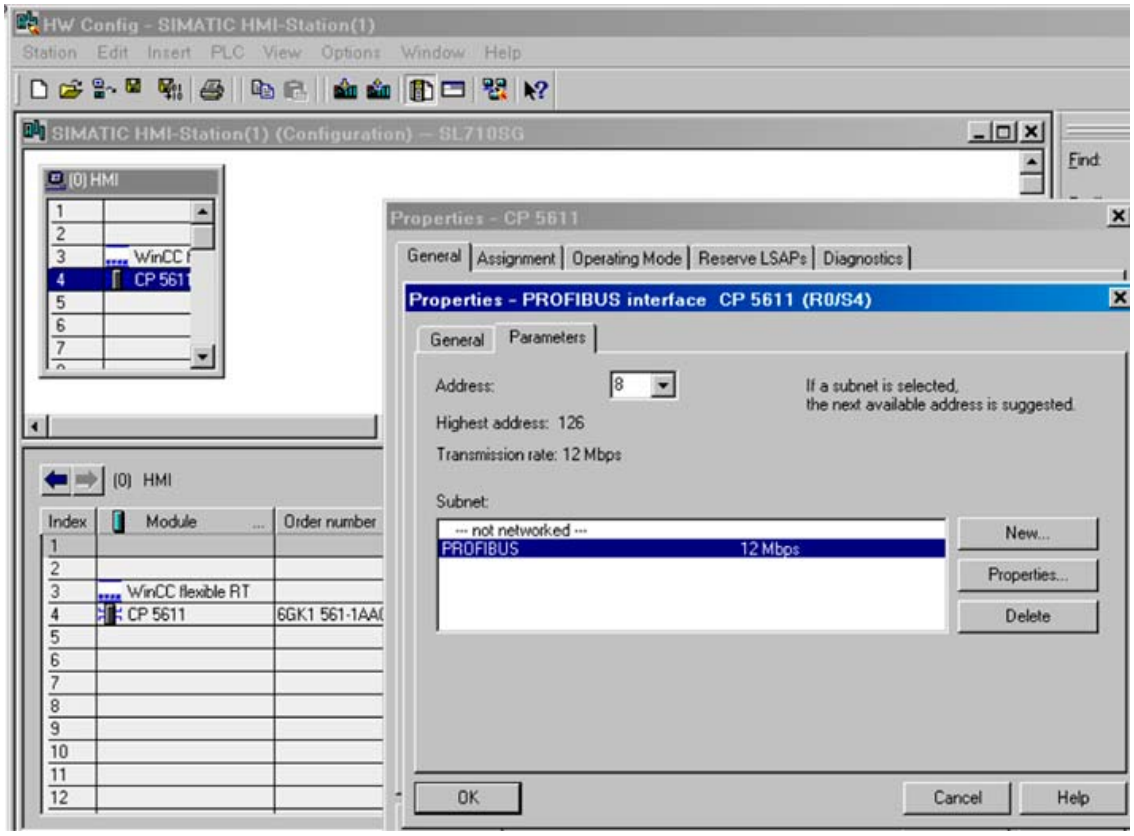
Figure 2-48 SINUMERIK PLC

If this has been correctly configured, the STEP 7 symbols can be accessed in the tag browser.

More detailed description of items 3 and 4 and tips on symbolic addressing

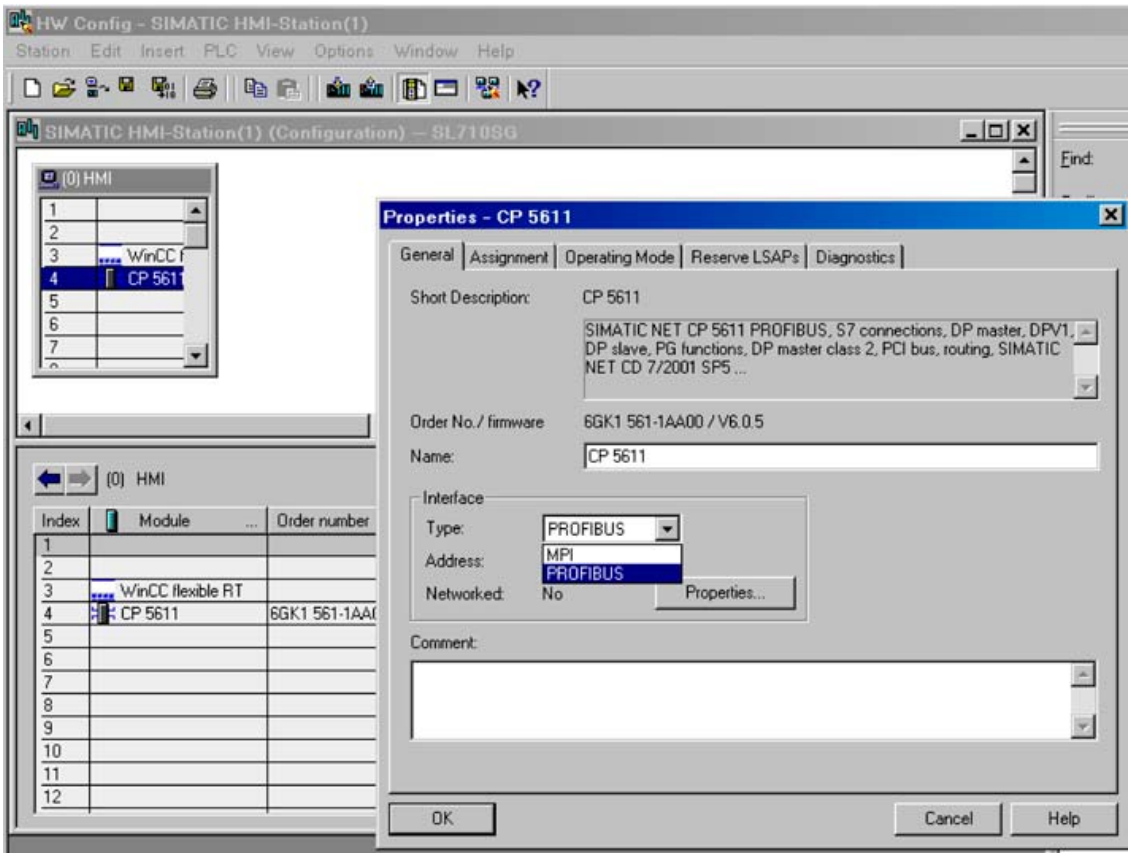
- The PLC in question (SINUMERIK PLC) is **not** created manually in the WinCC flexible project
- Open the integrated project in **STEP 7**

  Important:
  The WinCC flexible ES should be closed if you have not already done so (before opening NETPRO)
- Start the **NetPro** tool via "Options" → "Configure Network"
- Double-click on the HMI component ("SIMATIC HMI STATION") to open the **HW Config** for the HMI component
- The CP module (CP5611 for PCU 50) is selected from the component library and inserted in the HMI component; the CP module is connected to the existing PROFIBUS subnetwork for the **first time** in the properties dialog box that now opens

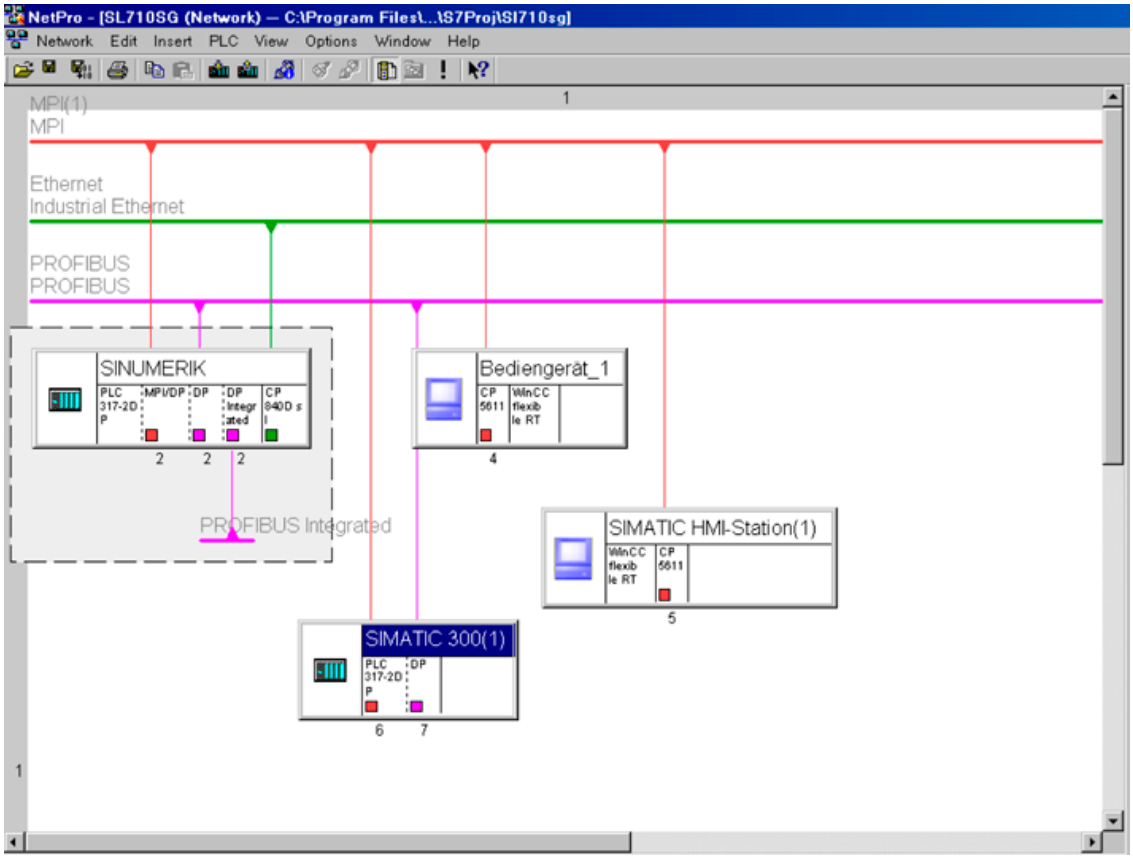


- **Please note: The CP is connected to the Profibus network and not to the MPI network.**

