SIEMENS

SINUMERIK

SINUMERIK Integrate Create MyInterface (CMI)

Configuration Manual

Valid for:
Controller
SINUMERIK 840D, SINUMERIK 840D sl / 840DE sl
Software
Create MyInterface, Version 2.6.SP2 HF10

06/2018
A5E44131243B AA

Legal information

Warning notice system

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
indicates that minor personal injury can result if proper precautions are not taken.

NOTICE
indicates that property damage can result if proper precautions are not taken.

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 product/system described in this documentation may be operated only by **personnel qualified** for the specific task in accordance with the relevant documentation, in particular its warning notices and safety instructions. Qualified personnel are those who, based on their training and experience, are capable of identifying risks and avoiding potential hazards when working with these products/systems.

Proper use of Siemens products

Note the following:

⚠WARNING
Siemens products may only be used for the applications described in the catalog and in the relevant technical documentation. If products and components from other manufacturers are used, these must be recommended or approved by Siemens. Proper transport, storage, installation, assembly, commissioning, operation and maintenance are required to ensure that the products operate safely and without any problems. The permissible ambient conditions must be complied with. The information in the relevant documentation must be observed.

Trademarks

All names identified by ® are registered trademarks of 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
Division Digital Factory
Postfach 48 48
90026 NÜRNBERG
GERMANY

A5E44131243B AA
℗ 05/2018 Subject to change

Table of contents

Preface 1

SINUMERIK documentation

The SINUMERIK documentation is organized into the following categories:

- General documentation/catalogs
- User documentation
- Manufacturer/service documentation

Additional information

You can find information on the following topics at the following address (https://support.industry.siemens.com/cs/de/en/view/108464614):

- Ordering documentation/overview of documentation
- Additional links to download documents
- Using documentation online (find and search in manuals/information)

If you have any questions regarding the technical documentation (e.g. suggestions, corrections), please send an e-mail to the following address (mailto:docu.motioncontrol@siemens.com).

mySupport/Documentation

At the following address (https://support.industry.siemens.com/My/ww/en/documentation), you can find information on how to create your own individual documentation based on Siemens' content, and adapt it for your own machine documentation.

Training

At the following address (http://www.siemens.com/sitrain), you can find information about SITRAIN (Siemens training on products, systems and solutions for automation and drives).

FAQs

You can find Frequently Asked Questions in the Service&Support pages under Product Support (https://support.industry.siemens.com/cs/de/en/ps/faq).

SINUMERIK

You can find information about SINUMERIK at the following address (http://www.siemens.com/sinumerik).

The Configuration Manual addresses users of the "RPC SINUMERIK" software for PCU 50 HMI Advanced and "SINUMERIK Integrate Create MyInterface" for SINUMERIK Operate. A version for SINUMERIK Operate embedded (Linux) is not available.

Benefits

The configuration manual familiarizes the target group with the software. In addition, it enables the target group to react to malfunctions in a targeted manner and to introduce appropriate measures.

Standard scope

This documentation describes the functionality of the standard scope. Additions or revisions made by the machine manufacturer will be documented by the machine manufacturer.

Other functions not described in this documentation might be executable in the controller. This does not, however, represent an obligation to supply such functions with a new control system or when servicing.

Furthermore, for the sake of clarity, this documentation does not contain all detailed information about all product types and cannot cover every conceivable case of installation, operation or maintenance.

Technical Support

Country-specific telephone numbers for technical support are provided in the Internet at the following address (https://support.industry.siemens.com/sc/ww/en/sc/2090) in the "Contact" area.

1.1 Security information on the product

Security information on the product

NOTICE

Misuse of data

It is essential to use secure data storage when saving your data - particularly your confidential data. It is best to store this data encrypted locally or encrypted on the network. Make sure that this data cannot be accessed by unauthorized personnel.

This applies to the following data:

- Archive files
- Image files
- Project files
- Trace files
- Safety-relevant files

Further information on secure data storage can be found at Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/us/en/view/108862708).

NOTICE

Danger due to insecure Internet connection

Before establishing a network connection, ensure your PC is connected to the Internet via a secure connection. Pay attention to the security-relevant notes. Further information about communications security can be found at Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/us/en/view/108862708).

NOTICE

Security risk through administrator rights

If you operate a PC with permanent administrator rights, there is an increased security risk. Therefore, ensure that there is a secure operating environment, and assign administrator rights only temporarily and selectively.

Further information on the secure administration of user accounts and the assignment of rights can be found at Industrial Security Configuration Manual (https://support.industry.siemens.com/cs/us/en/view/108862708).

EC Declaration of Conformity

The EC Declaration of Conformity for the EMC Directive can be found on the Internet at the following address (https://support.industry.siemens.com/cs/https://support.industry.siemens.com/cs/ww/de/ps/14604/certww/en/ps/13231/cert).

Fundamental safety instructions 2

2.1 General safety instructions

⚠WARNING

Danger to life if the safety instructions and residual risks are not observed

If the safety instructions and residual risks in the associated hardware documentation are not observed, accidents involving severe injuries or death can occur.

- Observe the safety instructions given in the hardware documentation.
- Consider the residual risks for the risk evaluation.

⚠WARNING

Malfunctions of the machine as a result of incorrect or changed parameter settings

As a result of incorrect or changed parameterization, machines can malfunction, which in turn can lead to injuries or death.

- Protect the parameterization (parameter assignments) against unauthorized access.
- Handle possible malfunctions by taking suitable measures, e.g. emergency stop or emergency off.

2.2 Warranty and liability for application examples

Application examples are not binding and do not claim to be complete regarding configuration, equipment or any eventuality which may arise. Application examples do not represent specific customer solutions, but are only intended to provide support for typical tasks.

As the user you yourself are responsible for ensuring that the products described are operated correctly. Application examples do not relieve you of your responsibility for safe handling when using, installing, operating and maintaining the equipment.

2.3 Industrial security

Note

Industrial security

Siemens provides products and solutions with industrial security functions that support the secure operation of plants, systems, machines and networks.

In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.

Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.

For additional information on industrial security measures that may be implemented, please visit:

Industrial security (http://www.siemens.com/industrialsecurity)

Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.

To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at:

Industrial security (http://www.siemens.com/industrialsecurity)

Further information is provided on the Internet:

Industrial Security Configuration Manual
(https://support.industry.siemens.com/cs/ww/en/view/108862708)

⚠WARNING

Unsafe operating states resulting from software manipulation

Software manipulations (e.g. viruses, trojans, malware or worms) can cause unsafe operating states in your system that may lead to death, serious injury, and property damage.

- Keep the software up to date.
- Incorporate the automation and drive components into a holistic, state-of-the-art industrial security concept for the installation or machine.
- Make sure that you include all installed products into the holistic industrial security concept.
- Protect files stored on exchangeable storage media from malicious software by with suitable protection measures, e.g. virus scanners.
- Protect the drive against unauthorized changes by activating the "know-how protection" drive function.

General information about CMI 3

With the SINUMERIK Integrate Create MyInterface (CMI) software you can integrate CNC controllers in your production system. The communication between the CNC controllers and the host computer of the production system takes place via a defined, open communication interface.

CMI can be installed on a PCU 50/WINDOWS under SINUMERIK Operate. In exceptional cases, CMI can be installed on an IPC/WINDOWS.

Overview

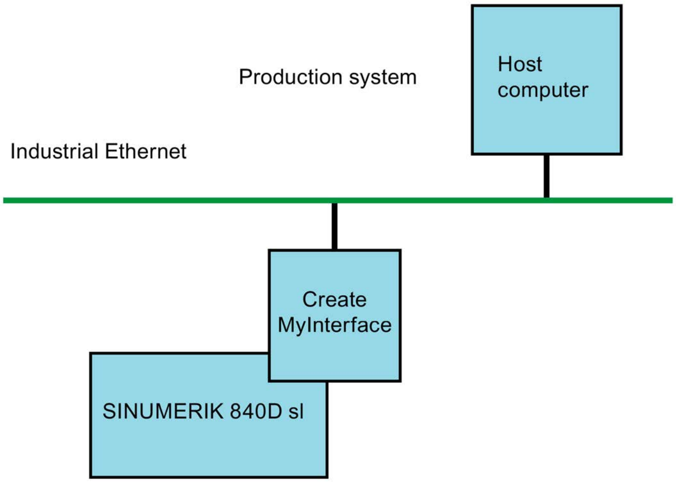
The following data is exchanged:

- CNC programs
- Configurable data from a host computer or CNC controller
- Mode switchover, synchronization
- Tool data
- Machine data, such as states, quantities
- Messages/Alarms
- Production catalog data

The active data exchange between the host computer and the CNC controllers also enables the following:

- Higher machine utilization through production planning, control and analysis
- Reduction of machine downtimes through transmission of maintenance data

This contributes to increased productivity.



Language selection

The default language setting is English.

It is not possible to switch the default language setting on an Industrial PC for operation of Create MyInterface.

System installation 4

4.1 CMI system requirements

You need the following products to operate Create MyInterface:

- SINUMERIK Integrate Create MyInterface (CMI)
- Maximum four SINUMERIK sl NCUs can be managed from an IPC.
- PCU 50.3/5 WinXP/Windows 7
- Host computer

Before installation, the following must be noted:

- The response to identical RPC calls to be processed concurrently, was changed with Create MyInterface V02.06.02.08. If a second RPC call is made that has yet to be processed while a call of the same type is still active, the second RPC call is rejected with RPC call `R_REPORT_H type = 4 errors = -200.`
- If you want to use the interface with Tool Ident Connection, you must first make sure that Tool Ident Connection has been correctly commissioned. You must also ensure that Tool Ident Connection is released for use with the relevant SINUMERIK Operate software version. Please contact your Siemens sales partner or product management if you have any queries.
- Hard real-time communication is not possible over conventional PC networks. The reaction time achievable using Create MyInterface depends on the utilization of the system and network.
- Note about operating the test program RPCSinumerikTest:

  Dialog boxes `T_DATA_M(), R_DATA_M(), C_DELETE_M,` and `C_ORDER_M()` each contain a combo box called "SFkt". In this combo box, a subfunction can be selected as a numeric value. Alternatively, the subfunction can be entered using the keyboard.

Supplementary conditions: Execution External Storage

The following limitations apply if you want to run the software Create MyInterface V02.06.02.08 on an industrial PC (IPC) while using the functionality "Execution External Storage" (EES):
Only the NC programs of the machine whose configured machine name corresponds to the default machine name can be selected in the EES memory; see example.

Excerpt from MMC.INI:

```
[Global]
NcddeMachineNames=NCU840D_1,NCU840D_2,SIM0
NcddeDefaultMachineName=NCU840D_1
```

If you want to run the software Create MyInterface V02.06.02.10 or higher on an industrial PC (IPC) together with the functionality "Execution External Storage" (EES), you can configure the EES in one of the following ways:

- Local EES option "Expanded CNC user storage" without "GDIR"

  (no CMI configuration required)
- Local EES option "Expanded CNC user storage" with "GDIR"

  (no CMI configuration required)
- Global EES option "Execution from external storage (EES)" without "GDIR"

  (CMI configuration required: Configure EES symbol name)
- Global EES option "Execution from external storage (EES)" with "GDIR"

  (no CMI configuration required)

If you are using Create MyInterface up to V02.06.02.09, you can configure the EES as follows:

- Global EES option "Execution from external storage (EES)" with "GDIR" (no CMI configuration required)

If you want to run the software Create MyInterface V02.06.02.08 on an industrial PC (IPC) together with the functionality "CF CARD", you cannot select NC programs that are stored on the CF card

Requirements for Create MyInterface Version 2.6.2.8 and higher

- The following installations must be completed beforehand:
  - SINUMERIK Operate for PCU
  - You can update Create MyInterface Version 2.6.2.1.

Supplementary conditions	Description
Create MyInterface	• CMI can manage up to four sl NCUs. If more than that number of NCUs exists, you can resort to a solution with industrial PCs connected in parallel. • Compatible with NCU-SW 2.6 / 2.7 / 4.4 / 4.5 / 4.7 and 4.8 • Installation is possible on: – A host computer – An IPC (Industrial PC) – A PCU 50 V3/V5 with SINUMERIK Operate for PCU, SW 2.6 and higher – PCU 50 with WinXP/Windows 7 is supported – SIMATIC MICROBOX PC with Windows10, and SINUMERIK Operate SW 4.8.2.1 and higher • No installation on PCU 50 with HMI-Advanced
Installation on • Host computer • Industrial PC	• The software is installed on: – WINDOWS XP SP3 (32-bit, English or German) – WINDOWS 7 SP1 (English or German, not embedded). A minimum of 4 GB RAM is required for this. At least 10 GB of disk space must be available. • SINUMERIK Operate is installed on these target systems in the background as a PC version. The corresponding system requirements also apply: • The target system must be configured for 24/7 operation (e.g. no hibernation or power saving mode).
	• When installation is performed on an Industrial PC, the performance of the Industrial PC must at least be equivalent to the current PCU 50 V3 and 2 NICs. • If a host computer of the production system or IPC is used, no other Siemens software must be installed on it. • Installation on the PC can take up to 20 min, during which time no windows will be visible for some of the time. Once the installation has been completed, a pop-up window is displayed which prompts you to reboot the system. The PC must be rebooted after installing the software. • CMI can manage up to four sl NCUs. If more than that number of NCUs exists, you can resort to a solution with industrial PCs connected in parallel. • If CMI is installed on an IPC, the IPC must be connected to the NCU via the X130 interface.

Supplementary conditions	Description
Note about Tool Ident Connection	• The interface for HMI Advanced to Tool Ident Connection available from MCIS RPC 2.0 is only supported in Create MyInterface for PCU 50 with SW 2.7 SP3 HF3, SW 4.4 SP3, SW 4.5 SP2, SW 4.5 SP3/SP4, SW 4.7 SP2HF1, SW 4.7 SP3 and SW 4.8 . • For installation on an Industrial PC, this interface to Tool Ident Connection is available for NCU SW 2.6 / 2.7 / 4.4/4.5/4.7, and 4.8. If the Ident Connection interface is used on an Industrial PC, one and only one NCU with Tool Ident Connection may be operated on the Industrial PC. The three other NCUs can then only be operated without Tool Ident Connection. • The operator interfaces familiar from MCIS RPC 2.0 for HMI Advanced are only partially available for the PCU 50.

Additional supplementary conditions

- The operation of MCIS RPC or Create MyInterface in a virtual machine or in a view-in-view configuration is not approved.
- The function `R_DATA_M` function number 26 is not available for Create MyInterface, because a complete CMI user interface is no longer available on the machine.

  It is only available for Tool Ident Connection and Create MyInterface on command code 4 (Load tool from host computer).
- The functions `R_REPORT_M` type 3 and `R_MESSAGE_M` are not available for Create MyInterface. The alarms are only triggered in the alarm server on the host computer and are not visible on the machine.
- For each configured machine, there is a SCVARSET.INI file in which the variable sets can be defined. This also applies to the alarm filter SCALARMEX.INI. The alarm filter can be created for every configured machine.
- The "Connection" parameter is omitted in SCVARSET.INI.
- NC programs that are being unloaded from the NCU or that are pending for loading into the NCU are saved in a directory structure above the BIN directory.

  ```
                     WKS.DIR\
  ..\<Maschinenname>\MPF.DIR\ <NC-PROGRAMM>
                     SPF.DIR\
  ```
- To prevent the time stamps between incoming and outgoing alarms from diverging, the "C_ORDER_M SFKT=100" command for time synchronization between the PCU and NCU must be called at regular intervals.
- The installation notes in "documentation_de.pdf" must be observed.
- For installation of CMI GUI on SINUMERIK Operate, WinSCP 5.5.1 must be installed on the PC, see Software link (https://sourceforge.net/projects/winscp/files/WinSCP/).
- Note: The Put and Get directories in the machine configuration are defined and cannot be changed.
- The configuration tool must be started with administrator rights.
- For Tracelevel 4, the required disk space must be set with the configuration tool.
- Special supplementary conditions apply to the support of EES, see section: Function EES (Execution External Storage) for CMI (Page 251).
- Special supplementary conditions apply to the support of CF cards; see section Function CF card (CompactFlash card) for CMI (Page 264).

4.2 Installing CMI

The following work steps are necessary for installation of CMI:

- Installation of the CMI software on a SINUMERIK controller

  Detailed information on the installation procedure can found in the following sections:

  - Installing CMI on a PCU 50 (Page 24)
  - Installing CMI on an Industrial PC (Page 26)
  - Installing CMI on the host computer (Page 29)
- For configuration of the INI.file of the program, see section: Configuring SINUMERIK Operate on the IPC or FLR with the "MMC.INI" file (Page 44)
- Set up of the DB12 on the PLC for the interface. You will find information on the structure of the data block in the following section: Interface and machine PLC (Page 197).