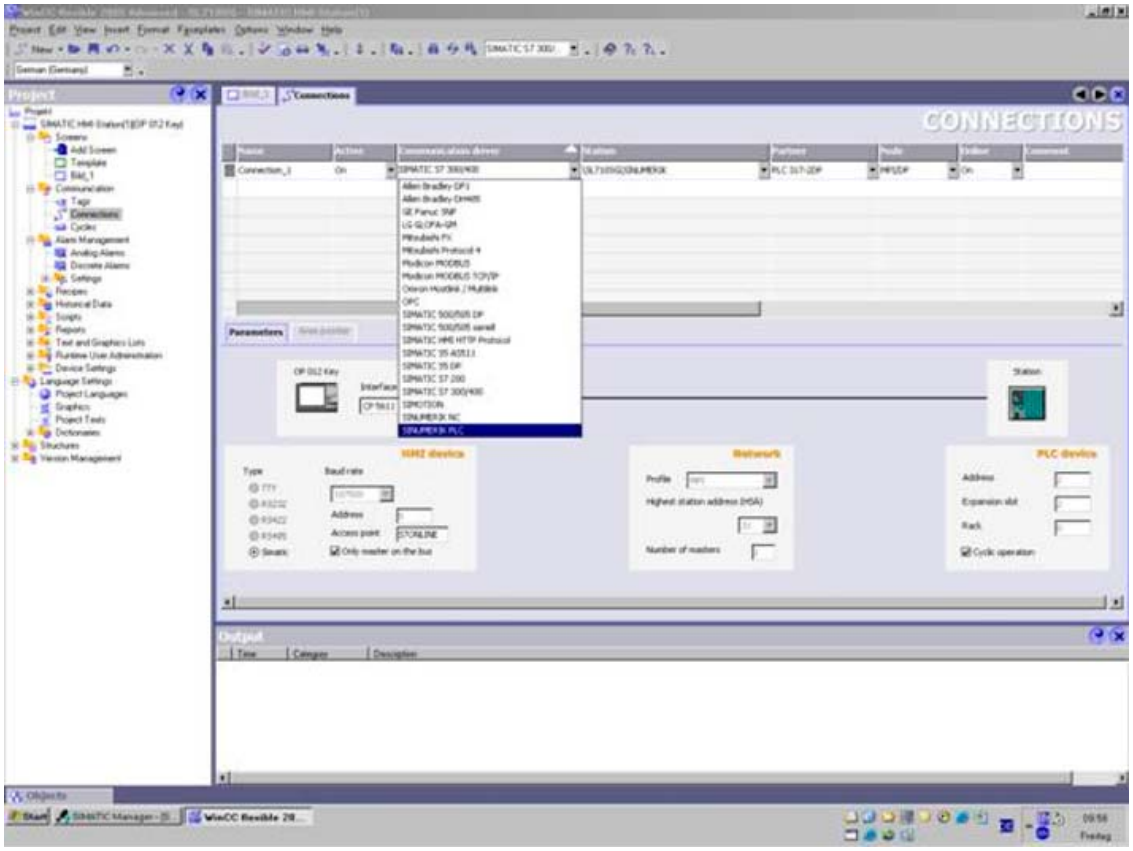
  Close the PROFIBUS window and reopen it. Now you can select either PROFIBUS or MPI. This is where the interface is changed to "MPI". You can select the MPI network (187.5 MBd) with the "Properties" button and set the address of the PCU (1).



- Translate and close HW Config
- The HMI component is now connected to the MPI network in NetPro; using the mouse you can now also connect the PLC station, address 13, to the MPI network.



- Translate and close NetPro
- In Step 7, **WinCC flexible** is displayed via the project tree → "WinCC flexible runtime" → "Communication" → "Connections".
- In WinCC flexible you can now see the (automatically created) PLC connection. **Please note: The automatically created connection may be of type "SIMATIC S7 300/400".** The name ("Connection_1") can be adapted, the connection is activated, and the communications driver ("SIMATIC S7 300/400") is converted to "SINUMERIK PLC"



- Now you can set the station and possible the partner and node.
- Symbolic addressing of PLC tags is now possible.

If transfer of STEP 7 symbols is not yet possible, the following steps may give the desired results:

- Export the integrated HMI project from the STEP 7 project, e.g. to a TMP directory. Now delete the WinCC flexible HMI in the STEP 7 project in NETPRO (Tools → Configure Network)!

  Save the STEP 7 project with Save As and **active reorganization**, to tidy up the project. Close STEP 7.

  Then open WinCC flexible with the project you have just exported and with Project→Integrate into STEP 7, integrate the WinCC flexible project into the STEP 7 project you have just saved using REORG Close WinCC flexible.

  Now you can again open the STEP 7 project, which once more contains the WinCC flexible project.

  Now repeat items 3 and 4.
- The project path may be too long → shorten the path!

2.8 Communication with SINUMERIK 840D sl (solution line)

Introduction

A communication link to the SINUMERIK 840D sl (NCU7x0) can be set up with the SINUMERIK HMI WinCC flexible 2007 configuration software .

General procedure

If you want to integrate a WinCC flexible project into an HMI you should first parameterize the HMI (Start-Up --> HMI --> Operator Panel for powerline or Start-Up --> HMI --> NCU Connection for solution line) to connect it to your control. The integrated WinCC flexible project then uses the existing communication routes.

Table 2-38 Access to SINUMERIK powerline/sl

Communication driver	SINUMERIK powerline	SINUMERIK 840D sl Ethernet	SINUMERIK 840D sl PROFIBUS
SINUMERIK NC (PC)	*	*	*
SINUMERIK PLC (PC)	*	*	*
SIMATIC NC (CE)	*	*	*

Special points to be noted

3

Overview

The section deals with special points to be noted during configuration. These include:

- Multilingual configuration
- Converting projects
- Error messages
- PLC string

3.1 Memory capacity of CE panel

What do you need to know about the memory capacity of CE panels?

You need to remember the following if you want to migrate an existing ProTool/Pro project to WinCC flexible:

1. No alarm texts were stored in the configuration for the SINUMERIK control in ProTool/Pro.
2. In WinCC flexible, all SINUMERIK alarm texts are stored as standard in the configuration.
3. You can deactivate the storage of alarm texts in the ES system.
4. The storage of alarm texts uses an extra 340KB of storage capacity per language.
5. When a ProTool/Pro project is migrated to WinCC flexible, the memory requirement of the RT file increases initially as a result.

3.2 Multilingual WinCC flexible RT project

For each project, you can store texts in as many configuration languages as you wish.

The configuration language is the language in which the configured texts will appear on the HMI device under WinCC flexible RT.

The WinCC flexible RT language and HMI Advanced user interface language of SINUMERIK controls are synchronized. Please note the following:

Note

For SINUMERIK controls under HMI Advanced, only the following WinCC flexible configuration languages are relevant:

Czech

French

Italian

Polish

English

Spanish

Hungarian

Swedish

Portuguese

Dutch

Finnish

Danish

Japanese

Korean

Chinese (PRC)

Chinese (traditional)

3.3 Communications settings without integration in HMI Advanced

See also:

Setting up a user database (GUD-Tool)

There are operating scenarios where WinCC flexible Runtime runs with a SINUMERIK connection without the WinCC flexible Runtime being integrated in the HMI Advanced. In this case, the settings for the communication channel must be made by hand in file mmc.ini.

This scenario also occurs when the GUD_Tool is used without an HMI Advanced running at the same time.

Please note the following points:

- Selecting the HW interface on the NCU

  -X101 (OPI interface - 1.5 MBaud) or
  -X122 (MPI interface - 187 KBaud)

  The PG/PC interface must be correctly parameterized for the transmission rate.

1. Correct setting of the MPI address in file mmc.ini

   File mmc.ini must be adapted in directory
   <drive>:\siemens\sinumerik\hmi_advanced\add_on. The following settings are required:

   OPI interface

   [NCU840D]
   ADDRESS0=13,NAME=/NC,SAP=d
   ADDRESS1=13,NAME=/PLC,MAX_BUB_CYCLICS=0

   MPI interface

   [NCU840D]
   ADDRESS0=3,NAME=/NC,SAP=d
   ADDRESS1=2,NAME=/PLC,MAX_BUB_CYCLICS=0

See also

- Creating a user database (GUD-Tool) (Page 119)

3.4 PLC string

If PLC string tags are defined for a SINUMERIK PLC, please note:

In contrast to the SIMATIC communication channel, in the case of the SINUMERIK communication channel, the maximum length of the PLC string in a transmission byte is taken into account.

Note

If, in the byte, the maximum length is represented by a zero, then no data will be transmitted by the PLC either. The maximum length of the PLC string must always be entered in the byte.

3.5 Configuring HMI Advanced applications

Within a WinCC flexible application, it is possible to perform a task switch to all applications managed by the master control.

To do this, you can assign a script (in which you execute the corresponding HMI Advanced application (task)) to an event (e.g., "Press key") when configuring the WinCC flexible application.

Example

This VB script must then be written as follows:

```
'Easy Regie Task switching
Dim g_testObj
Dim task
Dim ret

'Create RegieServer - Objekt
Set g_testObj = CreateObject ("VBScServer.VBscriptSvr.1")

'switch to task 0
task = 0
g_testObj.SwitchToTask task, ret
'check your return -code : 0 = OK, 1 = FALSE
'msgbox ret

'Important to destroy objekt!!
Set g_testObj = Nothing
```

The 8th line (task = 0) is used to assign the HMI Advanced application. The assignment of the task numbers to the HMI Advanced application is contained within the "regie.ini" file.

3.6 PCU50 power up with WinCC flexible

Introduction

If PCU 50 power up is to start a configuration with WinCC flexible instead of with the HMI user interface, proceed as follows:

Proceed as follows

Under Windows XP, the following registry entries must be changed:

- HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\SINUMERIK\Basesoftware\08.00.00.00 \HMIManager\CheckSINDesktopPassword=0
- HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\SINUMERIK\Basesoftware\08.00.00.00 \HMIManager\ShowSINDesktopDialog=0
- HKEY_LOCAL_MACHINE\SOFTWARE\Siemens\SINUMERIK\Basesoftware\08.00.00.00 \HMIManager\StartSINHMI=0

Furthermore, a link is to be created in

C:\Documents and Settings\All Users\Startmenu\Programs\Startup for

E:\Program Files\SIEMENS\SIMATIC WinCC flexible\WinCC flexible 2007 Runtime\hmirtm.exe.

The project to be started can be determined in file hmirtm.ini in the directory ' E:\Program Files\SIEMENS\SIMATIC WinCC flexible\WinCC flexible 2007 Runtime'.

For example:

[configuration]
LoadConfigFile= E:\Program Files\SIEMENS\SIMATIC WinCC flexible\WinCC flexible 2007\Samples\SINUMERIK\OP012_Demo.Bediengerät_1fwx
This fwx is started in ramp-up mode!

Following this procedure, the Windows-type key combinations such as ALT-CTRL-DELETE or ALT-TAB are not disabled.

If these key combinations are to be disabled, this can be executed in the configuration under device settings "Disable program switchover". The disable is cancelled once Runtime has finished. Finishing Runtime may be coupled with a certain protection level.

3.7 Special keys (Alarm Cancel, Channel)

Note

If you want to make use of the configurable keys "Alarm Cancel" or "Channel", these buttons must be enabled for the WinCC flexible application. You must also specify in the associated master control that these keys are no longer to be processed by the master control (see also Section 1.5.4 Integration into SINUMERIK HMI Advanced, entries in Regie.ini) but have been transferred to the application instead. This is done using the "GimmeKeys mechanism". This will result in these keys losing their standard functions.

3.8 Jumping specifically to individual WinCC flexible screens

Introduction

You may wish to show specific WinCC flexible screens, despite the fact that this is not currently possible using function keys. For this purpose, WinCC flexible features an image number store mechanism.

Fact

A range pointer is used to define an image number store to which the current image number is always written during runtime. The controller is then able to recognize which screen is active on the HMI device.

Example

Conversely, it is also possible to trigger a screen change on the HMI device via the PLC. For this purpose, the "Activate screen" function must be configured in WinCC flexible to allow the value of a tag to be changed. The PLC must now change the corresponding trigger tag and transfer the number of the screen required to another tag.

This mechanism can also be used in conjunction with PLC hotkeys to enable specific WinCC flexible screens to be displayed via these keys.

Note

Configuration of the PLC hotkeys is described in:

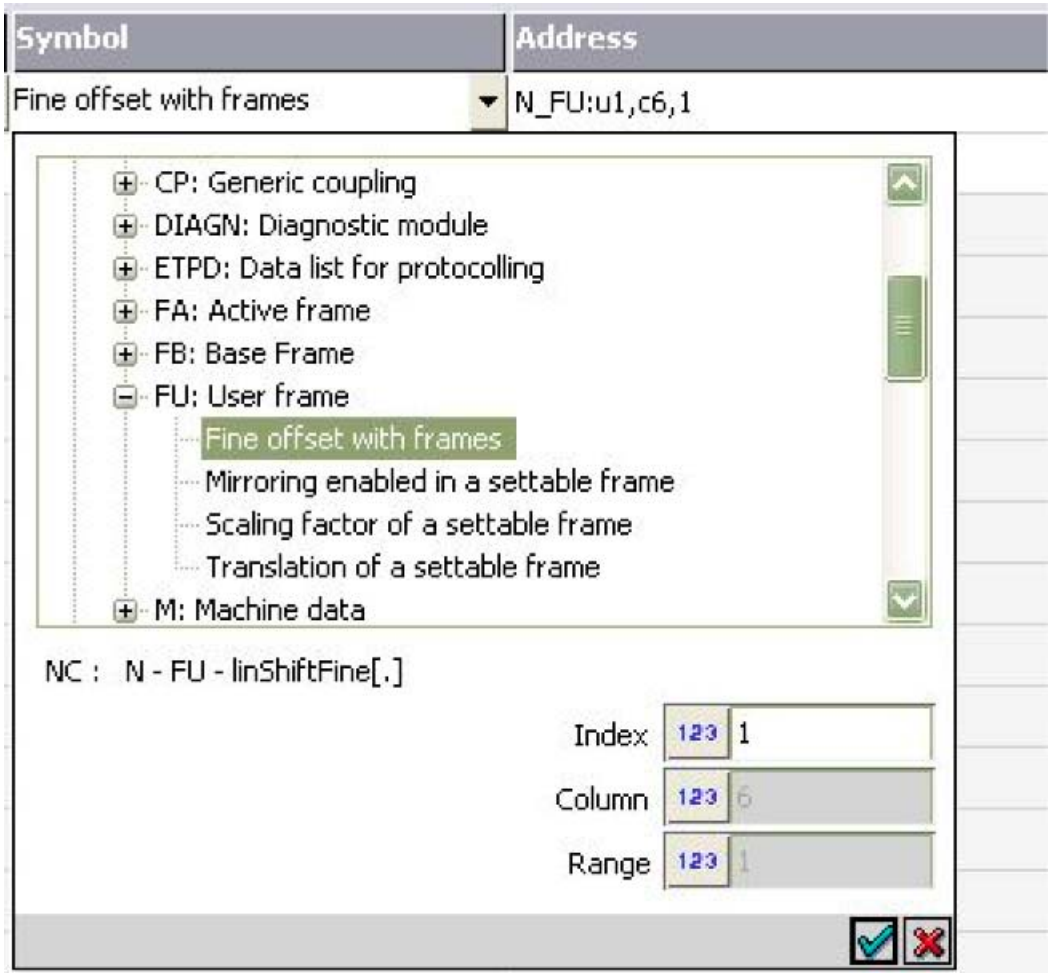
References: /IAM/ Installation & Startup Guide HMI/MMC

/BE1/ Expanding the Operator Interface

3.9 Configuring Userframes

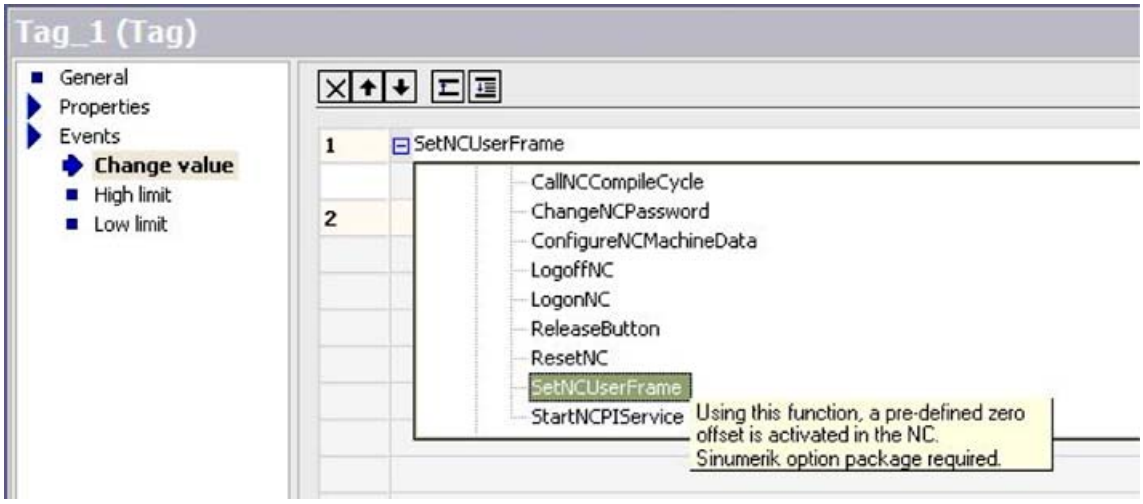
Overview

Please note the following when configuring Userframes.



When the settable zero offsets have been written, the new values are first temporarily stored and must then be activated with PI Service SETUFR.

The configurable function **SetNCUserFrame** is available for this.



This function should be configured in such a way that when a value is changed the corresponding Frame tags are then called up.

3.10 SINUMERIK functions in scripts

If the additional SINUMERIK functions are to be used in a script and support for PC systems **and** for CE panels was selected during installation, please pay attention to the following:

When a script is compiled, the SINUMERIK functions for CE panel are always used and thus cause a syntax error if you use a PC system. To avoid this, remove the function descriptions for CE panel from the relevant directory.