Requirement

You need ADMIN rights to install Create MyInterface.

Note

ADMIN rights

If you do not have ADMIN rights, you cannot install Create myInterface.

The following message is displayed: "No Admin-rights! Setup for Create MyInterface will not be installed".

- Log on as a user with ADMIN rights.

Note the following before the installation:

- If you want to use the interface with Tool Ident Connection, you must first make sure that Tool Ident Connection has been correctly commissioned.
- Ensure that use of Tool Ident Connection is enabled on the respective version of the operating software of SINUMERIK Operate.
- Hard real-time communication is not possible over conventional PC networks. The reaction time achievable using Create MyInterface depends on the utilization of the system and network.
- Note on operating the test program RPC SINUMERIK Test :

  Dialog boxes `T_DATA_M()`, `R_DATA_M()`, `C_DELETE_M`, and `C_ORDER_M()` each include the drop-down list box "SFkt". A subfunction can be selected as a numeric value in this drop-down list box. Alternatively, the subfunction can be entered via the keyboard.
- You can find other important information in section: CMI system requirements (Page 19).

Detecting installed versions

The installation is described using hotfix 8 as example. However, always first refer to the readme file with the current installation instructions.

- Open file "version.txt":

  <Root-Sinumerik-Operate>\addon\sinumerik\hmi\appl\version.txt

- The entry "Version" in section [MCIS_1] identifies the installed version.

  The entry must be "02.06.02.08".

Uninstalling CMI

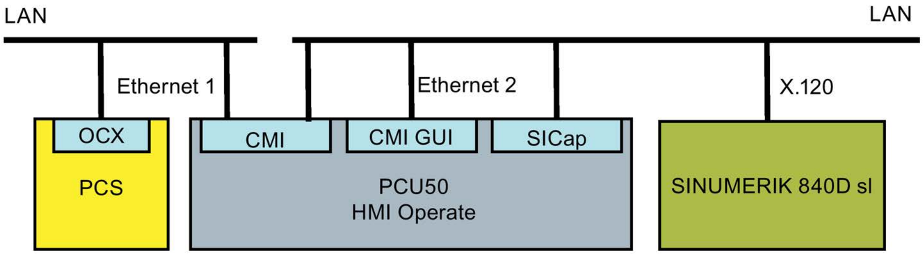
Hotfix CMI 2.6.2.8 cannot be uninstalled separately.

CMI can only be uninstalled in its entirety.

4.2.1 Installing CMI on a PCU 50

This section describes the installation of CMI on a PCU 50. The installation is described using hotfix 8 as example. However, always first refer to the readme file with the current installation instructions.

Installation environment



Requirements

- You need ADMIN rights to perform the installation.
- Read the information in the following sections for the installation:

  CMI system requirements (Page 19)

  Installing CMI (Page 23)

Log files for diagnostics

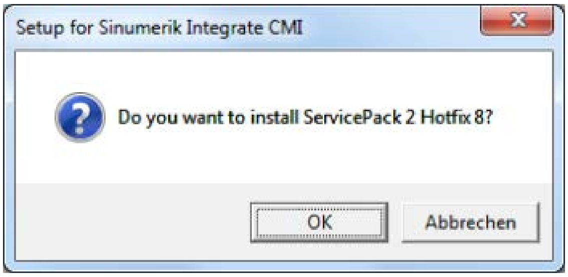
A log file is written during installation:

- If the Siemens hotline needs this file, you will find it here:
  <hmislpath>\addon\sinumerik\hmi\appl\<datum>_setup_cmi_sp2HF8.log
- If the file is not to be found in this directory, you must provide the following file:
  <WINDIR>\Installer\installCMI\install.log

Procedure

1. Copy the content of the CD to any temporary folder on your Industrial PC.
2. Double-click "Setup_CMI_SP2HF8.exe" file to start it.
3. The "Setup for SINUMERIK Integrate CMI" window opens.

   Click the "OK" button to start the installation.

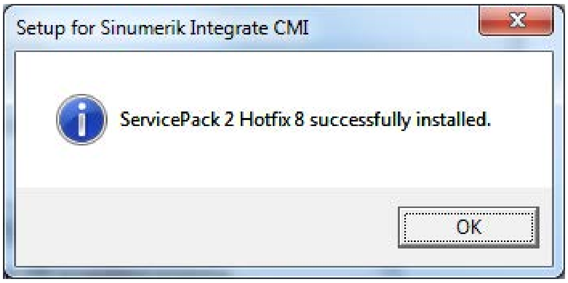


Setup for Sinumerik Integrate CMI

Do you want to install ServicePack 2 Hotfix 8?

4. After successful installation, the window opens again and you receive a success message.

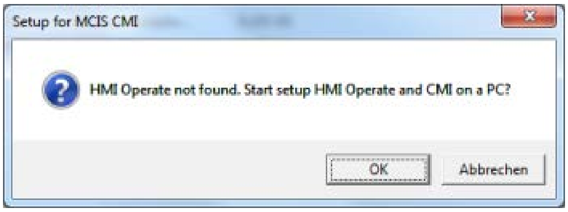
   Click the "OK" button to complete the installation.



Setup for Sinumerik Integrate CMI

ServicePack 2 Hotfix 8 successfully installed.

- OR -
If operating software is not yet installed, you receive the following message:



Setup for MCIS CMI

HMI Operate not found. Start setup HMI Operate and CMI on a PC?

5. The following menu commands are newly created in the Windows menu: "Start" > "Programs" > "SINUMERIK" > "Create MyInterface" > "Create MyInterface Configuration".
6. Now configure Create MyInterface, see section: Configuring CMI (Page 32)

Note

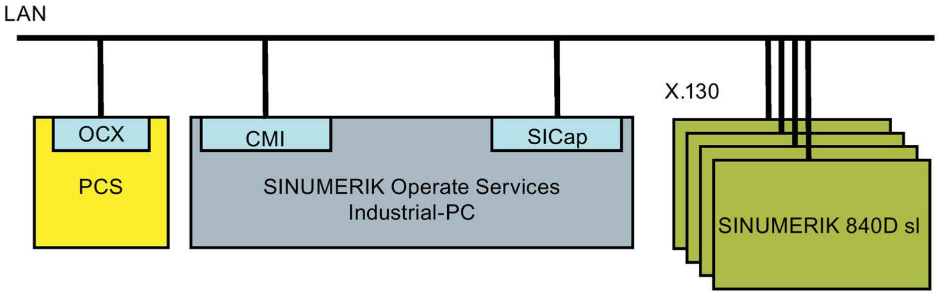
Screenshots

The screenshots show the installation based the example with SP2 HF8. The displayed version number will change for newer versions.

4.2.2 Installing CMI on an Industrial PC

This section describes the installation of CMI on an Industrial PC (IPC).

Installation environment



Requirements

- Installation and the system start thereafter must be executed under the SINUMERIK administrator "AUDUSER" with that user's standard password and administrator authorization.
- CMI and the "MMC.INI" file are configured, see section: Configuring SINUMERIK Operate on the IPC or FLR with the "MMC.INI" file (Page 44)
- Read the information in the following sections for the installation:

  CMI system requirements (Page 19)

  Installing CMI (Page 23)

Procedure for CMI 2.6 SP2 HF1

This CMI version installs operating software 4.5 SP3 without operator interface.

Note

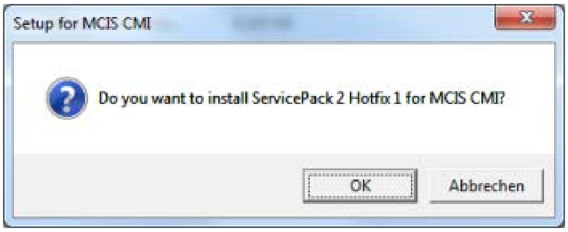
Execution on External Storage (EES)

This version does not support Execution on External Storage (EES)!

If you need this option, install CMI Version 2.6 SP2 HF8, see below, section: "Procedure for CMI 2.6 SP2 HF8"

1. Copy the content of the CD to any temporary folder on your Industrial PC.
2. Double-click the "Setup_CMI_SP2HF1.exe" file to start it.
3. The "Setup for MCIS CMI" window opens.

   Click the "OK" button to start the installation.

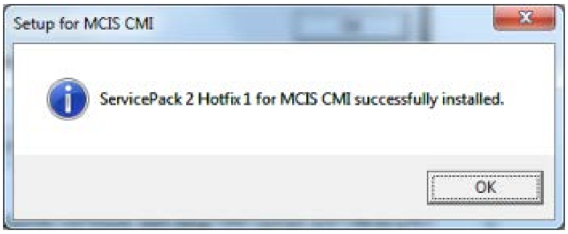


Setup for MCIS CMI

Do you want to install ServicePack 2 Hotfix 1 for MCIS CMI?

4. After successful installation, the window opens again and you receive a success message.

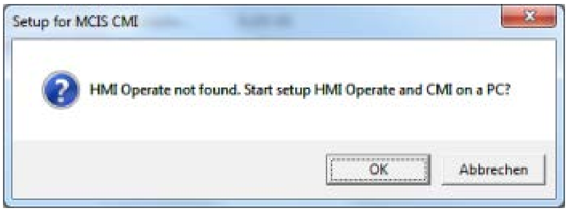
   Click the "OK" button to complete the installation.



Setup for MCIS CMI

ServicePack 2 Hotfix 1 for MCIS CMI successfully installed.

- OR -
If operating software is not yet installed, you receive the following message:



Setup for MCIS CMI

HMI Operate not found. Start setup HMI Operate and CMI on a PC?

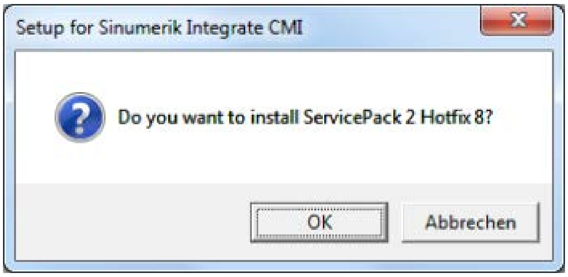
5. The following menu commands are newly created in the Windows menu: "Start" > "Programs" > "SINUMERIK" > "Create MyInterface" > "Create MyInterface Configuration".
6. Now configure Create MyInterface, see section: Configuring CMI (Page 32).

Procedure for CMI 2.6 SP2 HF8

This CMI version installs operating software 4.7 SP4 HF4 without operator interface.

1. Copy the content of the CD to any temporary folder on your Industrial PC.
2. Double-click "Setup_CMI_SP2HF8.exe" file to start it.
3. The "Setup for SINUMERIK Integrate CMI" window opens.

   Click the "OK" button to start the installation.

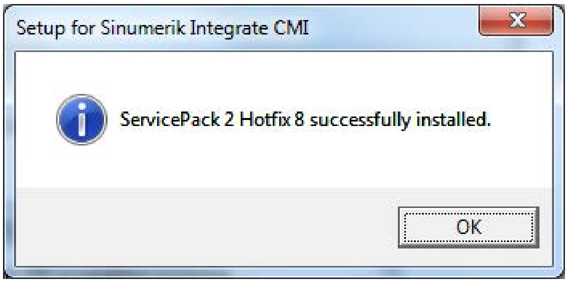
   

   Setup for Sinumerik Integrate CMI

   Do you want to install ServicePack 2 Hotfix 8?

4. After successful installation, the window opens again and you receive a success message.

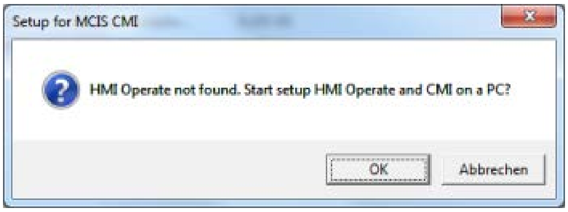
   Click the "OK" button to complete the installation.

   Setup for Sinumerik Integrate CMI

   ServicePack 2 Hotfix 8 successfully installed.

   - OR -
   If operating software is not yet installed, you receive the following message:

   Setup for MCIS CMI

   HMI Operate not found. Start setup HMI Operate and CMI on a PC?

5. The following menu commands are newly created in the Windows menu: "Start" > "Programs" > "SINUMERIK" > "Create MyInterface" > "Create MyInterface Configuration".
6. Now configure Create MyInterface, see section: Configuring CMI (Page 32).

4.2.3 Installing CMI on the host computer

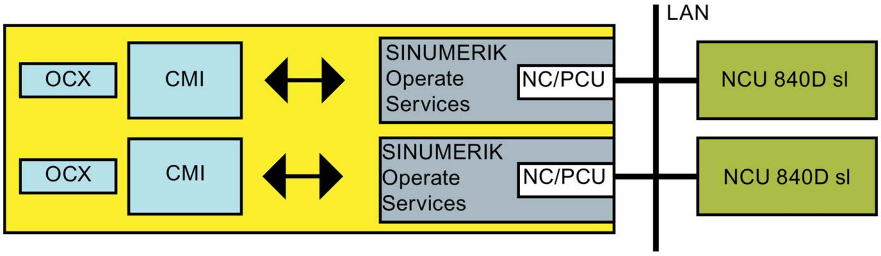
This section describes the installation of CMI on a host computer.

Note

Exception

This section is not relevant if you only want to install the "RPC.OCX" for use in your host computer software, see section: Configuring CMI without DB12 (Page 48).

Installation environment



Requirements

- You need ADMIN rights to perform the installation.
- Read the information in the following sections for the installation:

  CMI system requirements (Page 19)

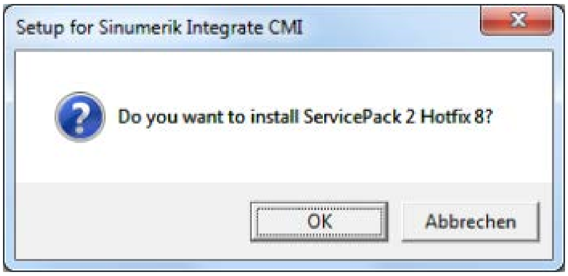
  Installing CMI (Page 23)

Procedure for CMI 2.6 SP2 HF8

This CMI version installs operating software 4.7 SP4 HF4 without operator interface.

1. Copy the content of the CD to any temporary folder on your Industrial PC.
2. Double-click "Setup_CMI_SP2HF8.exe" file to start it.
3. The "Setup for SINUMERIK Integrate CMI" window opens.

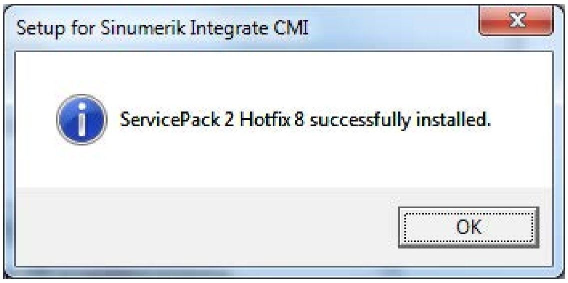
   Click the "OK" button to start the installation.


Setup for Sinumerik Integrate CMI

Do you want to install ServicePack 2 Hotfix 8?

4. After successful installation, the window opens again and you receive a success message.

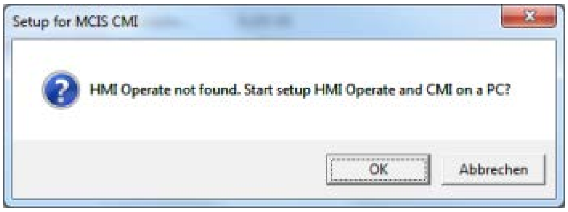
   Click the "OK" button to complete the installation.

   Setup for Sinumerik Integrate CMI

   ServicePack 2 Hotfix 8 successfully installed.

   - OR -
   If operating software is not yet installed, you receive the following message:

   Setup for MCIS CMI

   HMI Operate not found. Start setup HMI Operate and CMI on a PC?

5. The following menu commands are newly created in the Windows menu: "Start" > "Programs" > "SINUMERIK" > "Create MyInterface" > "Create MyInterface Configuration".

6. Now configure Create MyInterface, see section: Configuring CMI (Page 32).

4.2.4 Starting/exiting CMI

Create MyInterface is integrated in SINUMERIK Operate of the PCU 50 or Industrial PC.

- Start CMI:

  When you start SINUMERIK Operate, Create MyInterface also starts simultaneously.

- Exit CMI:

  When you exit SINUMERIK Operate using the "Exit" softkey, Create MyInterface is also exited.