Remove and back up the following files:

<InstDir>\WinCC flexible 2007\Scripting\SimaticNC-Functions*.xml

Tools supplied

4

Overview

The following tools are supplied with the SINUMERIK HMI WinCC flexible configuration software:

- Create user database (GUD_TOOL)
- Call WinCC flexible screens under HMI (Switchtp)
- Toolbox (insert SINUMERIK controllers in HW Config)

4.1 Creating a user database (GUD-Tool)

Introduction

The GUD tool (gud_tool.exe) is installed by Setup in the WinCC flexible installation directory.

If the configuration PC is connected to the controller, the GUD tool can be started directly from the configuration PC.

If the configuration PC is not connected to the controller, the GUD tool must be started directly from the HMI device in service mode. For this reason, in SINUMERIK HMI WinCC flexible 2007 and higher, the setup also installs the GUD tool in the WinCC flexible Runtime directory by setup.

In both cases, the dynamic user data are used to generate a user database (<filename>.mdb) and an NSK file (<filename>.nsk) in the NCK of the control.

The user database (*.mdb) is required to display the user data within WinCC flexible during configuration of the tags and the NSK file is required during runtime of the project configured with WinCC flexible.

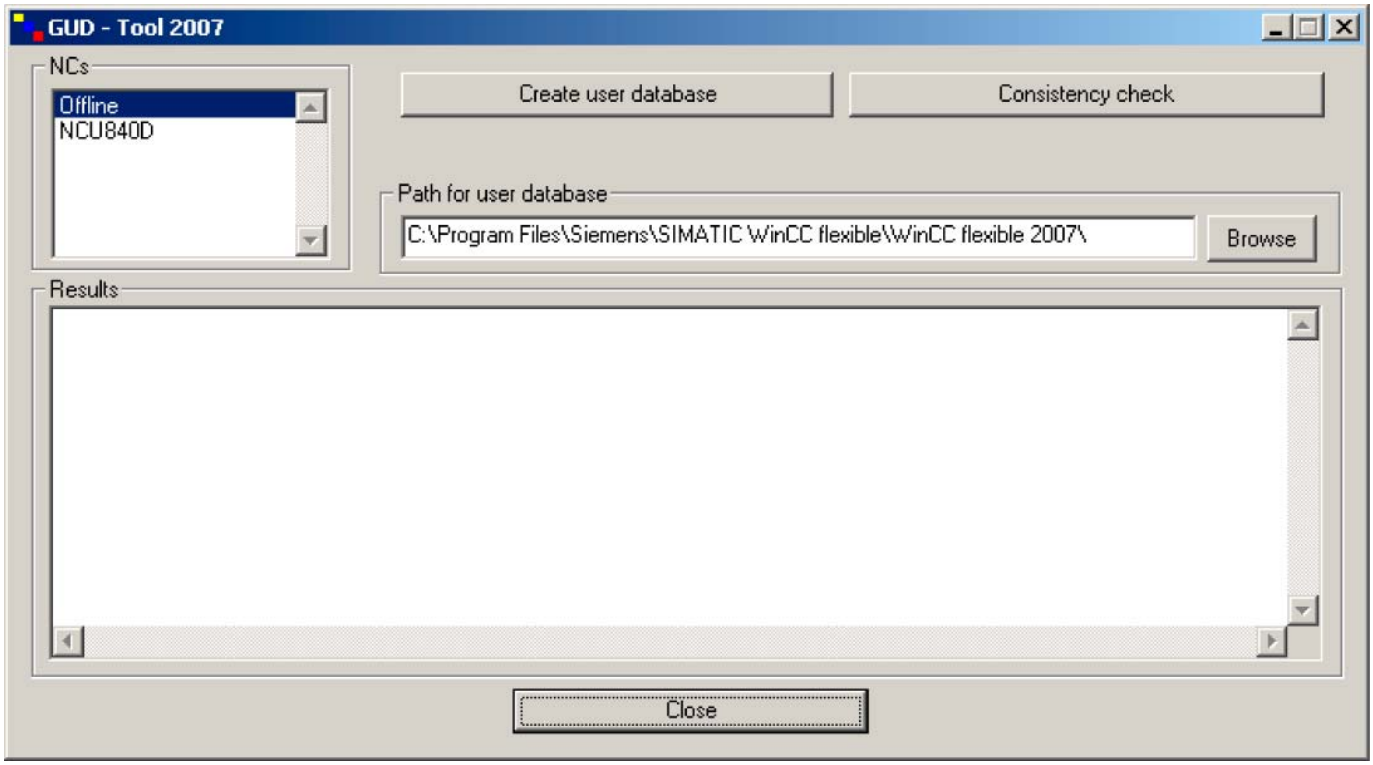
Changes in Version 2007

- The GUD tool features an offline mode
- The GUD tool can be installed together with the runtime on a PCU 50.

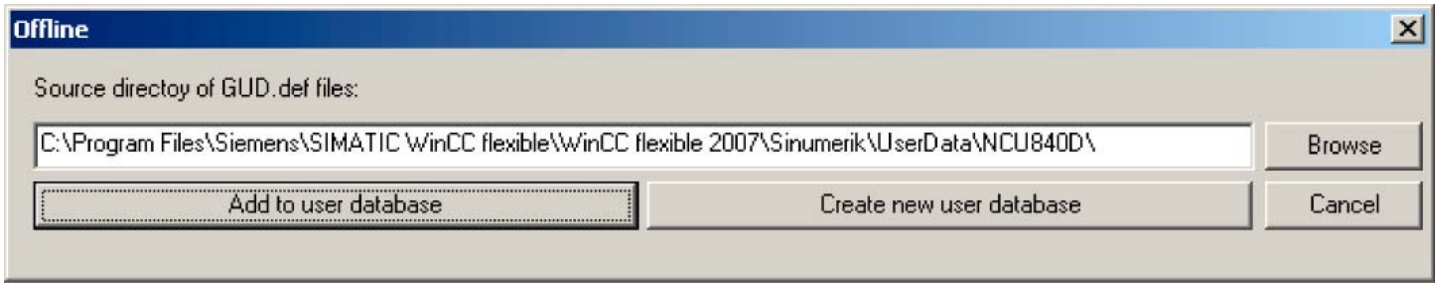
Offline mode on the configuring PC

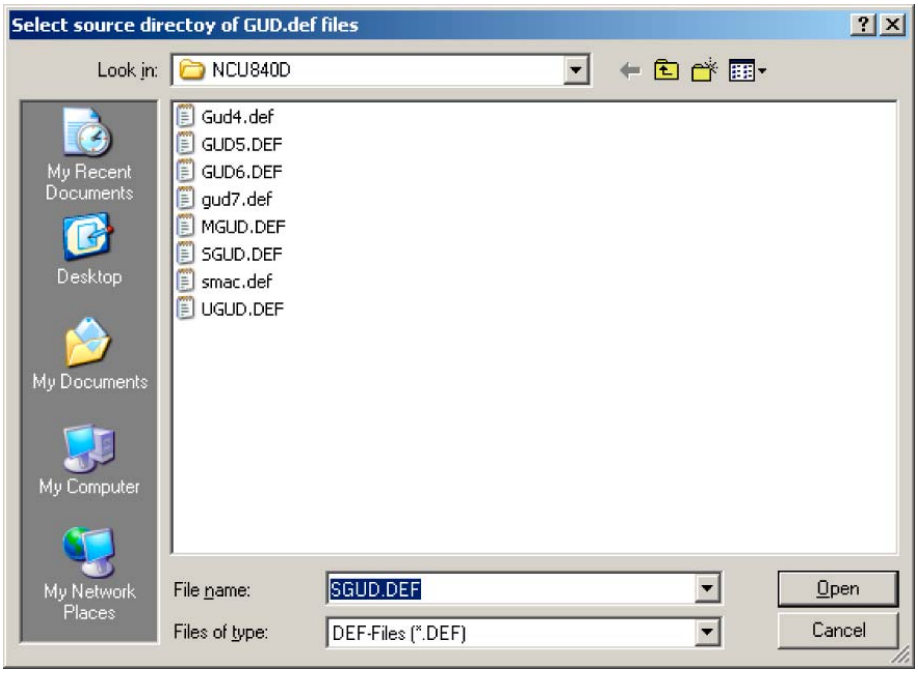
With the GUD tool for SINUMERIK HMI WinCC flexible 2007 it is possible during the configuration phase to import the user data in the form of DEF files into the user data database without first loading it into the control.

The configuring engineer can then write the user data that he or she needs for the configuration into DEF files using a text editor and then read the DEF files into the user data database with the GUD tool. The advantage of this is that configuring can be performed before the data are available on the control. Only if the project is to run on an HMI operating device does all user data first have to be loaded onto the control. If the dynamic user data loaded onto the control differs from that read into the user data database in offline mode, the user data database has to be updated with the GUD tool and the project regenerated with the current user data database.



If you want to read dynamic user data into the user database **Offline**, you must first create DEF files (*.def) out of them. To open this dialog box, select **Offline** in the NCs field and then press the **Create user database** button.



Field/Button	Definition/Action
Source directory of GUD.def files	This is where you enter the directory with the DEF files to be read in. The last directory name is interpreted as a control. The default setting is <path for user database>\Sinumerik\UserData\< *1st control in NCs field*>
Browse	Opens the dialog box for setting the DEF file directory  1. In the **Look in:** dropdown list create the required directory 2. If necessary, select a subdirectory by double-clicking it 3. Close the dialog box with **Open** The file you have entered in the **File name**does not necessarily have to exist. All the DEF files in the selected directory are read in!
Add to user database	If a directory is set the GUD tool reads out the DEF files and adds the user data to the existing user data database (control>.mdb) under <path for user database>\. The user database is generated if it does not already exist. NSK file and HmiRTm.ini entries do not have to be made here!
Create new user database	If a directory is set the GUD tool reads out the DEF files and adds the user data to the existing user database (control>.mdb) under <path for user database>\. If a user database already exists it is overwritten! NSK file and HmiRTm.ini entries do not have to be made here!
Cancel	Takes you back to the GUD Tool dialog box

Applications

- Generating the entire user data database offline
- Offline addition of data to user data databases created online, for example, additional DEF files

Installing the GUD tool on the PCU 50

In SINUMERIK HMI WinCC flexible 2007, the GUD tool is also installed on an HMI device (PCU 50) in which only the SINUMERIK HMI WinCC flexible runtime is installed.

The configuring engineer is then no longer forced to connect a configuring PC to the control whenever he or she wants to change dynamic user data on the control. It is much easier for the configurer to read out the dynamic user data from the control directly on the HMI device (PCU 50).

For configuration purposes, the user data database (*.mdb) generated on the PCU50 must then be copied to the configuring PC and entered in the field "User Database" next to the connection.

For the GUD consistency check, the GUD information must entered from the HmiRTm.ini file in the HmiRTm.ini file of the WinCC flexible Runtime directory of the HMI devices on which the configuration is to run.

To be able to call up the GUD tool, you must start the PCU 50 in service mode.

Operating sequences

1. Start the program - e.g. using the menu **Start > SIMATIC > WinCC flexible 2007 > Sinumerik HMI configuring package WinCC flexible > gud tool**.
   A dialog box is displayed (with the following screen). The dialog box displays the controllers entered in "mmc.ini" or "netnames.ini" and the directory for the user database to be created is pre-assigned the installation directory of WinCC flexible.

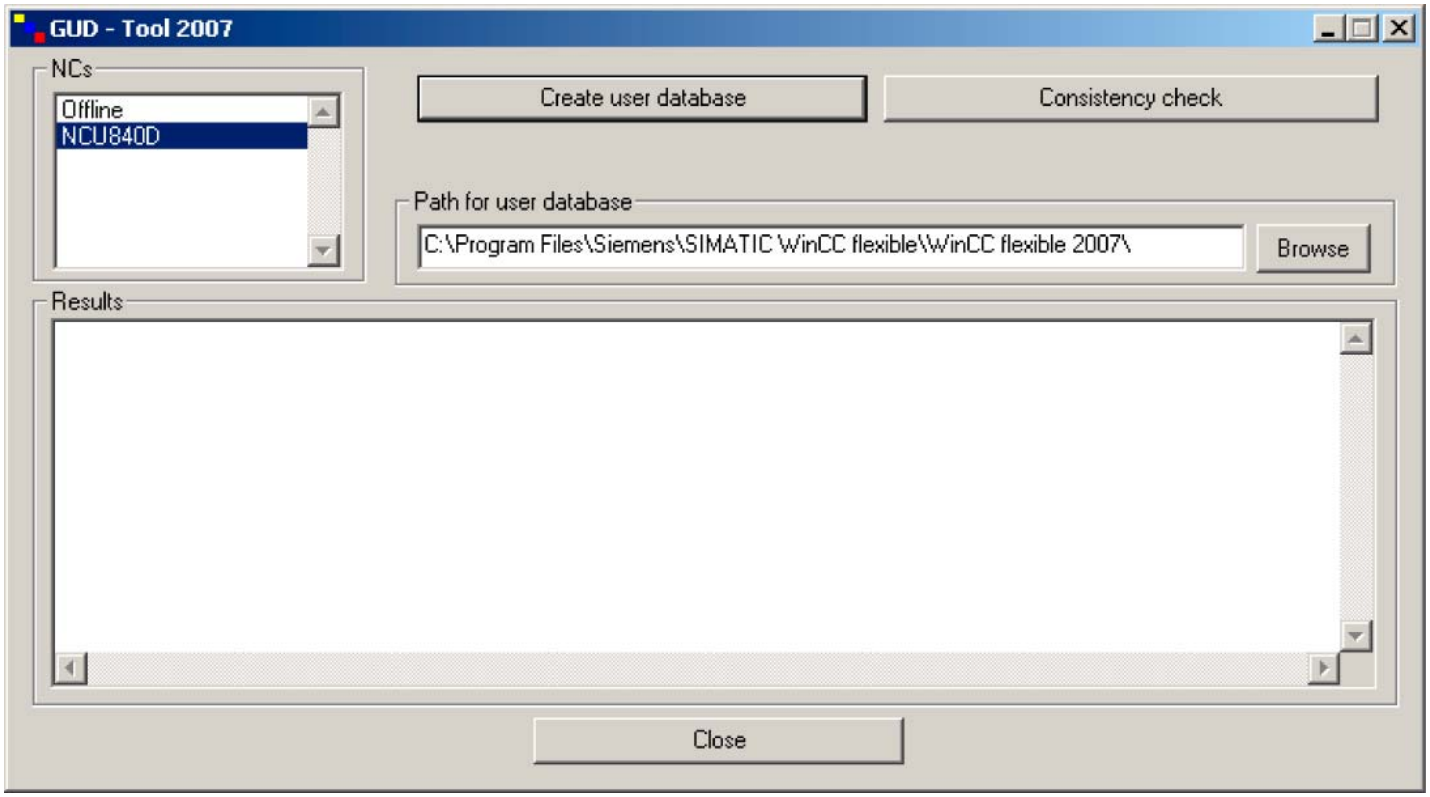


Figure 4-1 1:1 connection

2. You can enter a directory to save a user database in the **Path for user database**. if the field is empty, the GUD tool selects the directory for the user database in which the GUD tool is located.
3. Press the button **Create user database**. The GUD tool sets-up a user database as well as an associated *.nsk file.

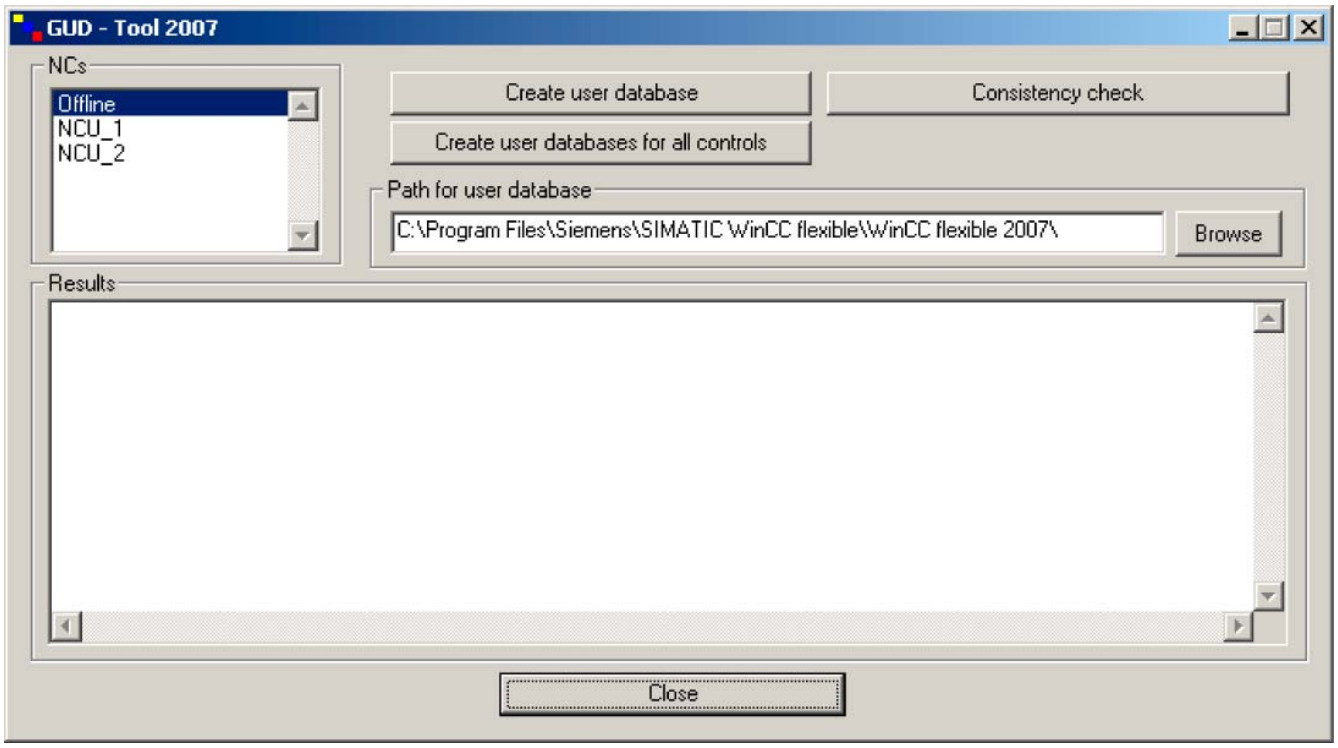


Figure 4-2 M:N connection

If you press the **Create user data base for all controls** button (for M:N connections only), the user databases of all the listed controls will be generated or updated irrespective of the item selected in the NCs field. The offline data are ignored.

4. By pressing the **Consistency check** button, you can select the user data blocks that are to be tested (refer to the following diagram).

Note

The consistency check is disabled as standard. If the check is to be activated, the file "hmirtm.ini" is appropriately adapted.

To do this, the file must be located in the runtime directory. If this file is still not available, then the GUD tool generates a new "hmirtm.ini" in its directory. You must then incorporate the entries generated in this file in the "hmirtm.ini" on the operator device.