4.2.5 Firewall setting

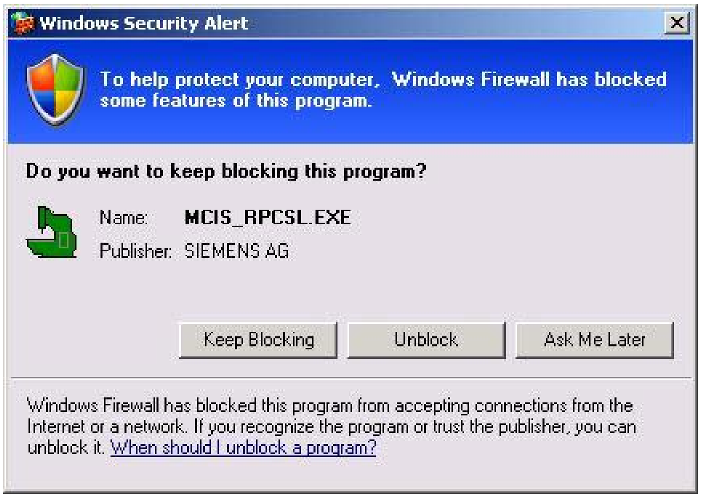
When you start the PCU or the IPC with CMI after a successful initial installation, you receive a Windows security alert.

Set the settings to "Unblock" for the RPC component.

Procedure

The "Windows Security Alert" is displayed

- Click the "Unblock" button to unblock the settings of the RPC components.



Windows Security Alert

To help protect your computer, Windows Firewall has blocked some features of this program.

Do you want to keep blocking this program?

Name: MCIS_RPCSL.EXE
Publisher: SIEMENS AG

Windows Firewall has blocked this program from accepting connections from the Internet or a network. If you recognize the program or trust the publisher, you can unblock it. When should I unblock a program?

4.3 Configuring CMI

4.3.1 Configuring CMI with Create MyInterface Configuration

This section describes the configuration of CMI after installation on a PCU 50.

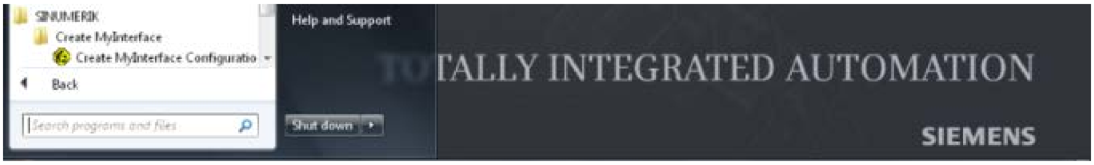
Requirement

CMI has been successfully installed on a PCU 50

Procedure

1. Start the configuration program "MCIS_RPCSLCFG.exe".

   You can configure a maximum of one machine.
2. Open "Create MyInterface Configuration" in the Windows menu.
   Select "Start" > "Programs" > "SINUMERIK" > "Create MyInterface" > "Create MyInterface Configuration".


SINUMERIK
Create MyInterface
Create MyInterface Configuratio
Back
Search programs and files
Help and Support
Shut down
TOTALLY INTEGRATED AUTOMATION
SIEMENS

3. Start the configuration by opening the individual tabs.

   Follow the instructions in the following sections.

4.3.2 "Machine" tab

You make the settings relevant for the CMI communication on the "Machine" tab.

Note

Names of the "Put" and "Get" directories

Starting from CMI V2.06.02.01, the configuration of the "Put" and "Get" directories for the machine is omitted. The names of the "Put" and "Get" directories are automatically generated and may not be changed.

The table below provides information about the individual parameters.

Parameters

Table 4- 1 "Machine" tab

Parameter	Description
Name	Symbolic machine name
NCU Name	Machine name from configuration of SINUMERIK Operate (MMC.INI)
Endpoint	Port number of the TCP/IP communication
IP-GuiServer	*, or IP address PCU 50 X.120 (plant network) for the GUI communication
Put-Directory	Temporary directory for data transfer to the host computer. **Note:** The directory is automatically created and may not be changed.
Get-Directory	Temporary directory for data transfer from the host computer. **Note:** The directory is automatically created and may not be changed.
Enable interface to Tool Ident Connection	Select this option if the Tool Ident Connection command codes 4 and 5 (Load/unload via host computer) are required.
no DB12	You can select this option if the PLC of the SINUMERIK 840D/810D/840Dsl does not have a data block (DB12). **Note:** In this case, the PLC functionality cannot be used.
Force Execution External Storage (EES)	You can select this option if the NC program functionalities are to be carried out in the EES memory and not in the NC memory. This option must not be set if the FLR already uses the new Sfkt for EES; see section: Function calls of EES and CF card (Page 251). **Note:** Cannot be combined with Force Compact Flash Card
EES symbol name	Symbolic device name for EES access, see section: CMI system requirements (Page 19); section Supplementary conditions Execution External Storage.

Parameter	Description
Force Compact Flash card	You can select this option if the NC program functionalities are to be carried out on the CF card and not in the NC memory. **Note:** Cannot be combined with Force Execution External Storage (EES).
Number	Not used for installation on PCU 50 You can connect up to four controllers. You toggle between the configuration data with the "<" and ">" buttons.

Note

Activation

Force Execution External Storage and Force Compact Flash Card are mutually exclusive. Only one function can be active at any one time. The two switches must be mutually interlocked.

Force Execution External Storage

The function "Force Execution External Storage" is used to execute all RPC calls received from the host computer for transferring, returning, loading, unloading, selecting, deselecting, and listing NC programs not in the NC memory of the SINUMERIK but in the EES memory of the SINUMERIK, where the action is not noticed by the host computer.
This option must not be set if the FLR already uses the new Sfkt for EES; see section: Function calls of EES and CF card (Page 251).

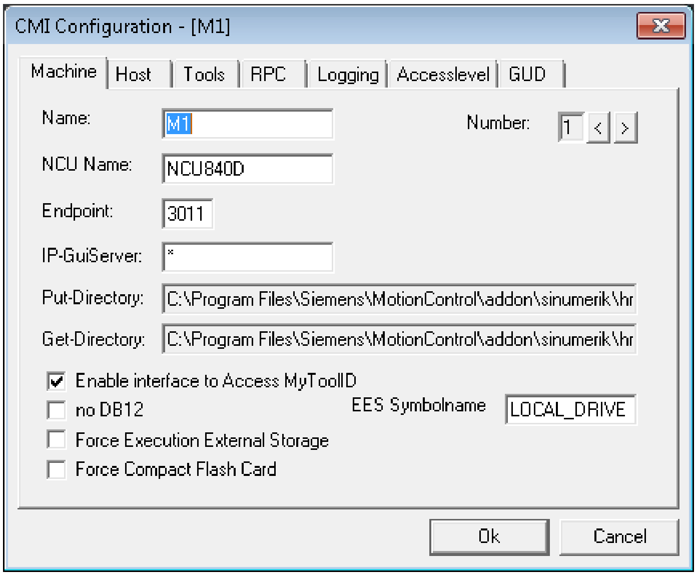
Force Compact Flash card

The function "Force Compact Flash Card" is used to execute all RPC calls received from the host computer for transferring, returning, loading, unloading, deselecting, and listing NC programs not in the NC memory of the SINUMERIK but on the CF card of the SINUMERIK, where the action is not noticed by the host computer.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Select the "Machine" tab.
2. Make your settings.
3. Click the "OK" button to save the settings.



CMI Configuration - [M1]

Name: M1

Number: 1

NCU Name: NCU840D

Endpoint: 3011

IP-GuiServer: *

Put-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr

Get-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr

☑ Enable interface to Access MyToolID

□ no DB12

EES Symbolname LOCAL_DRIVE

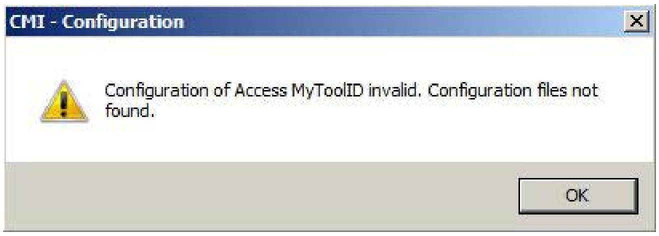
□ Force Execution External Storage

□ Force Compact Flash Card

Note

Check

Whenever the interface to Tool Ident Connection is activated, a check is made for the existence of the configuration files of Tool Ident Connection. A check is made for the existence of the files 'tdiidentcfg.xml', 'toolspec.xml', und 'wkonvert.mcx'. If one of the files is missing, activation of the interface to Tool Ident Connection is prevented and a fault message is output.



CMI - Configuration

Configuration of Access MyToolID invalid. Configuration files not found.

4.3.3 "Host" tab

You make the settings relevant for the CMI communication with the host computer on the "Host" tab. You can configure a maximum of eight hosts per machine.

The table below provides information about the individual parameters.

Parameters

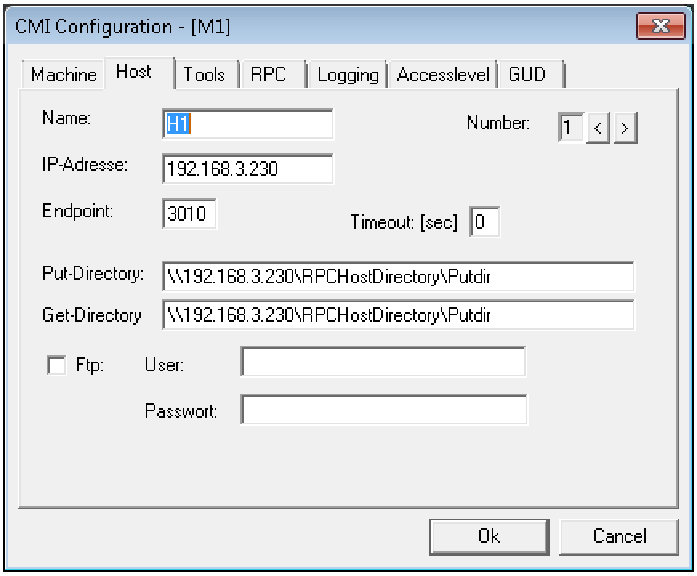
Table 4- 2 "Host" tab

Parameter	Description
Name	Symbolic name of the host computer
IP address	IP address of the host computer (no computer name)
Endpoint	Port number of the TCP/IP communication
Put-Directory	Directory into which Create MyInterface transfers files to the host computer. **Note:** This directory must **not** be identical to the machine directory.
Get-Directory	Directory in which Create MyInterface fetches files from the host computer. **Note:** This directory must **not** be identical to the machine directory.
Number	You toggle between the configuration data with the "<" and ">" buttons.
Ftp	If the host computer is a Windows computer (NT or Win95), FTP does not have to be used. The "User" and "Password" input fields are then not relevant because the data can be transmitted via SMB (Microsoft notation). For other host computers, e.g. UNIX or Linux computers, data must be transferred using FTP if, for example, Samba is not installed on the host computers, which in turn allows Microsoft notation. The "Ftp" option check box must then be selected and valid entries must be made in "User" and "Password" for logging onto the host computer. The specified user must have write access to the directories specified in Put directory and read access to the directories specified in Get directory.
User	
Password	
Timeout [sec]	The timing is influenced using this value.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Select the "Host" tab.
2. Make your settings.
3. Click the "OK" button to save the settings.



CMI Configuration - [M1]

Name: H1

Number: 1

IP-Adresse: 192.168.3.230

Endpoint: 3010

Timeout: [sec] 0

Put-Directory: \\192.168.3.230\RPCHostDirectory\Putdir

Get-Directory \\192.168.3.230\RPCHostDirectory\Putdir

□ Ftp:

User:

Passwort:

4.3.4 "Tools" tab

You make the necessary settings for the reading of tool data on the "Tools" tab. The settings pertain to the machine indicated in the title bar.

The table below provides information about the individual parameters.

Parameters

The reading of tool data of a tool from the NC requires one read call per data area.

Note

Reporting of tool data

It is not always necessary to report all of the tool data to the host computer.

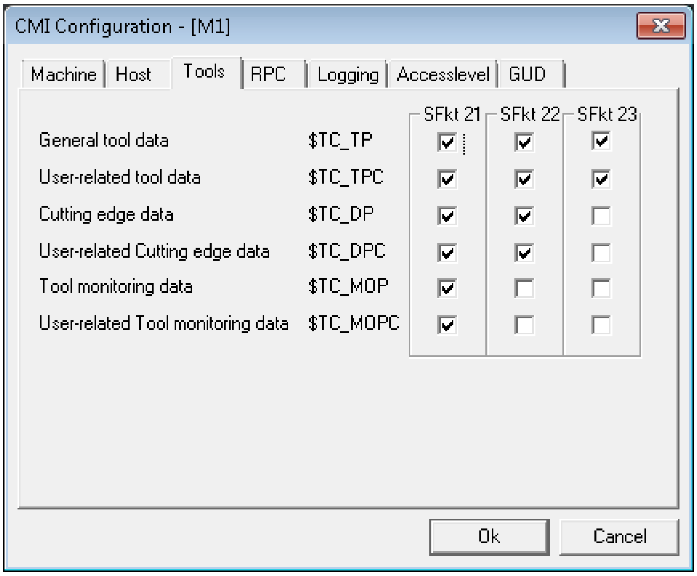
For this reason you can define three tool structures on the "Tools" tab (SFkt21 - SFkt23):

Define tool structures SFkt21 - SFkt23 depending on the needs of the specific project.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Select the "Tools" tab.
2. Select the check boxes of the desired tool data.
3. Click the "OK" button to save the settings.



		SFkt 21	SFkt 22	SFkt 23
General tool data	$TC_TP	☑	☑	☑
User-related tool data	$TC_TPC	☑	☑	☑
Cutting edge data	$TC_DP	☑	☑	□
User-related Cutting edge data	$TC_DPC	☑	☑	□
Tool monitoring data	$TC_MOP	☑	□	□
User-related Tool monitoring data	$TC_MOPC	☑	□	□

4.3.5 "RPC" tab

The host computer to which an RPC function is to be sent and the RPC function to be received from a given host computer must be specified for each RPC function.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Select the "RPC" tab.

2. Make your settings as follows:

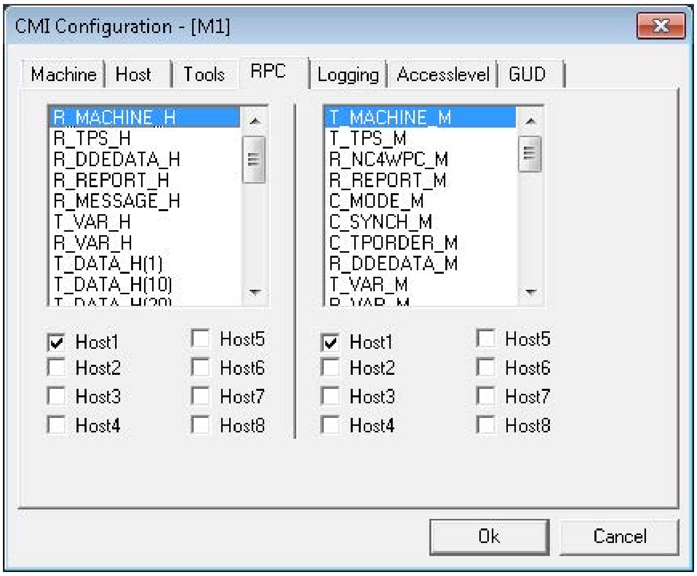
   – Left list

     Here you specify which host computer each individual RP will be sent to for the machine tool indicated in the title bar.

   – Right list

     Here you specify the RPC function to be received from a given host computer.

3. Click the "OK" button to save the settings.



R_MACHINE_H
R_TPS_H
R_DDEDATA_H
R_REPORT_H
R_MESSAGE_H
T_VAR_H
R_VAR_H
T_DATA_H(1)
T_DATA_H(10)

☑ Host1 □ Host5
□ Host2 □ Host6
□ Host3 □ Host7
□ Host4 □ Host8

T_MACHINE_M
T_TPS_M
R_NC4WPC_M
R_REPORT_M
C_MODE_M
C_SYNCH_M
C_TPORDER_M
R_DDEDATA_M
T_VAR_M

☑ Host1 □ Host5
□ Host2 □ Host6
□ Host3 □ Host7
□ Host4 □ Host8

4.3.6 "Logging" tab

You specify the size of the log file and the Tracelevel of the log function on the "Logging" tab. The settings pertain to the machine indicated in the title bar.

Note

Tracelevel numbering

- "Tracelevel 1" is preset in the delivery state.
- All other Tracelevels are used for diagnostics and may only be used in consultation with the SIEMENS Hotline. These Tracelevels are not permitted for continuous operation.

Height of the Tracelevel

- The higher the Tracelevel, the more extensive the recorded data will be.
- Tracelevel 4 is a special case and is used only for debugging. Additional files are written.

  The disk space used can be specified. Ensure that there is sufficient space on the hard disk.

The table below provides information about the individual parameters.