Figure 4-3 Check Consistency

5. Parameterize consistency check, as required

Consistency check

By default, all user blocks (GUD1 to GUD9) are checked for consistency. If certain blocks are to be eliminated from the check, they can be selected here.

Result

- User database for each control (<control>*)
- NSK file for each control (<control>.nsk)
- Consistency check data updated in file hmirtm.ini

Generating the user data database on the configuring PC

After you have generated the user data database on the configring PC, you must transfer the NSK file (for m:n one for each control) to the relevant HMI device.

If the settings for the consistency check have been changed, the content of hmirtm.ini file that has been modified must also be copied to the HmiRTm.ini file of the runtime directory on the HMI device.

Generating the user data database on the HMI device (PCU 50)

Once you have generated the user data database on the HMI device, you must transfer it from the HMI device to the configuring PC. The NSK file remains on the HMI operating device.

Note

When transferring/copying to the configuration device / HMI device by floppy disk or the network, always ensure that the files are copied to the correct directories.

4.2 Call WinCC flexible screens under HMI (switchtp)

4.2.1 Switchtp (screens)

Introduction

With application "Switchtp.exe", you can change to WinCC flexible RT using the free softkeys of the SINUMERIK HMI operator interface and open a specific WinCC flexible screen.

This TOOL is supplied on the CD that is supplied with the SINUMERIK HMI configuring package WinCC flexible.

This application activates the WinCC flexible task and writes the required image number to a configurable PLC data word.

The application can be assigned to several softkeys, enabling it to bring various WinCC flexible screens to the foreground.

4.2.2 Initialize application

Introduction

To switch to WinCC flexible RT using "Switchtp.exe" and non-assigned softkeys on the SINUMERIK HMI interface and select a specific WinCC flexible screen, the following settings must be made:

Proceed as follows

1. "Switchtp.exe" application is parameterized via an INI file (Switchtp.ini).
   The following parameters must be set in this file:
   [CONFIG]
   TaskToStart=34
   PLCTargetAdress=/Plc/Datablock/Word[c99,0]
   ValueForTask7=1
   ValueForTask8=2
   ValueForTask9=3
   - TaskToStart=34 means that Switchtp.exe is started as Task34, i.e., it is not possible to select it via a softkey.
   - PLCTargetAdress=/Plc/Datablock/Word[c99,0] specifies the PLC data word, to which the required image number is written. In this example: DB99.DBW0.
   - ValueForTask7=1 means that horizontal softkey7 (1st softkey=0) is used to activate the WinCC flexible task and screen number 1 is written to PLCTargetAdress.
2. To ensure that WinCC flexible selects the chosen screen following activation, a PLC tag must be created within the WinCC flexible project, making specific reference to the

PLCTargetAddress.
It is a characteristic of these tags that if a value is changed,
the "Bildanwahl_variabel" ("Open_screen_variable") function must also be configured with the "Tag with image number" parameter. As the "Tag with image number" parameter, the new tag must be configured for PLCTargetAdress.

3. Identify application with the horizontal softkeys used in file F:\OEM\regie.ini.
   To do this, the following entries must be made, for example, in this "Regie.ini" file in the [TaskConfiguration] section:
   [TaskConfiguration]

   Task34= name := oemframe,CmdLine:= "E:\\Program Files\\Siemens\\Simatic WinCC Flexible\\WinCC flexible 2007 Runtime\\HmiRTm.exe F:\\OEM\\BildNr.fwx",ClassName := "HmiRTm",TimeOut := 20000,PreLoad := False,HeaderOnTop := False,DisableKeys:=196863

   Task7= name := switchtp, TimeOut := 20000,PreLoad := False,HeaderOnTop := True

   Task8= name := switchtp, TimeOut := 20000,PreLoad := False,HeaderOnTop := True

   Task9= name := switchtp, TimeOut := 20000,PreLoad := False,HeaderOnTop := True

   – Task34..... is the WinCC flexible application to be started from Switchtp.ini -> TaskToStart=34
   – Taskx= name := switchtp... the corresponding WinCC flexible horizontal softkey is activated. There may be more than one instance of this entry with different numbers.

4. Label the required softkeys for all languages used.
   In the case of German, use the following text, for example, in directory F:\Oem\Language\re_gr.txt
   [HSoftkeyTexts]
   HSK7="BILD 1"
   HSK8="BILD 2"
   HSK9="BILD 3"

5. Copy the application (Switchtp.exe) and the INI file (Switchtp.ini) to directory F:\Oem.

4.3 TOOLBOX

Introduction

The "TOOLBOX" expands the hardware catalog of STEP7 by adding SINUMERIK components such as SINUMERIK 840D or 840D sl

These expansions enable WinCC flexible to be configured with the "SINUMERIK-PLC" protocol integrated in STEP7.

STEP7 integration makes the following configurations possible in WinCC:

- Symbolic addressing of STEP7 tags/data
- Use Alarm_S message class

Installation

The "TOOLBOX" is installed automatically by the SINUMERIK HMI WinCC flexible configuration software SETUP.

Appendix

A

A.1 SINUMERIK functions for HMI CE devices

Introduction

The following tables list the functions that are relevant for the individual SINUMERIK HMI CE devices.

See also

- Functions (Page 80)

SINUMERIK functions for HMI CE devices

Table A-1 SINUMERIK functions for HMI CE devices

Functions
SetS7OperatingMode SetzeS7Betriebsart
AcknowledgeNCCancelAlarms QuittiereNCCancelAlarms
LogonNC AnmeldenNC
LogoffNC AbmeldenNC
ChangeNCPassword ChangeNCPassword
SetNCUserFrame SetNCUserFrame
SelectNCPartProgram WaehleNCTeileprogramm
PrepareActualNCBlockForDisplay VorbereitenAktuellerNCSatzFuerAnzeige
StartNCPIService StarteNCPIDienst
ResetNC NeustartenNC

Functions
ConfigureNCMachineData ConfigureNCMaschineData
ScreenObjectCursorUp
ScreenObjectCursorDown
ScreenObjectPageUp
ScreenObjectPageDown
PartProgramListSelectNCPartProgram TeileProgrammListeWaehleNCTeileprogramm
PartProgramListDirUp TeileprogrammListeOrdnerNachOben
PartProgramListDirDown TeileprogrammListeOrdnerNachUnten
PartProgramListRefresh TeileprogrammListeAktualisieren

A.2 SINUMERIK events, which trigger HMI CE device functions

Introduction

The following tables list the functions for the relevant WinCC flexible objects and associated events that apply in the case of SINUMERIK with HMI CE devices. These functions are indicated by a **crpss** in the table.

There are five tables.

SINUMERIK events, which trigger HMI CE device functions, part 1

Object	Tags			Softkey			
				Global		Local	
Event	Value changed	Upper limit exceeded	Lower limit exceeded	Release	Press	Release	Press
Functions							
SetS7OperatingMode	X	X	X	X	X	X	X
AcknowledgeNCCancelAlarms	X	X	X	X	X	X	X
LogonNC	X	X	X	X	X	X	X
LogoffNC	X	X	X	X	X	X	X
ChangeNCPassword	X	X	X	X	X	X	X
SetNCUserFrame	X	X	X	X	X	X	X
SelectNCPartProgram	X	X	X	X	X	X	X
PrepareActualNCBlockForDisplay							
StartNCPIService	X	X	X	X	X	X	X
ResetNC	X	X	X	X	X	X	X
ConfigureNCMachineData	X	X	X	X	X	X	X
ScreenObjectCursorUp				X	X	X	X
ScreenObjectCursorDown				X	X	X	X
ScreenObjectPageUp				X	X	X	X
ScreenObjectPageDown				X	X	X	X
PartProgramListSelectNCPartProgram				X	X	X	X
PartProgramListDirUp				X	X	X	X
PartProgramListDirDown				X	X	X	X
PartProgramListRefresh				X	X	X	X

SINUMERIK events, which trigger HMI CE device functions, part 2

Object	Command buttons			
	Global		Local	
Event	Release	Press	Release	Press
Function				
SetS7OperatingMode	X	X	X	X
AcknowledgeNCCancelAlarms	X	X	X	X
LogonNC	X	X	X	X
LogoffNC	X	X	X	X
ChangeNCPassword	X	X	X	X
SetNCUserFrame	X	X	X	X
SelectNCPartProgram	X	X	X	X
StartNCPIService	X	X	X	X
ResetNC	X	X	X	X
ConfigureNCMachineData	X	X	X	X

SINUMERIK events, which trigger HMI CE device functions, part 3

Object	Image		Messages						
			Alarm				Event		
Events	Loaded	Cleared	Activate	Clear	Acknowledge	Edit	Activate	Clear	Edit
Functions									
LogonNC	X	X							
LogoffNC	X	X							
ChangeNCPassword	X	X							
SetNCUserFrame	X	X	X	X	X	X	X	X	X
SelectNCPartProgram	X	X	X	X	X	X	X	X	X
PrepareActualNCBlockForDisplay	X								
StartNCPIService	X	X	X	X	X	X	X	X	X
ResetNC	X	X	X	X	X	X	X	X	X
ConfigureNCMachineData	X	X	X	X	X	X	X	X	X
ScreenObjectCursorUp	X	X							
ScreenObjectCursorDown	X	X							
ScreenObjectPageUp	X	X							
ScreenObjectPageDown	X	X							
PartProgramListSelectNCPartProgram	X	X							
PartProgramListDirUp	X	X							
PartProgramListDirDown	X	X							
PartProgramListRefresh	X	X							