Parameters

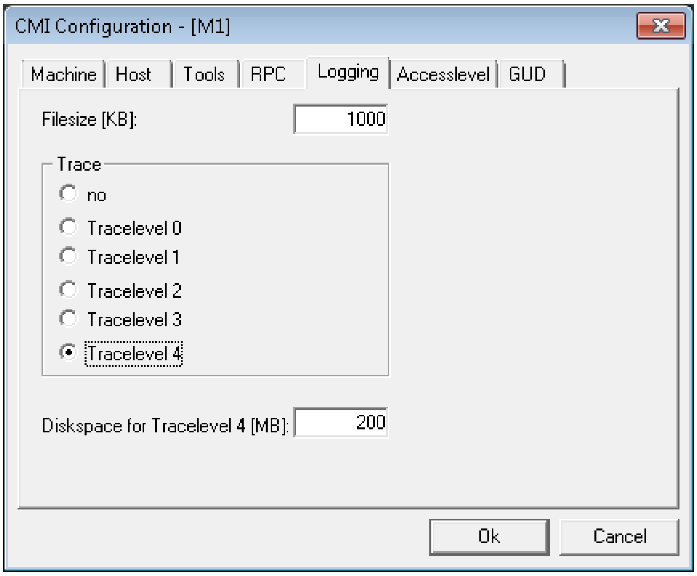
Table 4- 3 "Logging" tab

Parameter	Description
Filesize [KB]	
Trace	Choice of Tracelevel of the log function The following can be selected: • no (= no Tracelevel) • Tracelevel 0 to Tracelevel 4
Disk space for Tracelevel 4 [MB]	Specification of hard disk memory space specifically for Tracelevel 4 Preset with a value as of version CMI V02.06 SP2 HF8.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Select the "Logging" tab.
2. Make your settings.
3. Click the "OK" button to save the settings.



CMI Configuration - [M1]

Filesize [KB]: 1000

Trace
○ no
○ Tracelevel 0
○ Tracelevel 1
○ Tracelevel 2
○ Tracelevel 3
● Tracelevel 4

Diskspace for Tracelevel 4 [MB]: 200

4.3.7 "Accesslevel" tab

On the "Accesslevel" tab, you enter the altered machine passwords that deviate from the Siemens default for the machine and allow access to certain access levels of the NC.

The settings pertain to the machine indicated in the title bar.

The table below provides information about the individual parameters.

Parameters

Table 4- 4 "Accesslevel" tab

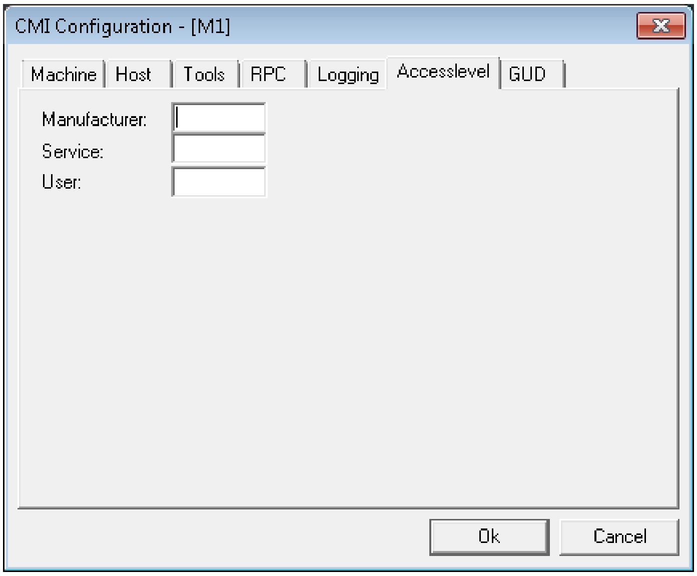
Parameter	Description
Manufacturer	Input of the password for access by the manufacturer.
Service	Input of the password for access by service personnel.
User	Input of the password for access by users.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Open the "Accesslevel" tab.

2. Enter the passwords.
3. Click the "OK" button to save the passwords.



CMI Configuration - [M1]

Manufacturer:
Service:
User:

4.3.8 "GUD" tab

You select global user data for NC-specific tasks on the "GUD" tab. The settings pertain to the machine indicated in the title bar.

Only select the GUD variables that are also used by CMI.

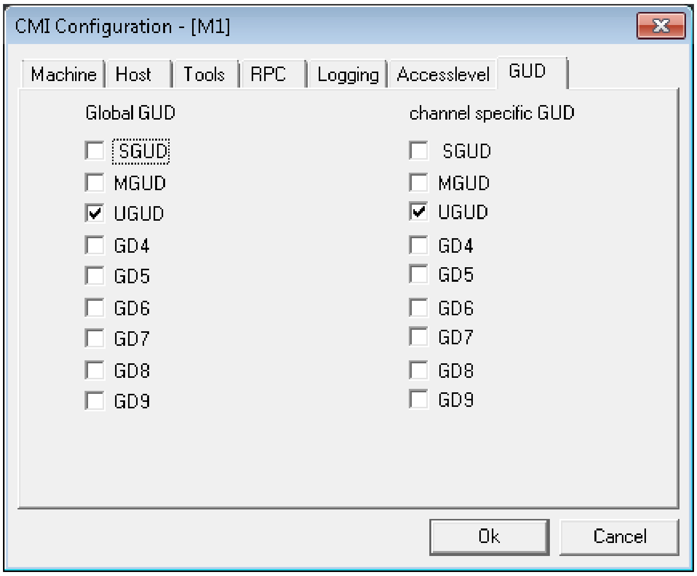
You will find further information on GUD variables in the following sections: Explanation of GUD variables (Page 225), Access to global user variables GUD, SGUD, MGUD, UGUD, GD3 ... GD9 (Page 339).

The table below provides information about the individual parameters.

Procedure

1. The "CMI Configuration - [M1]" window is open.

   Open the "GUD" tab.
2. Select the check boxes of the desired GUDs.
3. Click the "OK" button to save the settings.



Global GUD	channel specific GUD
□ SGUD	□ SGUD
□ MGUD	□ MGUD
☑ UGUD	☑ UGUD
□ GD4	□ GD4
□ GD5	□ GD5
□ GD6	□ GD6
□ GD7	□ GD7
□ GD8	□ GD8
□ GD9	□ GD9

4.3.9 Configuring SINUMERIK Operate on the IPC or FLR with the "MMC.INI" file

You configure controllers with the SINUMERIK Operate operating software with the "MMC.INI" file.

You can configure a maximum of four controllers.

Note

Activation of ports

Create MyInterface must access the NCU of the SINUMERIK 840D sl via the X130 interface.

For this purpose, the following ports must be activated in the firewall of the SINUMERIK 840D sl:

- Port 5900
- Port 102
- Port 22

The ports are activated in file "BASESYS.INI".

Extract from the "BASESYS.INI" file:

```
; Protocol/Port pairs to open in the firewall (e.g., TCP/5900, UDP/514, ...)
FirewallOpenPorts="TCP/5900 TCP/102 TCP/22"
```

Example: Configuring "NCU840D" controller

Windows XP operating system:

(extract from C:\Program Files\Siemens\Sinumerik\hmisl\user\sinumerik\hmi\cfg\mmc.ini)

- OR -

Windows 7 operating system:

(extract from C:\Program Files (x86)\Siemens\MotionControl\user\sinumerik\hmi\cfg\mmc.ini)

```
[Global]
NcddeMachineNames=NCU840D,SIM0
NcddeDefaultMachineName=NCU840D
;===========================================================================
; SlCap Communication from Windows to NC, PLC, DRIVES und SIMOTION
;===========================================================================
[NCU840D]
ADDRESS0=195.208.2.208,LINE=14,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4_INT
ADDRESS1=195.208.2.208,LINE=14,NAME=/PLC,SAP=0202,PROFILE=CLT1__CP_L4_INT
ADDRESS2=195.208.2.208,LINE=14,NAME=/CP,SAP=0502,PROFILE=CLT1__CP_L4_INT
ADDRESS10=195.208.2.208,LINE=14,NAME=/DRIVE_00_000,SAP=0201,SUBNET=0000-
00000000:000,PROFILE=CLT1__CP_L4_INT
ADDRESS11=195.208.2.208,LINE=14,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT1__CP_L4_INT
ADDRESS12=
ADDRESS13=
ADDRESS14=
ADDRESS15=
ADDRESS16=
ADDRESS17=
```

Example: Configuring two controllers: "NCU840D_1" and "NCU840D_2"

Windows XP operating system:

(extract from C:\Program Files\Siemens\Sinumerik\hmisl\user\sinumerik\hmi\cfg\mmc.ini)

- OR -

Windows 7 operating system:

(extract from C:\Program Files (x86)\Siemens\MotionControl\user\sinumerik\hmi\cfg\mmc.ini)

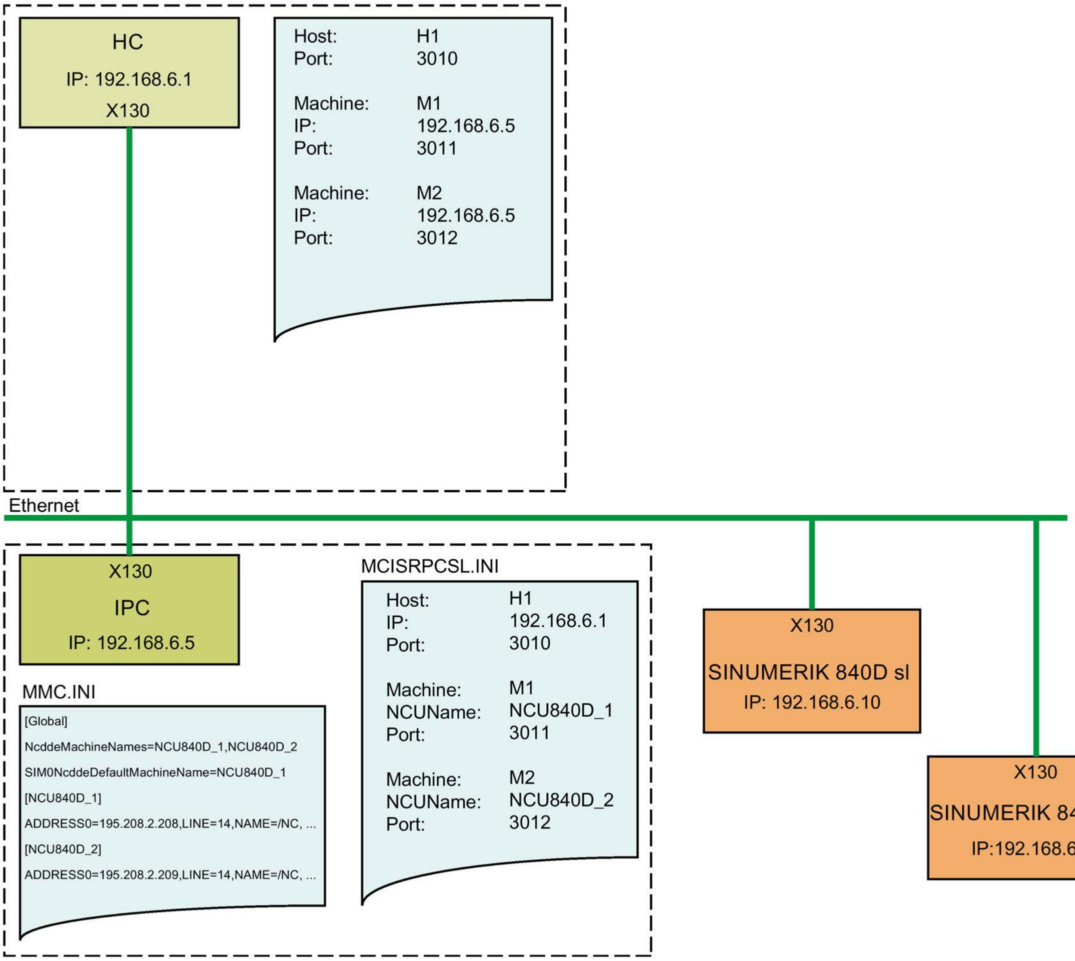
```
[Global]
NcddeMachineNames=NCU840D_1,NCU840D_2,SIM0
NcddeDefaultMachineName= NCU840D_1
;============================================================================
; SlCap Communication from Windows to NC, PLC, DRIVES und SIMOTION
;============================================================================
NCU840D_1]
ADDRESS0=195.208.2.208,LINE=14,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4_INT
ADDRESS1=195.208.2.208,LINE=14,NAME=/PLC,SAP=0202,PROFILE=CLT1__CP_L4_INT
ADDRESS2=195.208.2.208,LINE=14,NAME=/CP,SAP=0502,PROFILE=CLT1__CP_L4_INT
ADDRESS10=195.208.2.208,LINE=14,NAME=/DRIVE_00_000,SAP=0201,SUBNET=0000-
00000000:000,PROFILE=CLT1__CP_L4_INT
ADDRESS11=195.208.2.208,LINE=14,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT1__CP_L4_INT
ADDRESS12=
ADDRESS13=
ADDRESS14=
ADDRESS15=
ADDRESS16=
ADDRESS17=

[NCU840D_2]
ADDRESS0=195.208.2.209,LINE=14,NAME=/NC,SAP=040d,PROFILE=CLT1__CP_L4_INT
ADDRESS1=195.208.2.209,LINE=14,NAME=/PLC,SAP=0202,PROFILE=CLT1__CP_L4_INT
ADDRESS2=195.208.2.209,LINE=14,NAME=/CP,SAP=0502,PROFILE=CLT1__CP_L4_INT
ADDRESS10=195.208.2.209,LINE=14,NAME=/DRIVE_00_000,SAP=0201,SUBNET=0000-
00000000:000,PROFILE=CLT1__CP_L4_INT
ADDRESS11=195.208.2.209,LINE=14,NAME=/DRIVE_03_003,SAP=0900,PROFILE=CLT1__CP_L4_INT
ADDRESS12=
ADDRESS13=
ADDRESS14=
ADDRESS15=
ADDRESS16=
ADDRESS17=
```

4.3.10 Example: Host computer with host application and IPC

Below are two options for configuring the host computer with host application:

- Host computer with host application and IPC for two SINUMERIK controllers
- Host computer with two host applications and IPC for two SINUMERIK controllers

Host computer with host application and IPC for two SINUMERIK controllers



```
Host:       H1
Port:       3010

Machine:    M1
IP:         192.168.6.5
Port:       3011

Machine:    M2
IP:         192.168.6.5
Port:       3012
```

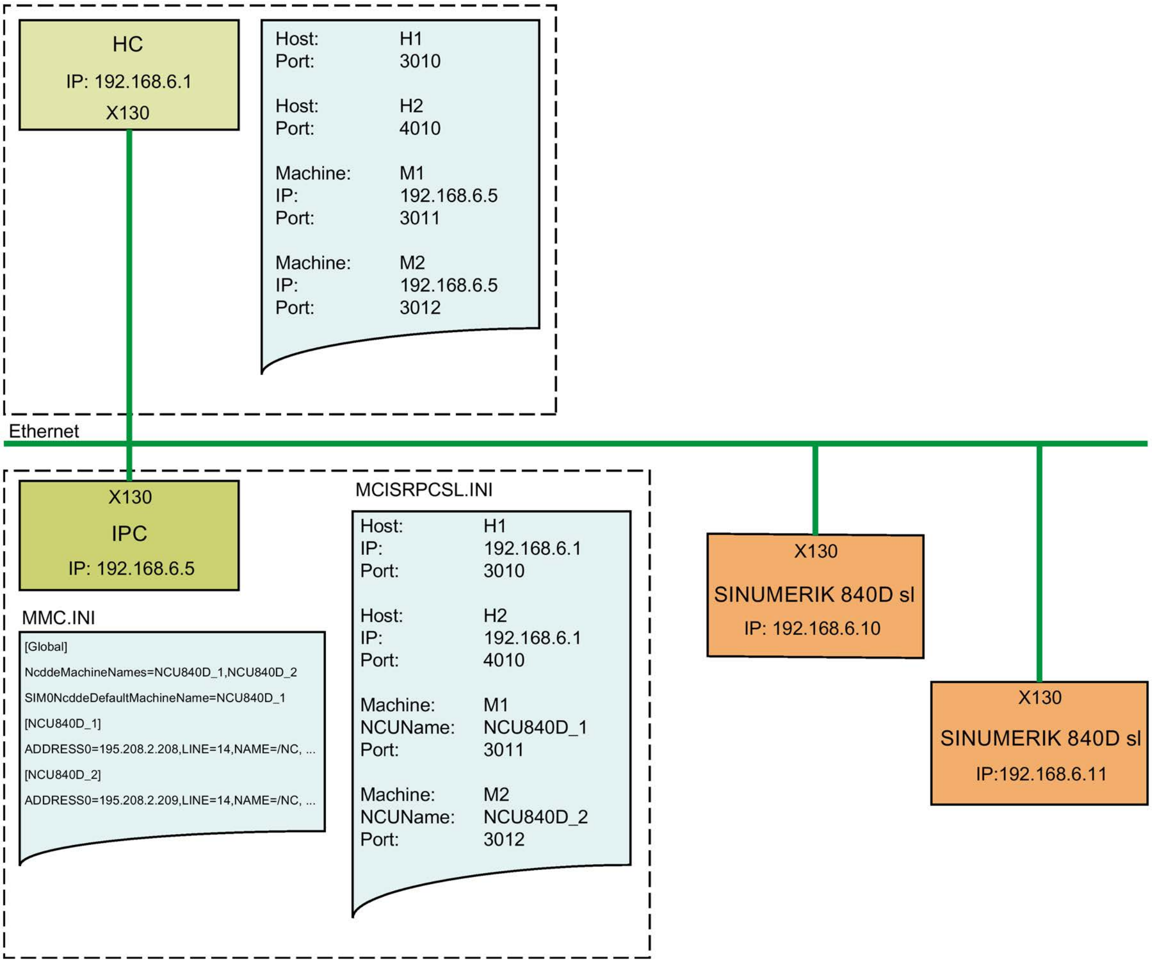
MMC.INI

```
[Global]
NcddeMachineNames=NCU840D_1,NCU840D_2
SIM0NcddeDefaultMachineName=NCU840D_1
[NCU840D_1]
ADDRESS0=195.208.2.208,LINE=14,NAME=/NC, ...
[NCU840D_2]
ADDRESS0=195.208.2.209,LINE=14,NAME=/NC, ...
```

MCISRPCSL.INI

```
Host:       H1
IP:         192.168.6.1
Port:       3010

Machine:    M1
NCUName:    NCU840D_1
Port:       3011

Machine:    M2
NCUName:    NCU840D_2
Port:       3012
```

Host computer with two host applications and IPC for two SINUMERIK controllers



```
Host:       H1
Port:       3010

Host:       H2
Port:       4010

Machine:    M1
IP:         192.168.6.5
Port:       3011

Machine:    M2
IP:         192.168.6.5
Port:       3012
```

MMC.INI

```
[Global]
NcddeMachineNames=NCU840D_1,NCU840D_2
SIM0NcddeDefaultMachineName=NCU840D_1
[NCU840D_1]
ADDRESS0=195.208.2.208,LINE=14,NAME=/NC, ...
[NCU840D_2]
ADDRESS0=195.208.2.209,LINE=14,NAME=/NC, ...
```

MCISRPCSL.INI

```
Host:       H1
IP:         192.168.6.1
Port:       3010

Host:       H2
IP:         192.168.6.1
Port:       4010

Machine:    M1
NCUName:    NCU840D_1
Port:       3011

Machine:    M2
NCUName:    NCU840D_2
Port:       3012
```

4.4 Configuring CMI without DB12

To enable operation without DB12 with Create MyInterface, you must select the "no DB12" option using the "Create MyInterface Configuration" tool.

The "no DB12" option merely prevents CMI from trying to read data from a non-existent DB12. The operation of Create MyInterface "without data block 12" does not replace any functionality implemented or needed by the customer in the PLC.

Note

Exceptions

- The following functionalities are not available:
  - Report tool
  - Workpiece holder status changed
  - Status change

  See also Table 11-2 Request from PLC (Page 200)
- Data that the PCU could otherwise manage and report using DB12 is then not available. The host computer software must take this into account accordingly!
- The machine status data reported in this case with `R_MACHINE_H` do not come from the PLC. Rather they are supplied with the value "0" by CMI. The host computer software must evaluate the "no DB12" option and then take this into account appropriately.

Variable access

With activation of option "NoDB12", the following variable accesses to DB12 are no longer queried by the SINUMERIK but always have the value 0.

1. /Plc/DataBlock/Byte[c12,0]
2. /Plc/DataBlock/Byte-FAST[c12,1]
3. /Plc/DataBlock/Byte[c12,0,19]
4. /Plc/DataBlock/Byte[c12,2]
5. /Plc/DataBlock/Byte[c12,3]
6. /Plc/DataBlock/Byte[c12,4]
7. /Plc/DataBlock/Word[c12,8]
8. /Plc/DataBlock/Bit[c12,2.0]
9. /Plc/DataBlock/Bit[c12,2.1]
10. /Plc/DataBlock/Bit[c12,2.2]
11. /Plc/DataBlock/Bit[c12,2.3]
12. /Plc/DataBlock/Bit[c12,3.2]
13. /Plc/DataBlock/Bit[c12,5.2]

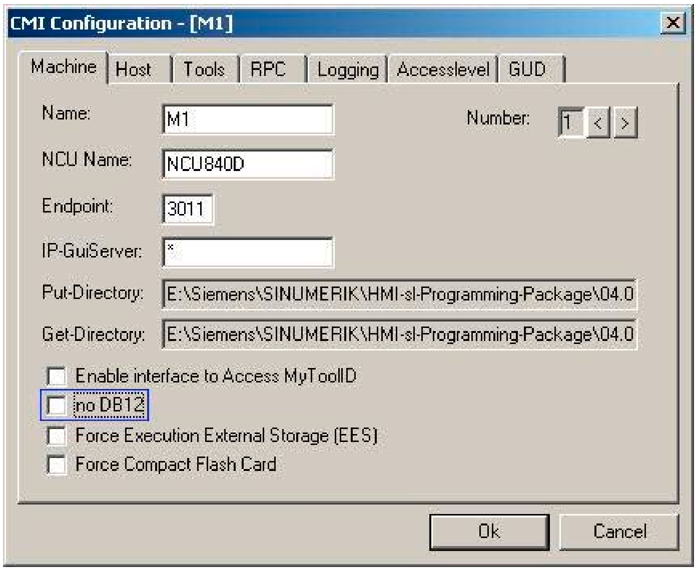
14./Plc/DataBlock/Byte[c12,5]

15./Plc/DataBlock/Byte[c12,0,74]

Additional information can be found in the following section: Sending machine status data (Page 82).

Procedure

1. Start the configuration program "Create MyInterface Configuration"; see also e.g. section: Configuring CMI with Create MyInterface Configuration (Page 32).
2. Select the relevant machine.
3. Open the "Machine" tab.
   – Select the "no DB12" check box.
   – Click "OK" to save the selection.



Name: M1

Number: 1

NCU Name: NCU840D

Endpoint: 3011

IP-GuiServer: *

Put-Directory: E:\Siemens\SINUMERIK\HMI-sl-Programming-Package\04.0

Get-Directory: E:\Siemens\SINUMERIK\HMI-sl-Programming-Package\04.0

□ Enable interface to Access MyToolID

□ no DB12

□ Force Execution External Storage (EES)

□ Force Compact Flash Card

4. Restart the operating software.

4.5 Installing/uninstalling the CMI operator interface on an NCU

4.5.1 Introduction

This section describes the installation of the operator interface of CMI on an NCU.

Note

Performing the installation

Installation must be performed by a Siemens specialist. Please contact your Siemens sales partner if you have any queries.

Note

Installation

- The installation of the operator interface on an NCU always requires the installation of the actual CMI software on a Windows IPC. The actual CMI software cannot be installed on an NCU.
- The configuration of the operator interface of Create MyInterface must be completed before the initial start.

Requirements

- You need ADMIN rights to perform the installation.
- The installation must be carried out from a PC that reaches the controller via the network.
- The following software must have already been installed on the computer.
  - WinSCP V5.5.1; see CMI system requirements (Page 19)
  - .NET Framework 4

Overview

The Create MyInterface (CMI) product consists of two software components:

- Actual functionality
- GUI

The GUI software component is directly installed on an NCU with the SINUMERIK Operate operating software.

Create MyInterface (CMI) must be installed without the CMI GUI component on a suitable IPC in parallel.

Delivery

The NCU Installer works with software packages in the form of ZIP files. A "package" contains all necessary information and files for performing an installation. The package also contains installation instructions in the form of an XML file (manifest.xml).

Package:

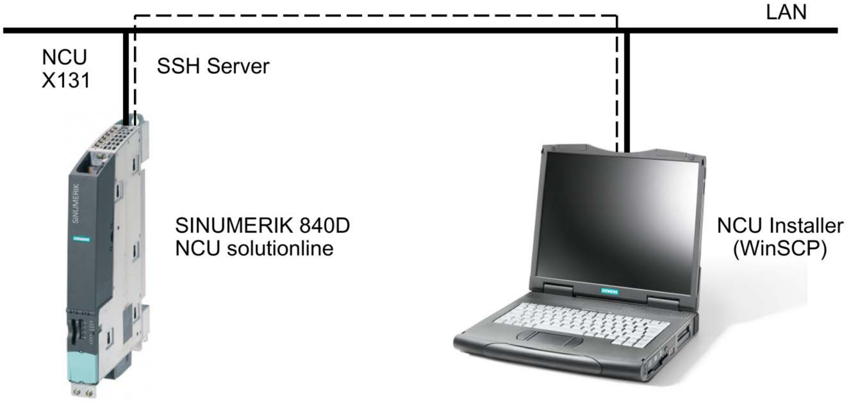
The "CMI_GUINCU.ZIP" software package needed for CMI is part of the CMI product package (full version or SP).

System configuration

"NCU Installer (WinSCP)" is software you use to install the CMI GUI on an NCU from your PC/notebook.

Data is transferred between the PC and the controller in your company network.

The "WinSCP" auxiliary program is used to establish an SSH connection between the installation PC and the controller. With an SSH connection, all data is transported as encrypted data. The "NCU Installer" installation program initiates the SSH connection and transfers the needed files via this connection.

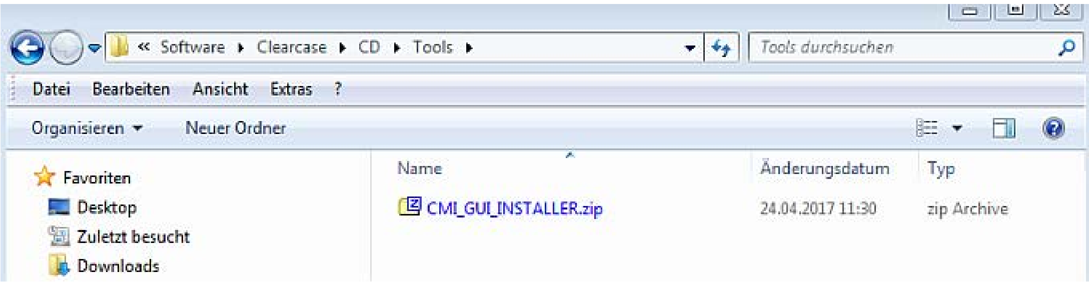


4.5.2 Installing CMI GUI

Procedure

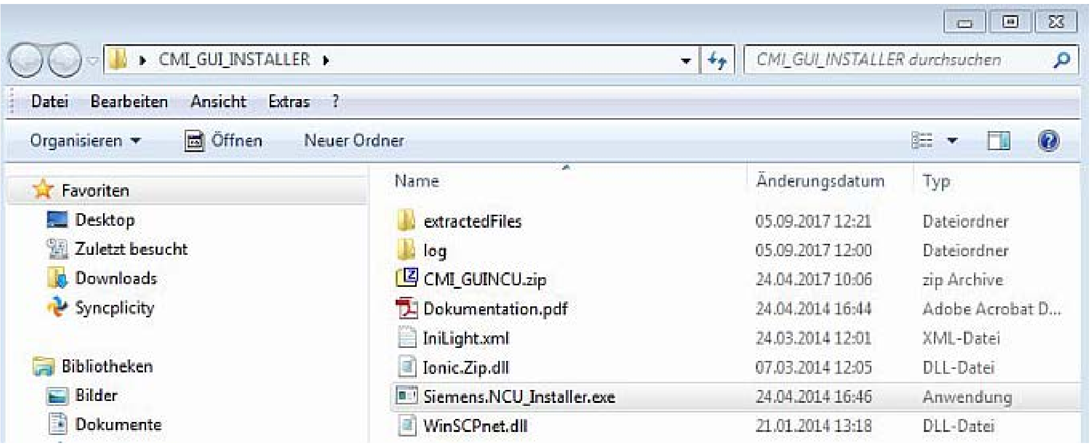
1. You will find the NCU Installer for installing the GUI of Create MyInterface zipped on the product CD of Create MyInterface in the Tools subdirectory.

   Navigate through this directory:


« Software ▸ Clearcase ▸ CD ▸ Tools ▸
Tools durchsuchen
Datei Bearbeiten Ansicht Extras ?
Organisieren ▾ Neuer Ordner
Favoriten
Desktop
Zuletzt besucht
Downloads

Name	Änderungsdatum	Typ
CMI_GUI_INSTALLER.zip	24.04.2017 11:30	zip Archive

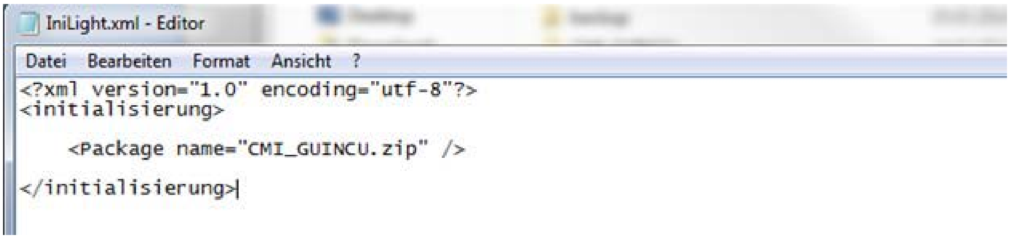
2. Unzip file "CMI_GUI_INSTALLER.zip" into a temporary directory on your PC.



Name	Änderungsdatum	Typ
extractedFiles	05.09.2017 12:21	Dateiordner
log	05.09.2017 12:00	Dateiordner
CMI_GUINCU.zip	24.04.2017 10:06	zip Archive
Dokumentation.pdf	24.04.2014 16:44	Adobe Acrobat D...
IniLight.xml	24.03.2014 12:01	XML-Datei
Ionic.Zip.dll	07.03.2014 12:05	DLL-Datei
Siemens.NCU_Installer.exe	24.04.2014 16:46	Anwendung
WinSCPnet.dll	21.01.2014 13:18	DLL-Datei

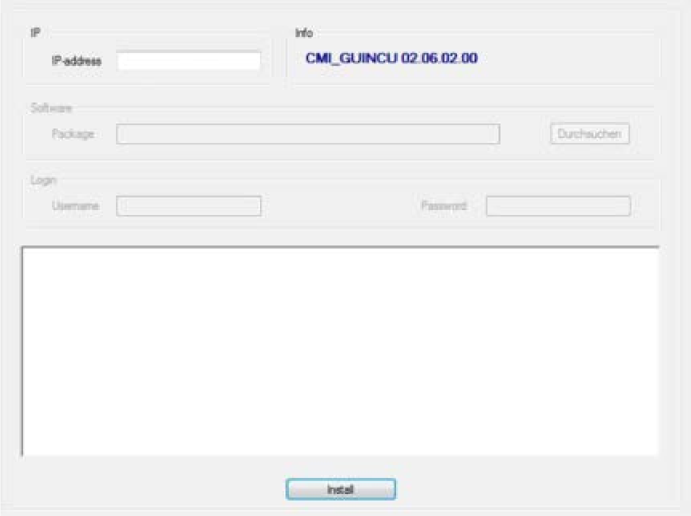
3. Start the NCU installer by double-clicking file "Siemens.NCU_Installer.exe".

4. The package specified in the "IniLight.xml" configuration file is loaded.



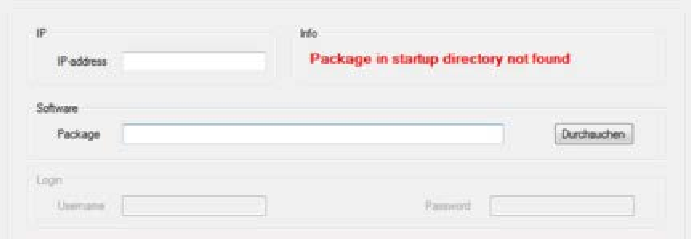
```
<?xml version="1.0" encoding="utf-8"?>
<initialisierung>

    <Package name="CMI_GUINCU.zip" />

</initialisierung>
```

5. If the NCU Installer finds the package specified in the "IniLight.xml" file, the XML file "manifest.xml" is read from this package, see also section: XML file "manifest.xml" (Page 55).
   The name and version number of the software to be installed is shown in the "Info" display field.