SINUMERIK events, which trigger HMI CE device functions, part 4

Objects	OCX			Script
	OCX (ScreenItem)			InScripts
Event	Press, Release, Click, Change (or Toggle when used on a Switch), SwitchOn, SwitchOff	Activate	Deactivate	
Functions				
SetS7OperatingMode	X	X	X	X
AcknowledgeNCCancelAlarms	X	X	X	X
LogonNC	X	X	X	X
LogoffNC	X	X	X	X
ChangeNCPassword	X	X	X	X
SetNCUserFrame	X	X	X	X
SelectNCPartProgram	X	X	X	X
PrepareActualNCBlockForDisplay				
StartNCPIService	X	X	X	X
ResetNC	X	X	X	X
ConfigureNCMachineData	X	X	X	X
ScreenObjectCursorUp	X	X	X	
ScreenObjectCursorDown	X	X	X	
ScreenObjectPageUp	X	X	X	
ScreenObjectPageDown	X	X	X	
PartProgramListSelectNCPartProgram	X	X	X	
PartProgramListDirUp	X	X	X	
PartProgramListDirDown	X	X	X	
PartProgramListRefresh	X	X	X	

SINUMERIK events, which trigger HMI CE device functions, part 5

Object	Scheduler					
Event	ExpireTime	ScreenChange	ChangeUser	AlarmOverflowBuffer	StartUp	ShutDown
Function						
LogoffNC			X			X

A.3 Description of section [TaskConfiguration] of regie.ini

Overview

Section [TaskConfiguration] contains all area applications and other applications that are to be started by the master control. For each application, one or more attributes can be specified (separated by a comma) that describe special properties of the application in more detail.

Entries

The individual applications are started in the order in which they are numbered by the *entry identifier*Task (e.g. Task5).

For example, an application configured with the entry
Task5 = Name := IB

can assume during its powering up that an application configured with
Task2 = Name := SERVICES
already exists in the system.

Note

The following exception exists: The application entered in section [Miscellaneous] under the entry PoweronTaskIndex (usually this will be Task0) is the last to be started in the power-up. The avoids having to change the area during power-up.

Entries for OEM

The entries in the range 0 to 23 are reserved for tasks that can be activated by means of the area menu. The remaining numbers (24 to 63) are available for so-called child-applications. Numbers 51 through 63 are available for OEM customers for their child applications.

Softkey for master control

The softkey assignment of the master control is defined by the task entries: The task with the lowest number is assigned to the machine area button, Task1 on softkey 1 etc. up to Task 23 which is assigned to the right-most key on the third extension of the softkey bar.

Labeling of softkeys

The labeling of the softkeys is defined in file *language*.INI. One *language*.INI (e.g. RE_GR.INI for German) exists in subdirectory LANGUAGE for each language. Only if a softkey text has been entered in the *language*.INI can a configured application actually be started via this softkey.

Attributes

The attributes of the entries in the task configuration are listed in the table.

Attributes of the task configuration

Attribute	Significance
Name	Name of the application as it appears in the file manager, but without file extension ".EXE".
Timeout	Maximum duration for power-up in milliseconds The default is 10000 (10 secs.)
CmdLine	Parameter for the client application: Name of an application in OEM Frame
HeaderOnTop	Where the header for an application appears
PreLoad	Start an application during system start-up
TerminateTask	Remove other applications from the memory when starting an application
AccessLevel	Access level
ClassName	Class name of an application
WindowName	Windowname of an application
ShowTask	Visibility of the application. Default TRUE
GIMMEKEY	Release mask for keys of the master control that are to be handled by the application
ShowAppMenu key	Release mask for the area switchover key that is to be used by the application
<empty>	Usually cancels the configuration of the task in question defined in file \mmc2\regie.ini. This task is then no longer selectable in the area switchover bar.

Attribute Name:

Name of the application as listed in the file manager (without specifying the file extension '.EXE').

Attribute Timeout:

Duration (in milliseconds) made available by the master control for no longer than the time it takes to start up the application.

Attribute CmdLine

With the attribute CmdLine, parameters can also be introduced for the client application: As the command line for the area applications. The default is the empty string "".

This is a static assignment.

Dynamically, command line parameters can be assigned to a task with the functions WriteCmdLine of REGIE.DLL, which can be read with

the functions ReadCmdLine. The following applies to standard DOS and WINDOWS applications: The proxy OEMFRAME is entered as the name of the task and the default DOS program name or WINDOWS task name is transferred with the attribute CmdLine.

Example: Attribute CmdLine

```
[TaskConfiguration]
Task2 = Name := OEMFRAME, CmdLine := "EXCEL.EXE TAB.XLS"
; The spreadsheet program EXCEL is called via
; OEMFRAME.
```

Attribute ClassName and WindowName:

The attributes WindowName and ClassName are used in conjunction with the attribute CmdLine if a WINDOWS program is to be integrated as the area application under the proxy application OEMFRAME.

Example: Attribute ClassName and WindowName

```
[TaskConfiguration]
Task2 = Name := OEMFRAME, CmdLine := "WinWord",ClassName :=
"OpusApp",
WindowName := "Microsoft Word"
; Calls the text processing program WinWord with ClassName and
; WindowsName.
```

The statements about task2 must be written in one line.

For client applications that exist in source code, this information can be accessed directly, and must be checked for in the two Windows API calls *RegisterClass* and *CreateWindow*.

Note

If no source code exists, these character strings can be derived with public domain tools such as WinWalk.

Note

Start of an area application essentially correlates to instruction WinExec (lpszCmdLine, SW_SHOW).

For a more detailed explanation, see the description of parameter *lpszCmdLine* of WinExec in the "Programmer's Reference Manual, Volume 2: Functions" /WPR/.

Black screen with hourglass.

Note

If you enter an application in REGIE.INI and a black screen with hourglass appears when you start the application, this means that the application is integrated into the Siemens sequence control and is also being started via an OEM Frame.

Remedy: Integrate the application directly into REGIE.INI without OEM Frame.

Attribute HeaderOnTop:

With this optional attribute, you can define whether the application *Header* (displays the operating area, operating mode etc.) is inactive (HeaderOnTop = False) or active (HeaderOnTop = True) for this area application.

The default setting is True, i.e. the header always appears unless it is deactivated.

Example: Attribute HeaderOnTop

```
[TaskConfiguration]
Task5 = Name := MACHINE, HeaderOnTop := True
; Application Machine with Header
Task6 = Name := DP, HeaderOnTop := False
```

```
; Dialog Programming without Header
Task7 = Name := aeditor
; ASCII Editor with Header (default: Header active)
```

Attribute PreLoad

With this optional attribute, you can prevent an area application from starting when the HMI system starts up.

If this attribute is not set (PreLoad := False), the area application is not started until it is activated for the first time.

Advantage:

HMI system start-up can be speeded up in this way.

Disadvantage:

During HMI system is start-up, it is not possible to ascertain whether all configured area applications can be handled during runtime. The default is: PreLoad := True, i.e. applications without this attribute are started during start-up.

Example: Attribute PreLoad

```
[TaskConfiguration]
Task5 = Name := SIMULA, PreLoad := False
; Operating area Simulation (SIMULA) is configured in REGIE.INI
; but is not started during system start-up.
```

Note

This attribute should only be set to *False* for development and testing purposes. In a product variant, all operating areas should be started during system start-up for safety reasons. This is taken into account by the default setting.

Attribute Terminate Tasks:

With this attribute the user can remove certain area applications from the memory when starting other memory applications. This solves the problem of limited WINDOWS resources.

Removing an area application

Example: Removing an area application

```
[TaskConfiguration]
Task6 = name := param, TerminateTasks := dp;
When application PARAM.EXE is called, application
; DP.EXE is removed from the memory.
```

Removing multiple area applications

To remove multiple applications simultaneously, place their names, separated by comma, in brackets.

Example: Removing multiple area applications

```
[TaskConfiguration]
Task7 = name := dp, TerminateTasks := (param, dg);
```

```
When application DP.EXE is called, the
; applications PARAM.EXE and DG.EXE are removed from the memory.
```

Mutual exclusion of two applications

If two applications of the same name are to exclude each other, their names must be extended with their task name placed after a colon.

Example of mutual exclusion of two applications

```
[TaskConfiguration]
Task8 = name := oemframe, cmdline := "EXCEL.EXE",
TerminateTasks := oemframe:12
Task12 = name := oemframe, cmdline := "NOTEPAD.EXE",
TerminateTasks := oemframe:8;
Two applications run alternately under OEMFRAME: Either
; EXCEL or NOTEPAD. If OEMFRAME with EXCEL is called as Task 8
; OEMFRAME exits the memory with NOTEPAD as Task 12 and
; vice versa.
```

The statements about Task8 and Task12 must be written in a single line.

Removing multiply active applications

If multiple applications are to be removed at the same time and one of them, e.g., is multiply active under different task numbers, their names, separated by commata, must be placed in brackets and, if necessary, complemented by their task number.

Example: Removing multiply active applications

```
[TaskConfiguration]
Task3 = name := services, TerminateTasks := (rh, aeditor:25)
Task7 = name := aeditor, PreLoad := False, TerminateTasks := rh
Task25 = name := aeditor, PreLoad := False, TerminateTasks := rh;
The program ASCII editor (aeditor) is started more than once, e.g.
by services
; as Task 7, by programming as task 25. If Services
; is called as Task 3, the master control only deactivates
; the ASCII editor for programming (Task 25), but not the editor
; for services itself (Task 7). In all cases, RemoteHelp (rh)
; is deactivated.
```

Attribute AccessLevel:

With this attribute, the OEM developer can define the access rights for tasks of the master control: In SW 3.2 and higher, the softkeys for operating the master control are assigned access rights.