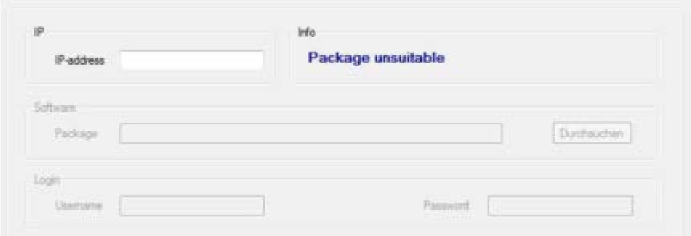
CMI_GUINCU 02.06.02.00

- OR -
If the NCU installer does not find the package specified in the "IniLight.xml" file, the NCU installer searches for a package named "Package.zip" in the Start directory.
- OR -
If this package is also not found, an error message is displayed in the "Info" display field.
The "Package" input field is activated in the "Software" area.
Click the "Browse" button to conduct your own package search and to load the package.



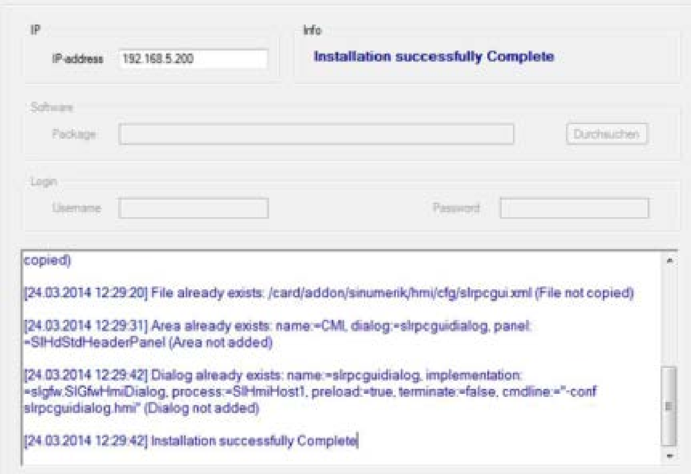
Package in startup directory not found

If you choose a package that was not created according to the instructions of the package manufacturer, the package will not be loaded.
You receive a corresponding message in the "Info" display field.



Package unsuitable

6. When the desired package has been selected and loaded, you must enter the IP address of the controller.
   In the "IP address" input field in the "IP" area, enter the address under which the package is to be installed.
   Click the "Install" button to start the installation.
7. You can receive a success message.

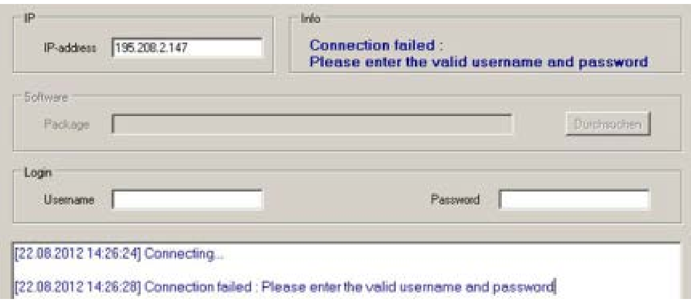


IP-address 192.168.5.200

Installation successfully Complete

```
copied)

[24.03.2014 12:29:20] File already exists: /card/addon/sinumerik/hmi/cfg/slrpcgui.xml (File not copied)

[24.03.2014 12:29:31] Area already exists: name:=CMI, dialog:=slrpcguidialog, panel:
=SlHdStdHeaderPanel (Area not added)

[24.03.2014 12:29:42] Dialog already exists: name:=slrpcguidialog, implementation:
=slgfw.SlGfwHmiDialog, process:=SlHmiHost1, preload:=true, terminate:=false, cmdline:="-conf
slrpcguidialog.hmi" (Dialog not added)

[24.03.2014 12:29:42] Installation successfully Complete
```

- OR -
You receive an error message in the "Info" display field if the NCU Installer with the standard data does not connect to the controller.
The "Username" and "Password" input fields are activated in the "Login" area.
Enter the user name and password.
Click the "Install" button to restart the installation.



IP-address 195.208.2.147

Connection failed :
Please enter the valid username and password

```
[22.08.2012 14:26:24] Connecting...

[22.08.2012 14:26:28] Connection failed : Please enter the valid username and password
```

4.5.3 XML file "manifest.xml"

Note

Do not make any manual change

You must not make any changes to file "manifest.xml".

Structure of the file

The document has a root element named <Package> with the following attributes:

Attribute	Description
`Name`	Software name
`Version`	Software version information
`Description`	Description
`Target`	Target: Target directory on the NCU

The root element can contain additional subelements:

Subelement	Attribute	Description
`File`	`src` `dest`	The `<File>` element inserts a file into the NCU during installation.
`Service`	`value` `dest`	The `<Service>` element adds service entries to the "SYSTEMCONFIGURATION.INI" file.
`Area`	`value` `dest`	The `<Area>` element adds area entries to the SYSTEMCONFIG.INI file.
`Firewall`	`port` `protocol`	The `<Firewall>` element is responsible for the port enable.
`Dialog`	`value` `dest`	The `<Dialog>` element adds dialog entries to the "SYSTEMCONFIGURATION.INI" file.
`Directory`	`name` `dest`	The `<Directory>` element create a new folder on the controller.

`Dir`	`section`
	`key`
	`value`
	`dest`
		The `<Dir>` element adds Dir entries to the "SYSTEMCONFIGURATION.INI" file.

4.5.4 Parameters for CMI GUI

File or entry

The following files or entries are involved:

Table 4- 5 Parameters for CMI GUI

File or entry	Target directory
libslrpcgui.so	/card/addon/sinumerik/hmi/appl
slrpcgui.xml	/card/addon/sinumerik/hmi/cfg
slrpcgui _deu.qm	/card/addon/sinumerik/hmi/lng
slrpcgui _eng.qm	/card/addon/sinumerik/hmi/lng
slrpcguidialog.hmi	/card/addon/sinumerik/hmi/appl

`[areas] AREA###= name:=CMI, dialog:= slrpcguidialog, panel:=SlHdStdHeaderPanel"`
`dest="/card/addon/sinumerik/hmi/cfg`
	/card/addon/sinumerik/hmi/cfg
	File "SYSTEMCONFIGURATION.INI"

`[dialogs] DLG###= name:= slrpcguidialog, implementation:= slgfw.SlGfwHmiDialog, process:= SlHmiHost1, preload:=true, terminate:= false, cmdline:="-conf slrpcguidialog.hmi"'`
`dest="/card/addon/sinumerik/hmi/cfg`
	/card/addon/sinumerik/hmi/cfg
	File "SYSTEMCONFIGURATION.INI"

`[Firewall]`	No entries are needed in the "BASESYS.INI" file, the CMI-IPC is the server.

Procedure

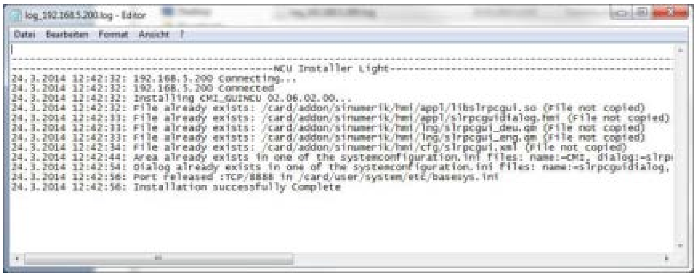
1. Insert the following files in the corresponding target directories during installation:
   - libslrpcgui.so
   - slrpcgui.xml
   - slrpcgui _deu.qm
   - slrpcgui _eng.qm
   - slrpcguidialog.hmi
2. Insert the entries of `[areas]` and `[dialogs]` into the "SYSTEMCONFIGURATION.INI" configuration file.

4.5.5 Log files

Overview

A log file is created in the "log" subdirectory when the NCU Installer is connected to a controller.

The file name contains the IP address of the controller.



log_192.168.5.200.log - Editor
Datei Bearbeiten Format Ansicht ?

```
-------------------------------------------------------------------------------------------------------------
-----------------------------------------NCU Installer Light------------------------------------------------
24.3.2014 12:42:32: 192.168.5.200 Connecting...
24.3.2014 12:42:32: 192.168.5.200 Connected
24.3.2014 12:42:32: Installing CMI_GUINCU 02.06.02.00...
24.3.2014 12:42:32: File already exists: /card/addon/sinumerik/hmi/appl/libslrpcgui.so (File not copied)
24.3.2014 12:42:33: File already exists: /card/addon/sinumerik/hmi/appl/slrpcguidialog.hmi (File not copied)
24.3.2014 12:42:33: File already exists: /card/addon/sinumerik/hmi/lng/slrpcgui_deu.qm (File not copied)
24.3.2014 12:42:33: File already exists: /card/addon/sinumerik/hmi/lng/slrpcgui_eng.qm (File not copied)
24.3.2014 12:42:34: File already exists: /card/addon/sinumerik/hmi/cfg/slrpcgui.xml (File not copied)
24.3.2014 12:42:44: Area already exists in one of the systemconfiguration.ini files: name:=CMI, dialog:=slrp
24.3.2014 12:42:54: Dialog already exists in one of the systemconfiguration.ini files: name:=slrpcguidialog,
24.3.2014 12:42:56: Port released :TCP/8888 in /card/user/system/etc/basesys.ini
24.3.2014 12:42:56: Installation successfully Complete
```

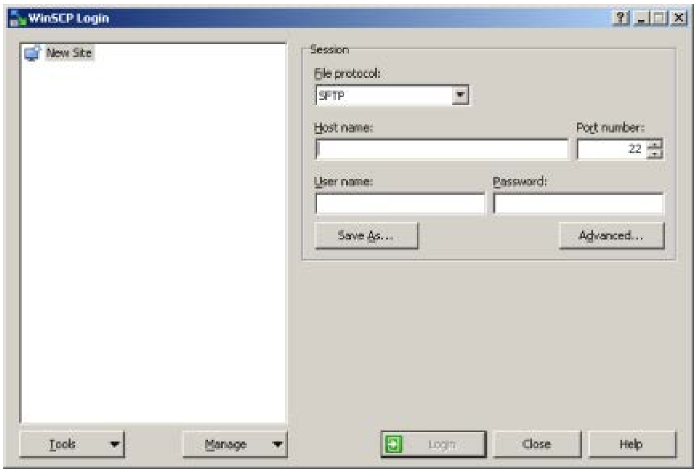
4.5.6 Uninstalling the CMI user interface

The information provided in this section refer to the user interface of Create MyInterface under SINUMERIK Operate for NCU.

- To uninstall the Create MyInterface user interface under SINUMERIK Operate for NCU, WinSCP 5.5.1 must be installed on the PC, see also section: CMI system requirements (Page 19); section: "Additional supplementary conditions" → Software link.
- Uninstall CMI manually.

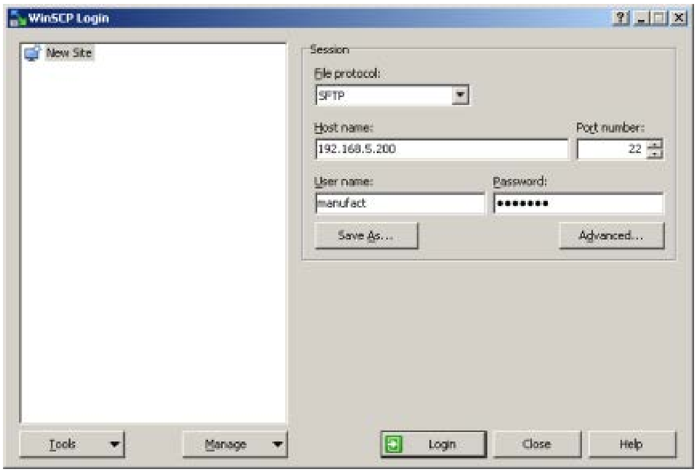
Procedure

1. Start the "WinSCP" software.
   The "WinSCP Login" window opens.



File protocol: SFTP
Host name:
Port number: 22
User name:
Password:

2. Enter the IP address of the controller in the "Host name:" input field.
   Enter the user name and password in the "User name:" and "Password:" input fields.
   Click the "Login" button.



WinSCP Login

Session

File protocol: SFTP

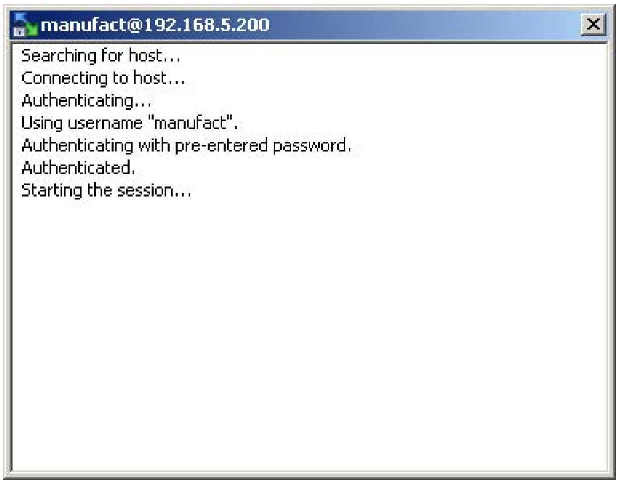
Host name: 192.168.5.200

Port number: 22

User name: manufact

Password: •••••••

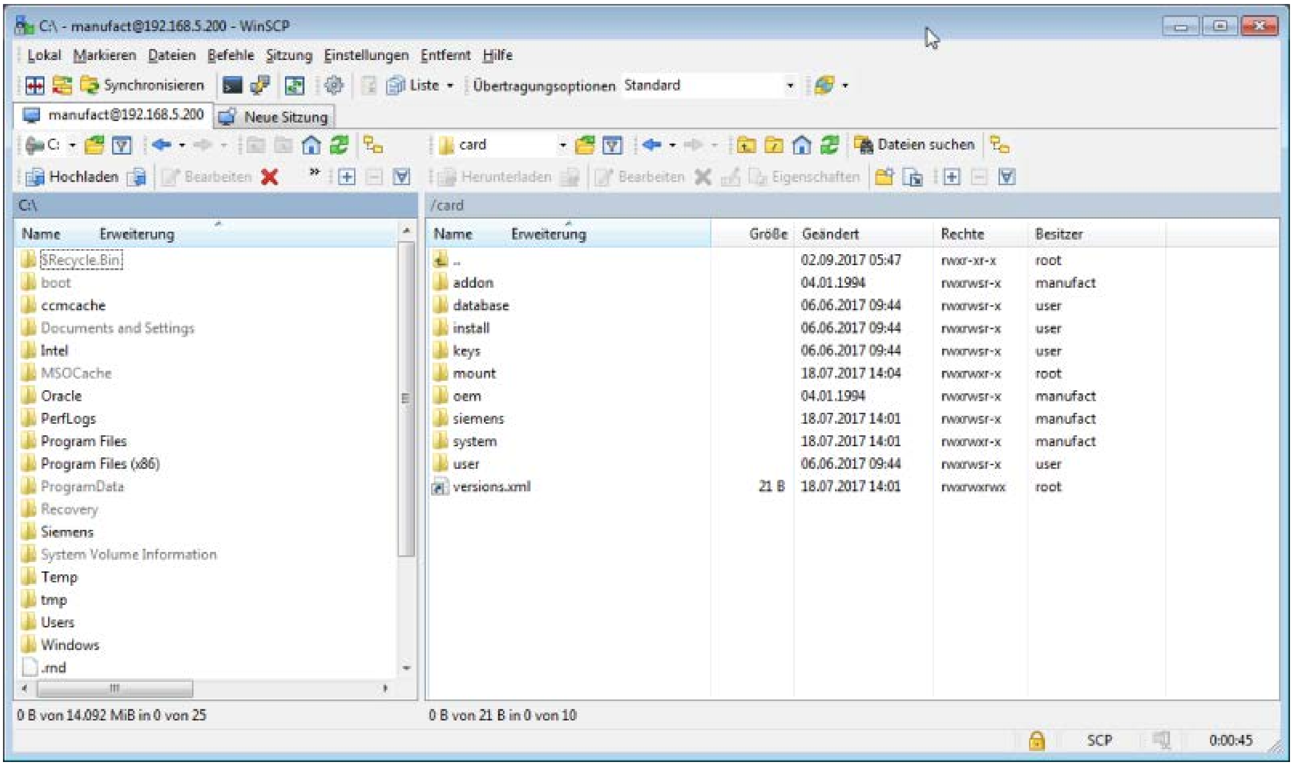
3. The login operation is started and its progress is displayed.



manufact@192.168.5.200

```
Searching for host...
Connecting to host...
Authenticating...
Using username "manufact".
Authenticating with pre-entered password.
Authenticated.
Starting the session...
```

4. After successful connection to the controller, the following window is displayed.


C:\ - manufact@192.168.5.200 - WinSCP
Lokal Markieren Dateien Befehle Sitzung Einstellungen Entfernt Hilfe
Synchronisieren
Liste
Übertragungsoptionen Standard
manufact@192.168.5.200
Neue Sitzung
C:
card
Dateien suchen
Hochladen
Bearbeiten
Herunterladen
Eigenschaften
SCP
0:00:45

C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

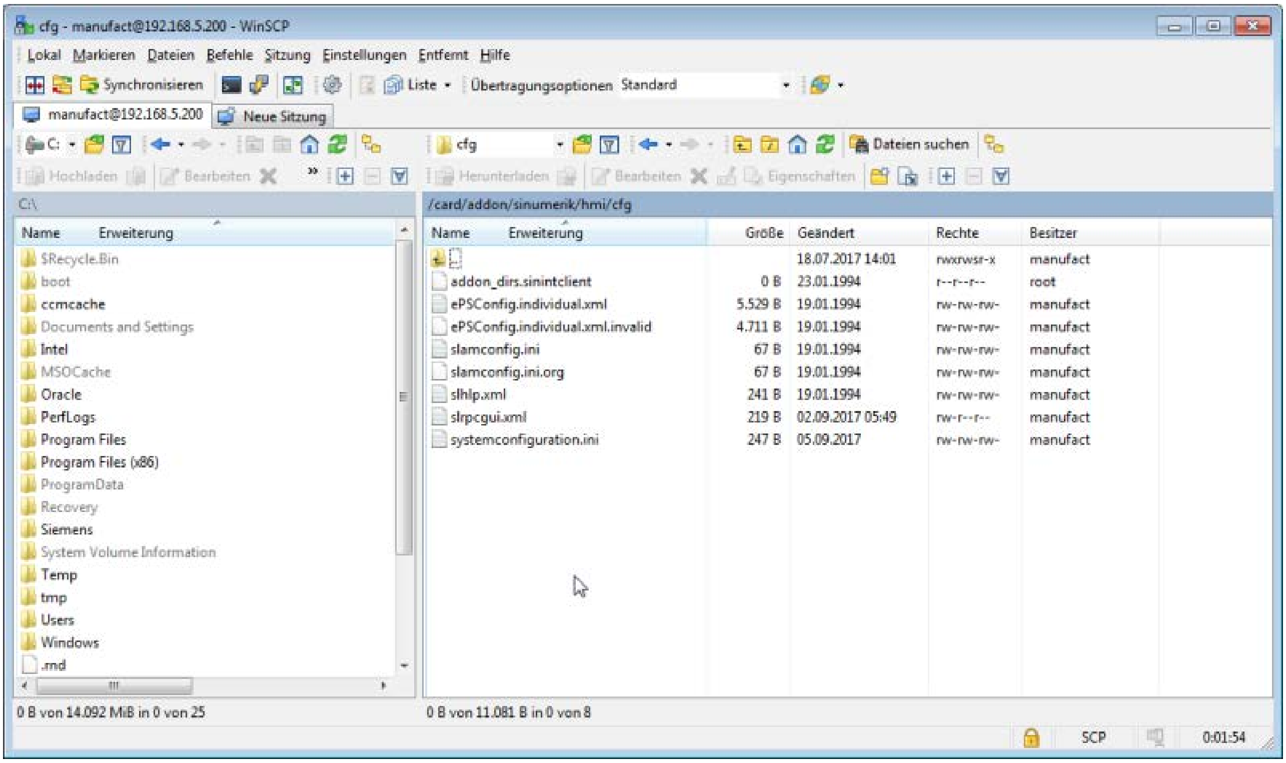
0 B von 14.092 MiB in 0 von 25

/card

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
..			02.09.2017 05:47	rwxr-xr-x	root
addon			04.01.1994	rwxrwsr-x	manufact
database			06.06.2017 09:44	rwxrwsr-x	user
install			06.06.2017 09:44	rwxrwsr-x	user
keys			06.06.2017 09:44	rwxrwsr-x	user
mount			18.07.2017 14:04	rwxrwxr-x	root
oem			04.01.1994	rwxrwsr-x	manufact
siemens			18.07.2017 14:01	rwxrwsr-x	manufact
system			18.07.2017 14:01	rwxrwxr-x	manufact
user			06.06.2017 09:44	rwxrwsr-x	user
versions.xml		21 B	18.07.2017 14:01	rwxrwxrwx	root

0 B von 21 B in 0 von 10

5. In the right window area, navigate to the following directory: /card/addon/sinumerik/hmi/cfg.


cfg - manufact@192.168.5.200 - WinSCP
Lokal Markieren Dateien Befehle Sitzung Einstellungen Entfernt Hilfe
Synchronisieren
Liste
Übertragungsoptionen Standard
manufact@192.168.5.200
Neue Sitzung
C:
cfg
Dateien suchen
Hochladen
Bearbeiten
Herunterladen
Eigenschaften
SCP
0:01:54

C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

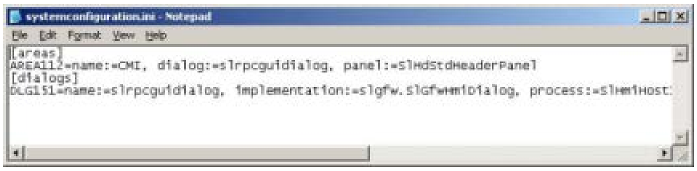
0 B von 14.092 MiB in 0 von 25

/card/addon/sinumerik/hmi/cfg

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
			18.07.2017 14:01	rwxrwsr-x	manufact
addon_dirs.sinintclient		0 B	23.01.1994	r--r--r--	root
ePSConfig.individual.xml		5.529 B	19.01.1994	rw-rw-rw-	manufact
ePSConfig.individual.xml.invalid		4.711 B	19.01.1994	rw-rw-rw-	manufact
slamconfig.ini		67 B	19.01.1994	rw-rw-rw-	manufact
slamconfig.ini.org		67 B	19.01.1994	rw-rw-rw-	manufact
slhlp.xml		241 B	19.01.1994	rw-rw-rw-	manufact
slrpcgui.xml		219 B	02.09.2017 05:49	rw-r--r--	manufact
systemconfiguration.ini		247 B	05.09.2017	rw-rw-rw-	manufact

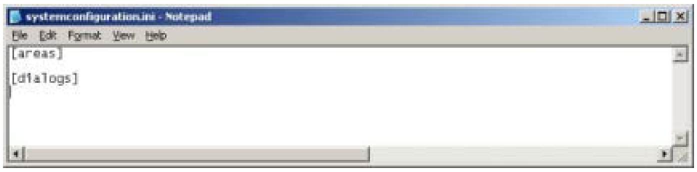
0 B von 11.081 B in 0 von 8

6. Double-click on the "SYSTEMCONFIGURATION.INI" file to open it.



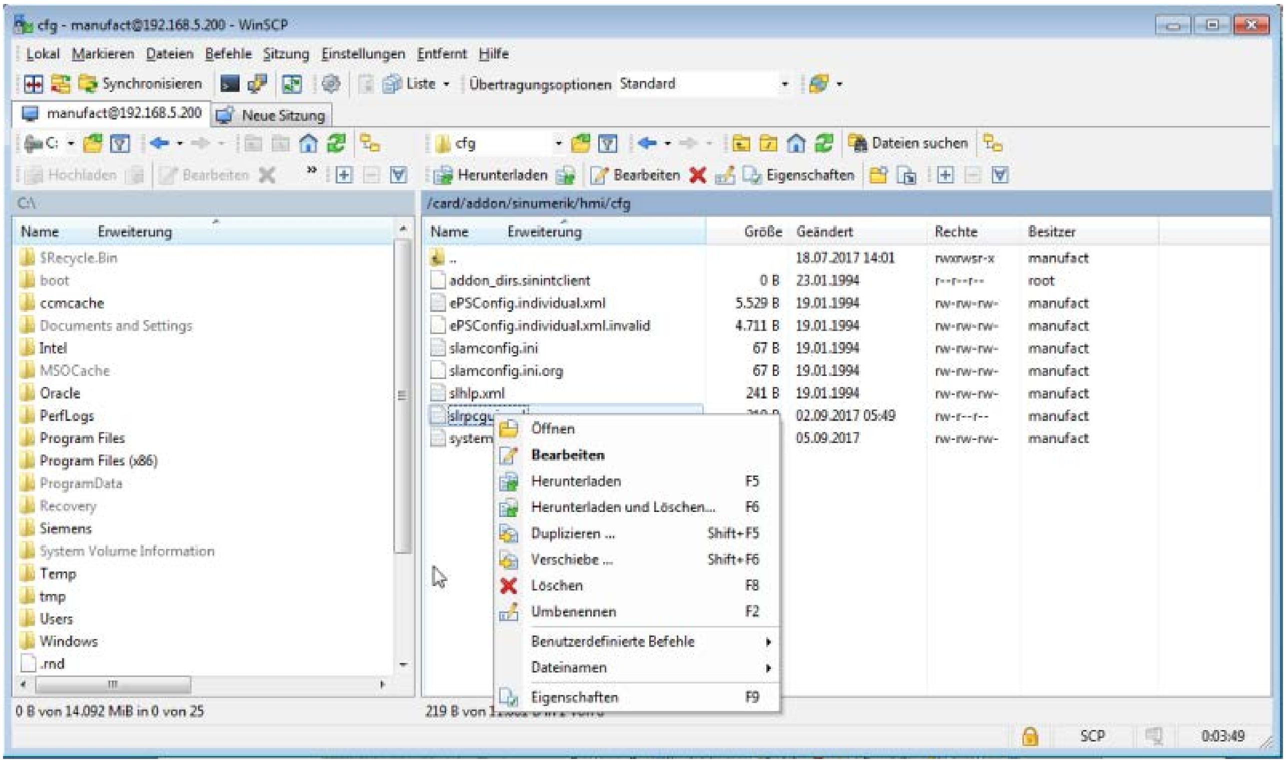
```
[areas]
AREA112=name:=CMI, dialog:=slrpcguidialog, panel:=SlHdStdHeaderPanel
[dialogs]
DLG151=name:=slrpcguidialog, implementation:=slgfw.SlGfwHmiDialog, process:=SlHmiHost:
```

7. Delete in full the entry in the `[areas]` attribute that contains the name:=CMI.
   Delete in full the entry in the `[dialogs]` attribute that contains the name:=slrpcguidialog.
   Save the "SYSTEMCONFIGURATION.INI" file.



```
[areas]

[dialogs]
```

8. Select the "slrpcgui.xml" file in the right window area.
   Right-click to open the menu and click the "Delete" menu command.
   - OR -
   Press the <F8> key to delete the file.



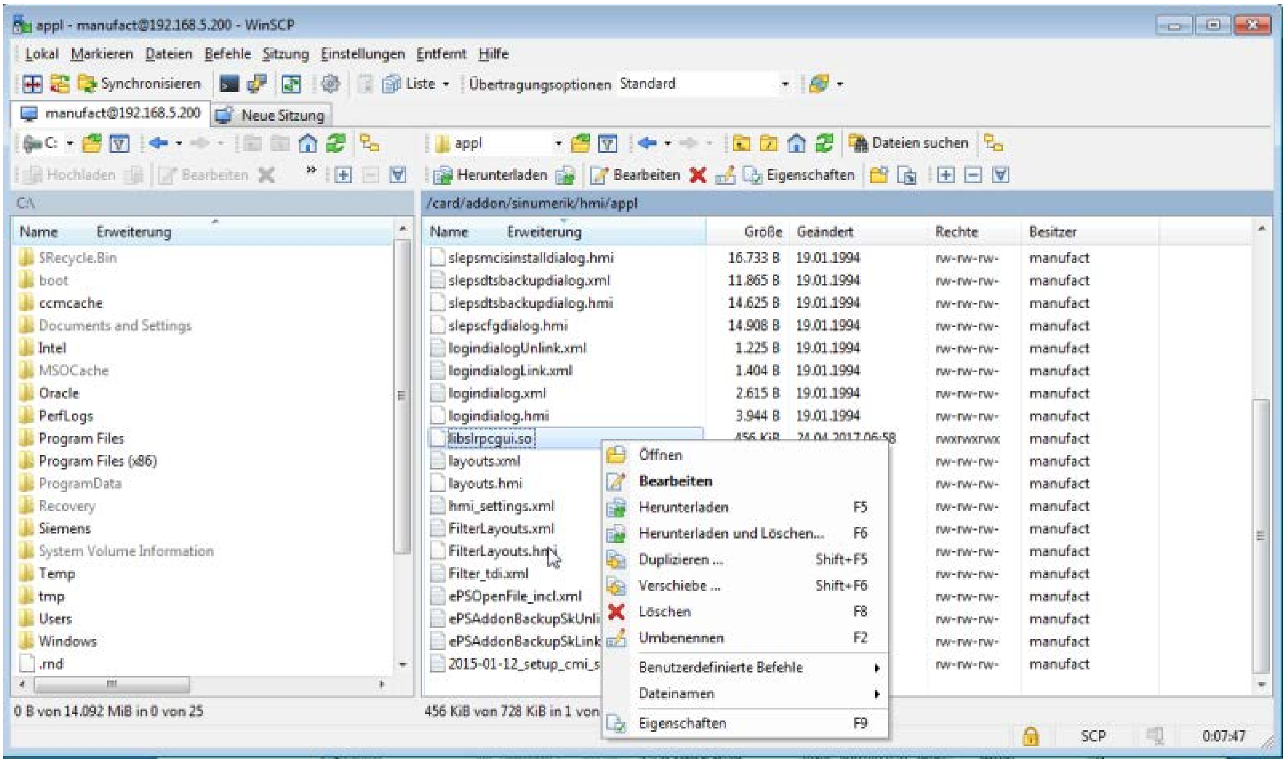
Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

0 B von 14.092 MiB in 0 von 25

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
..			18.07.2017 14:01	rwxrwsr-x	manufact
addon_dirs.sinintclient		0 B	23.01.1994	r--r--r--	root
ePSConfig.individual.xml		5.529 B	19.01.1994	rw-rw-rw-	manufact
ePSConfig.individual.xml.invalid		4.711 B	19.01.1994	rw-rw-rw-	manufact
slamconfig.ini		67 B	19.01.1994	rw-rw-rw-	manufact
slamconfig.ini.org		67 B	19.01.1994	rw-rw-rw-	manufact
slhlp.xml		241 B	19.01.1994	rw-rw-rw-	manufact
slrpcgu			02.09.2017 05:49	rw-r--r--	manufact
system			05.09.2017	rw-rw-rw-	manufact

219 B von

9. In the right window area, navigate to the following directory:
   /card/addon/sinumerik/hmi/appl.
   Select the "ibslrpcgui.so" file.
   Right-click to open the menu and click the "Delete" menu command.
   - OR -
   Press the <F8> key to delete the file.