Softkeys for operating the master control

Machine	Parameters	Program	Services	Diagnostics	Startup

The default setting is AccessLevel := 4.

An overview of the access levels is given in the table below, the default setting is S4...

The 8 access authorization levels

Access level:	Can be accessed with:	User group
S0	System password	SIEMENS
S1	Machine manufacturer password	Machine manufacturer
S2	Service password	Startup engineer, service (machine manufacturer)
S3	End user password	Privileged end user (in-house service)
S4	Key switch position 3	Programmers
S5	Key switch position 2	Trained operator
S6	Key switch position 1	Operator
S7	Key switch position 0	Job-trained operator (NC Start/NC Stop, MCP)

Example: Attribute AccessLevel

```
[TaskConfiguration]
Task2 = Name := ib, AccessLevel := 2;
Only the machine manufacturer's commissioning engineer
; can access the area application with the
; service password.
```

Attribute ShowTask:

In the default setting start-up of the area applications is visible. If an area application is to be started iconized, the attribute **ShowTask :=FALSE** must be used. In this case, the application cannot be started via softkey.

Control of the keyboard filter

In the expansion described below, keys such as the area switchover key, the channel switchover etc. can be processed for individual area applications. So, for instance, the F10 key can be programmed for OEM applications by the OEM and area switchover can be triggered with a different key (parameterized with ShowappMenukey).

The necessary settings are made in file REGIE.INI in section [TaskConfiguration] using the new GIMMEKEYS and ShowAppMenuKey setting options:

```
...
[TaskConfiguration]
TaskX = name := oemframe, ..., GIMMEKEYS := n,
ShowAppMenuKey := m
```

Parameters m and n are bit masks, whose meaning is described by the attributes.

Attribute GIMMEKEYS

Release mask for keys of the master control that are to be handled by the application.

GIMMEKEYS:=n ,

where n represents a 32-bit bitmask that defines which of the master control keys will handle the application.

Bit0: Area switchover

1=OEMApp wants F10 for its own use; 0=OEMApp wants to use F10 in the standard way (area switchover)

Bit1: Channel switchover key

1=OEMApp wants F11 for its own use; 0=OEMApp wants to use F11 in the standard way

Bit2: Cancel key (BigMac)

1=OEMApp wants ESC for its own use; 0=OEMApp wants to use ESC in the standard way

Bit3: Machine area key

1=OEMApp wants SH-F10 for its own use; 0=OEMApp wants to use SH-F10 in the standard way

Bit4: End key instead of Tab key

1=OEMApp uses the End key; 0=OEMApp gets a Tab when the End key is pressed

Bit 5: Horizontal and vertical softkeys

1=OEMApp for own use

0=master control uses F1--F8, SH-F1--SH-F8, F9, SH--F9, F12 and Home itself

Example:

GIMMEKEYS:=15 OEMApp wants to use F10,F11,ESC,SH-F10 itself

GIMMEKEYS:=1 OEMApp wants to use F10 itself

Note

To allow an application to take over area switchover from the master control, the application is configured with GimmeKey that it would like to use the F10 key itself and at the same time the master control is informed with ShowAppMenuKey=0 that it will no longer respond.

The table below lists the codes of the keys that are released with attribute GIMMEKEYS and can therefore be directly handled by the application.

Key codes

Key	Virtual key code	Key code (hex)
1. Horizontal softkey bar	VK_F1 -- VK_F8	70 --77
2. Horizontal softkey bar	(VK_CONTROL, VK_F1) -- (VK_CONTROL, VK_F8)	(11,70) -- (11,77)
Right vertical softkey bar	(VK_SHIFT, VK_F1) -- (VK_SHIFT, VK_F8)	(10,70) -- (10,77)
Left vertical softkey bar	(VK_SHIFT, VK_CONTROL, VK_F1) -- (VK_SHIFT, VK_CONTROL, VK_F8)	(10,11,70) -- (10,11,77)
Recall-Taste	VK_F9	78
ETC Key	(VK_SHIFT, VK_F9)	(10,78)
Area switchover key	VK_F10	79
Machine area key	(VK_SHIFT, VK_F10)	(10,79)
(Alarm) cancel key	VK_ESCAPE	1B
Channel switchover key	VK_F11	7A

End key	VK_END	23
Tab key	VK_TAB	09

Attribute ShowAppMenuKey:

Release mask for the area switchover key that is to be used by the application.

ShowAppMenuKey:=m, where m represents a 32-bit wide bit mask used to define which key will activate area switchover in this application (replacing F10)

Bit0-7 virtual keycode of the area switchover key to be defined

(see winuser.h VK_xxx entries)

Bit16 1=Shift must be pressed; 0=Shift must not be pressed

Bit17 1=Ctrl must be pressed; 0=Ctrl must not be pressed

Bit18 1=Alt must be pressed; 0=Alt must not be pressed

Example:

ShowAppMenuKey := 65659

65659 = 0x1007B VK_F12 = 0x7B

à Shift-F12 triggers the area switchover key for this application

VK_F1 = 0x70, VK_F1 = 0x71, ..., VK_F24 = 0x87

Note

If another key for area switchover is assigned to an application with 'ShowAppMenuKey', the 'F10' key continues to function. If this is to be suppressed, it must be defined explicitly with 'GimmeKey'.

Attribute DisableKeys

The function DisableKeys configures the response of the keyboard filter. The keyboard filter can be used to skip certain key sequences and to map keys and softkeys. Mapping means replacement of the original keyboard code with codes generated by the keyboard filter. The keyboard filter function is task-specific. If the keyboard filter is not configured

for a particular task, the standard response of the HMI takes over for this task. The standard response is defined in file system.ini under key "seqact" in section [Keyboard].

Keyboad sequences

The keyboard sequences are activated and deactivated with DisableKeys.

The following keyboard sequences are defined:

Bit0	CTRL-ALT
Bit1	ALT-F4
Bit2	ALT-TAB
Bit3	LEFTSHIFT-ALT-TAB
Bit4	RIGHTSHIFT-ALT-

Bit5	CTRL-ESC	
Bit6	ALT-ESC	
Bit7	ALT-SPACE	
Bit8	(SHIFT)_CTRL_F1	Left vertical and lower horizontal softkey bar
Bit9	(SHIFT)_CTRL_F2	Left vertical and lower horizontal softkey bar
Bit10	(SHIFT)_CTRL_F3	Left vertical and lower horizontal softkey bar
Bit11	(SHIFT)_CTRL_F4	Left vertical and lower horizontal softkey bar
Bit12	(SHIFT)_CTRL_F5	Left vertical and lower horizontal softkey bar
Bit13	(SHIFT)_CTRL_F6	Left vertical and lower horizontal softkey bar
Bit14	(SHIFT)_CTRL_F7	Left vertical and lower horizontal softkey bar
Bit15	(SHIFT)_CTRL_F8	Left vertical and lower horizontal softkey bar
Bit16	Left Windows key (task menu opens) VK_LWIN	
Bit17	Right Windows key (task menu opens) VK_RWIN	

A keyboard sequence is suppressed if the corresponding bit is set in the DisableKeys mask. The mask must be entered as a decimal value!

Note

The first 8 sequences are required for correct functioning of HMI Advanced and must not be skipped! These sequences may only be deactivated for service requirements

The other 8 entries suppress the sequences SHIFT-CTRL-F1 to SHIFTCTRL- F8 and CTRL-F1 to CTRL-F8. On a PCU50, these correspond to the 3rd and 4th sofkey bar.

Example:

An application called app.exe is to receive the key sequences (SHIFT)-CTRL-F1 to (SHIFT)-CTRL-F8.

BitMaske = 00FFh corresponds to 255 (decimal)

Parameterization in Regie.ini:

Task8 = name := app, Timeout := 10000, DisableKeys := 255

Attribute DisableKeysHigh

Key sequences CTRL-F1 to CTRL.F8 can be mapped onto key sequences CTRL-F13 to CTRL-F20 with a mapping function. This might be necessary because the operating system already responds to CTRL-F4 and CTRL-F6 in certain situations. The key mapping is set with entry DisableKeysHigh:

Bit0-Bit28 reserved
Bit29=1: Key mapping CTRL-F1 to CTRL-F8 --> CTRL-F13 to CTRL-F20 is active
Bit30 reserved
Bit31 reserved

The mask must be entered as a decimal value!

Example:

An application called app.exe is to receive the key sequences (SHIFT)-CTRL-F1 to (SHIFT)-CTRL-F8 and the sequences CTRL-F1 to CTRL-F8 are to be mapped onto CTRL-F13 to CTRL-F20.

DisableKeys:= 00FFh corresponds to 255 (decimal)
DisableKeysHigh := 20000000h corresponds to 536870912 (decimal)
Parameteriziation in Regie.ini:
Task8 = name := app, Timeout := 10000, DisableKeys := 255, DisableKeysHigh :=536870912

Attribute <empty>:

The attribute <empty> is used to disable the configuration of a task made in file \mmc2\regie.ini -- where all standard tasks of the HMI Advanced are configured -- subsequently without deactivating file \mmc2\regie.ini. For this, the <empty> attribute is configured in the relevant task in file regie.ini in either the OEM or AddOn directory. Typically, the attribute is configured in combination with installation of OEM and AddOn products if standard tasks of HMI Advanced are no longer to be selected by softkey in the area swithover bar after installation of the OEM or AddOn product.

Example: Skipping standard task 5 (installation and start-up)

File regie.ini in the OEM or Add_On directory must contain the following configuration:

```
...
[TaskConfiguration]
Task5 = <empty>
...
```

Index

T

U

W