C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

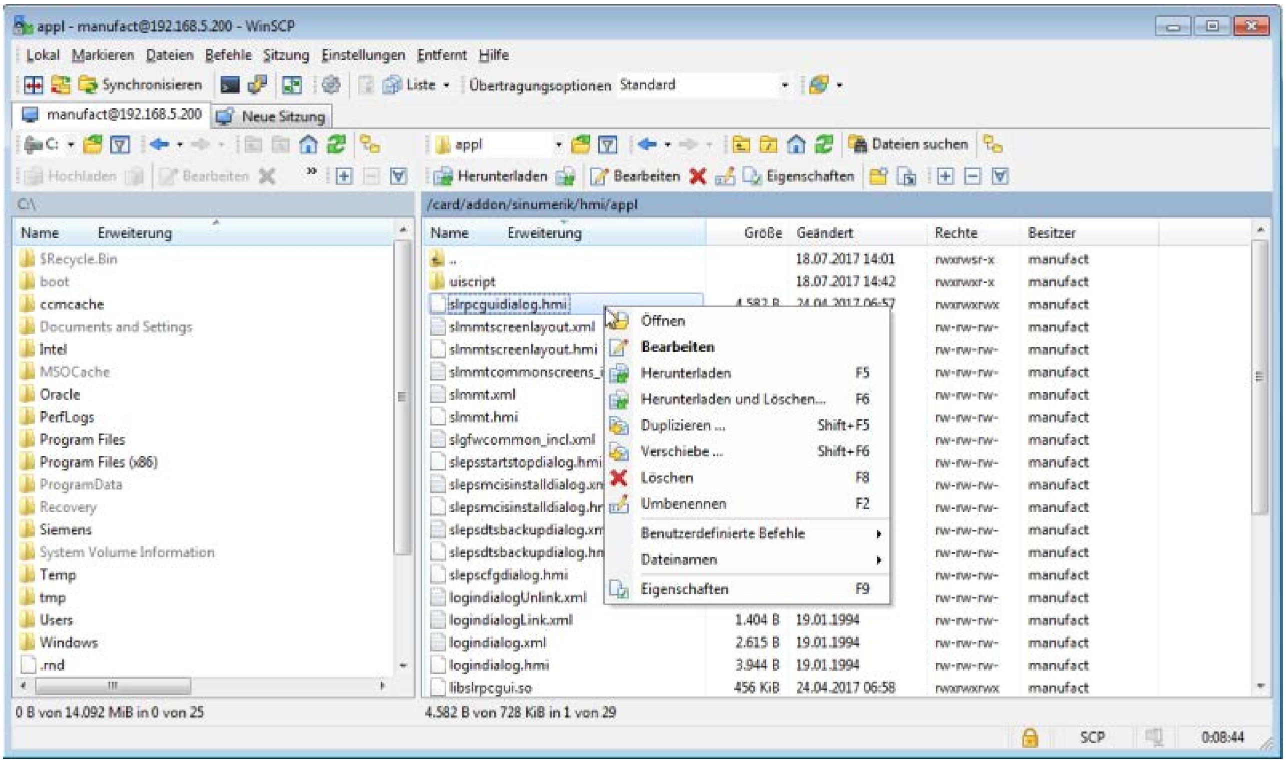
0 B von 14.092 MiB in 0 von 25

/card/addon/sinumerik/hmi/appl

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
slepsmcisinstalldialog.hmi		16.733 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackupdialog.xml		11.865 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackupdialog.hmi		14.625 B	19.01.1994	rw-rw-rw-	manufact
slepscfgdialog.hmi		14.908 B	19.01.1994	rw-rw-rw-	manufact
logindialogUnlink.xml		1.225 B	19.01.1994	rw-rw-rw-	manufact
logindialogLink.xml		1.404 B	19.01.1994	rw-rw-rw-	manufact
logindialog.xml		2.615 B	19.01.1994	rw-rw-rw-	manufact
logindialog.hmi		3.944 B	19.01.1994	rw-rw-rw-	manufact
libslrpcgui.so		456 KiB	24.04.2017 06:58	rwxrwxrwx	manufact
layouts.xml				rw-rw-rw-	manufact
layouts.hmi				rw-rw-rw-	manufact
hmi_settings.xml				rw-rw-rw-	manufact
FilterLayouts.xml				rw-rw-rw-	manufact
FilterLayouts.hmi				rw-rw-rw-	manufact
Filter_tdi.xml				rw-rw-rw-	manufact
ePSOpenFile_incl.xml				rw-rw-rw-	manufact
ePSAddonBackupSkUnli				rw-rw-rw-	manufact
ePSAddonBackupSkLink				rw-rw-rw-	manufact
2015-01-12_setup_cmi_s				rw-rw-rw-	manufact

456 KiB von 728 KiB in 1 von

10. Select the "slrpcguidialog.hmi" file.
    Right-click to open the menu and click the "Delete" menu command.
    - OR -
    Press the <F8> key to delete the file.



C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

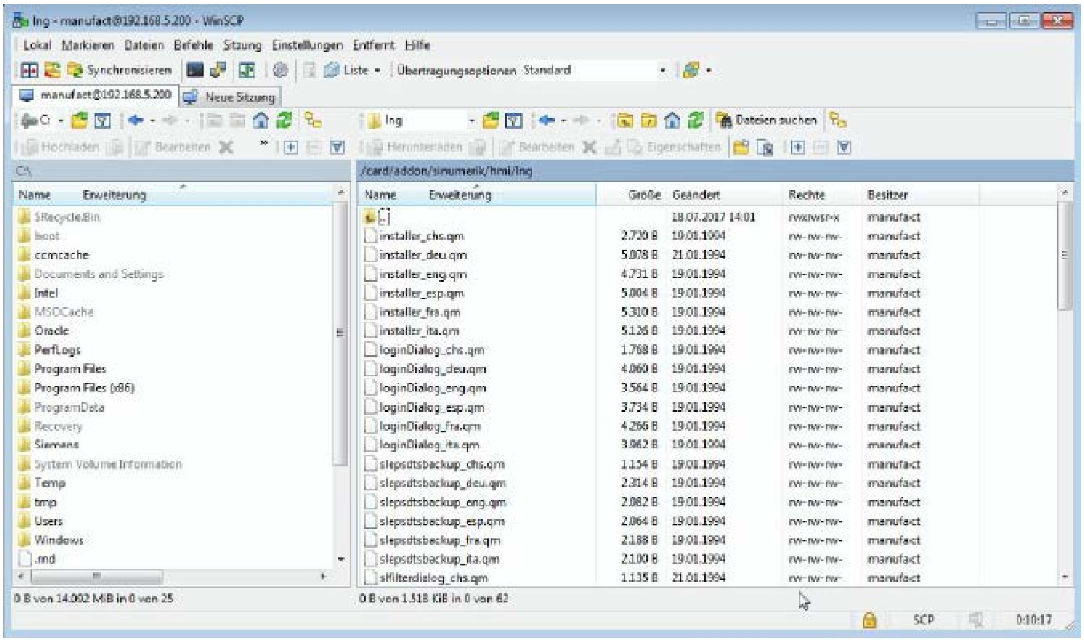
/card/addon/sinumerik/hmi/appl

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
..			18.07.2017 14:01	rwxrwsr-x	manufact
uiscript			18.07.2017 14:42	rwxrwxr-x	manufact
slrpcguidialog.hmi		4.582 B	24.04.2017 06:57	rwxrwxrwx	manufact
slmmtscreenlayout.xml				rw-rw-rw-	manufact
slmmtscreenlayout.hmi				rw-rw-rw-	manufact
slmmtcommonscreens_i				rw-rw-rw-	manufact
slmmt.xml				rw-rw-rw-	manufact
slmmt.hmi				rw-rw-rw-	manufact
slgfwcommon_incl.xml				rw-rw-rw-	manufact
slepsstartstopdialog.hmi				rw-rw-rw-	manufact
slepsmcisinstalldialog.xn				rw-rw-rw-	manufact
slepsmcisinstalldialog.hr				rw-rw-rw-	manufact
slepsdtsbackupdialog.xn				rw-rw-rw-	manufact
slepsdtsbackupdialog.hn				rw-rw-rw-	manufact
slepscfgdialog.hmi				rw-rw-rw-	manufact
logindialogUnlink.xml				rw-rw-rw-	manufact
logindialogLink.xml		1.404 B	19.01.1994	rw-rw-rw-	manufact
logindialog.xml		2.615 B	19.01.1994	rw-rw-rw-	manufact
logindialog.hmi		3.944 B	19.01.1994	rw-rw-rw-	manufact
libslrpcgui.so		456 KiB	24.04.2017 06:58	rwxrwxrwx	manufact

0 B von 14.092 MiB in 0 von 25

4.582 B von 728 KiB in 1 von 29

11. In the right window area, navigate to directory "/card/addon/sinumerik/hmi/lng".



C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

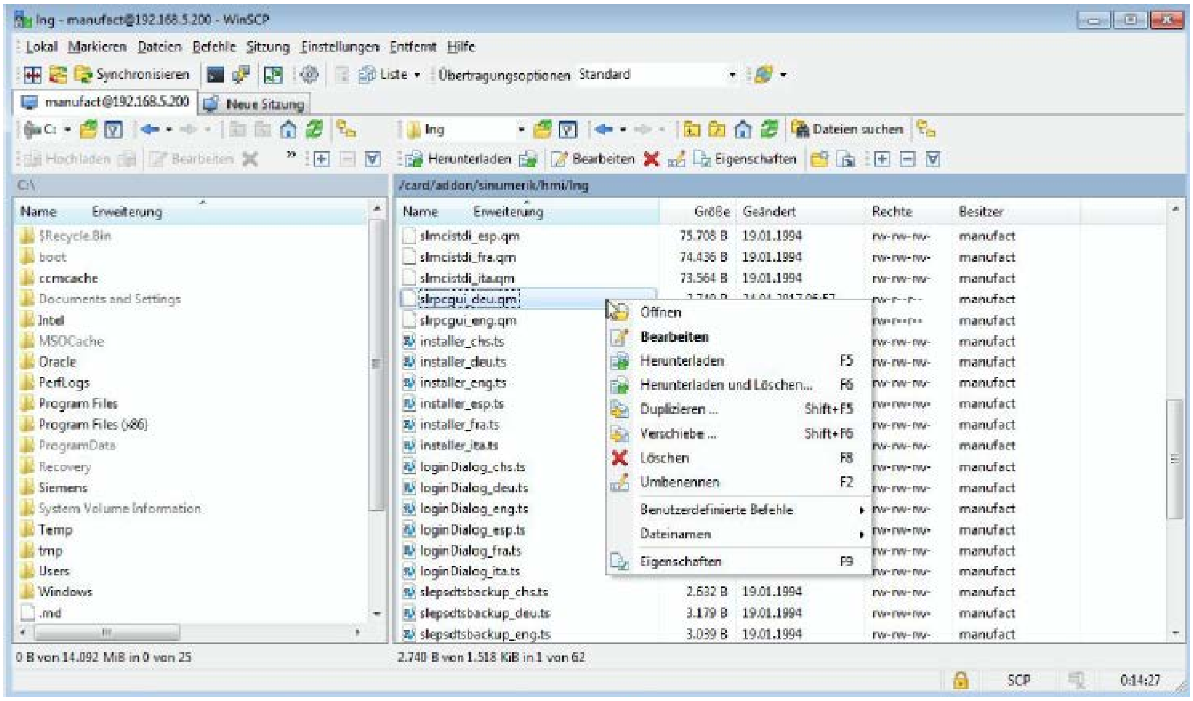
/card/addon/sinumerik/hmi/lng

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
..			18.07.2017 14:01	rwxrwsr-x	manufact
installer_chs.qm		2.720 B	19.01.1994	rw-rw-rw-	manufact
installer_deu.qm		5.078 B	21.01.1994	rw-rw-rw-	manufact
installer_eng.qm		4.731 B	19.01.1994	rw-rw-rw-	manufact
installer_esp.qm		5.004 B	19.01.1994	rw-rw-rw-	manufact
installer_fra.qm		5.310 B	19.01.1994	rw-rw-rw-	manufact
installer_ita.qm		5.126 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_chs.qm		1.768 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_deu.qm		4.060 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_eng.qm		3.564 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_esp.qm		3.734 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_fra.qm		4.266 B	19.01.1994	rw-rw-rw-	manufact
loginDialog_ita.qm		3.862 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_chs.qm		1.154 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_deu.qm		2.314 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_eng.qm		2.082 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_esp.qm		2.064 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_fra.qm		2.188 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_ita.qm		2.100 B	19.01.1994	rw-rw-rw-	manufact
slfilterdialog_chs.qm		1.135 B	21.01.1994	rw-rw-rw-	manufact

0 B von 14.092 MiB in 0 von 25

0 B von 1.318 KiB in 0 von 62

12. In the right window area, mark file "slrpcgui_deu.qm".
    Right-click to open the shortcut menu and choose menu item "Delete".
    - OR -
    Press the <F8> key to delete the file.



C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

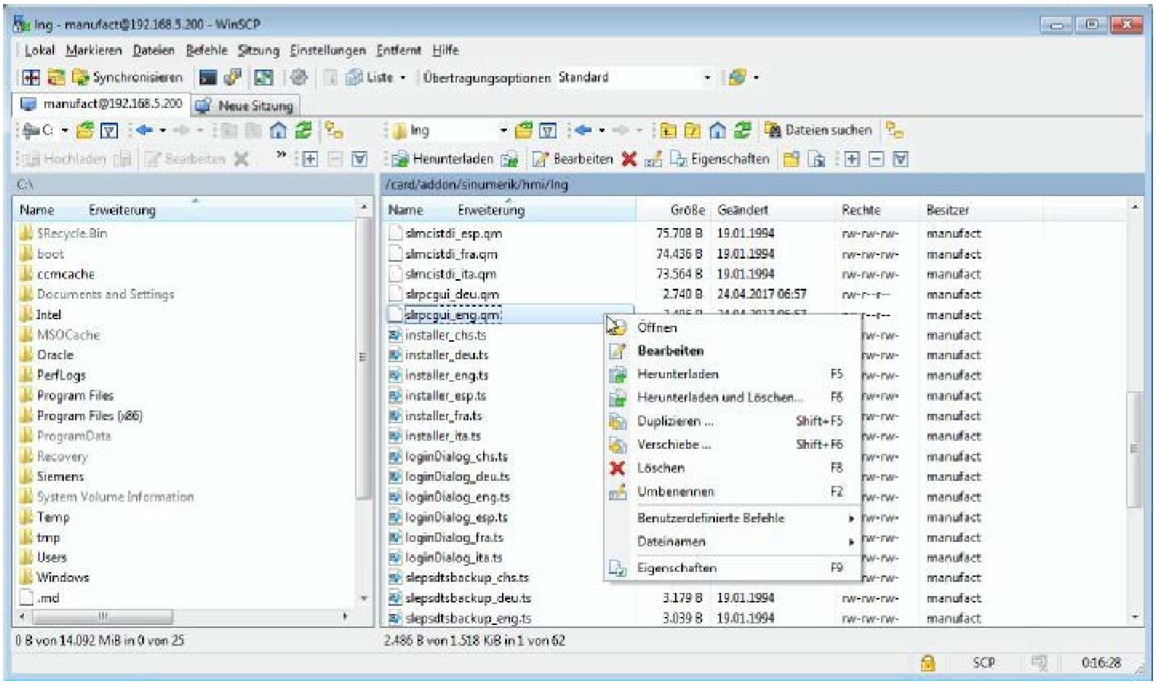
0 B von 14.092 MiB in 0 von 25

/card/addon/sinumerik/hmi/lng

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
slmcistdi_esp.qm		75.708 B	19.01.1994	rw-rw-rw-	manufact
slmcistdi_fra.qm		74.436 B	19.01.1994	rw-rw-rw-	manufact
slmcistdi_ita.qm		73.564 B	19.01.1994	rw-rw-rw-	manufact
slrpcgui_deu.qm		2.740 B	14.01.2017 06:57	rw-r--r--	manufact
slrpcgui_eng.qm				rw-r--r--	manufact
installer_chs.ts				rw-rw-rw-	manufact
installer_deu.ts				rw-rw-rw-	manufact
installer_eng.ts				rw-rw-rw-	manufact
installer_esp.ts				rw-rw-rw-	manufact
installer_fra.ts				rw-rw-rw-	manufact
installer_ita.ts				rw-rw-rw-	manufact
loginDialog_chs.ts				rw-rw-rw-	manufact
loginDialog_deu.ts				rw-rw-rw-	manufact
loginDialog_eng.ts				rw-rw-rw-	manufact
loginDialog_esp.ts				rw-rw-rw-	manufact
loginDialog_fra.ts				rw-rw-rw-	manufact
loginDialog_ita.ts				rw-rw-rw-	manufact
slepsdtsbackup_chs.ts		2.632 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_deu.ts		3.179 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_eng.ts		3.039 B	19.01.1994	rw-rw-rw-	manufact

2.740 B von 1.518 KiB in 1 von 62

13. In the right window area, mark file "slrpcgui_eng.qm".
    Right-click to open the shortcut menu and choose menu item "Delete".
    - OR -
    Press the <F8> key to delete the file.



C:\

Name	Erweiterung
$Recycle.Bin	
boot	
ccmcache	
Documents and Settings	
Intel	
MSOCache	
Oracle	
PerfLogs	
Program Files	
Program Files (x86)	
ProgramData	
Recovery	
Siemens	
System Volume Information	
Temp	
tmp	
Users	
Windows	
.rnd	

0 B von 14.092 MiB in 0 von 25

/card/addon/sinumerik/hmi/lng

Name	Erweiterung	Größe	Geändert	Rechte	Besitzer
slmcistdi_esp.qm		75.708 B	19.01.1994	rw-rw-rw-	manufact
slmcistdi_fra.qm		74.436 B	19.01.1994	rw-rw-rw-	manufact
slmcistdi_ita.qm		73.564 B	19.01.1994	rw-rw-rw-	manufact
slrpcgui_deu.qm		2.740 B	24.04.2017 06:57	rw-r--r--	manufact
slrpcgui_eng.qm				--r--	manufact
installer_chs.ts				w-rw-	manufact
installer_deu.ts				w-rw-	manufact
installer_eng.ts				w-rw-	manufact
installer_esp.ts				w-rw-	manufact
installer_fra.ts				w-rw-	manufact
installer_ita.ts				w-rw-	manufact
loginDialog_chs.ts				w-rw-	manufact
loginDialog_deu.ts				w-rw-	manufact
loginDialog_eng.ts				w-rw-	manufact
loginDialog_esp.ts				w-rw-	manufact
loginDialog_fra.ts				w-rw-	manufact
loginDialog_ita.ts				w-rw-	manufact
slepsdtsbackup_chs.ts				w-rw-	manufact
slepsdtsbackup_deu.ts		3.179 B	19.01.1994	rw-rw-rw-	manufact
slepsdtsbackup_eng.ts		3.039 B	19.01.1994	rw-rw-rw-	manufact

2.486 B von 1.518 KiB in 1 von 62

14. Uninstallation of the user interface of Create MyInterface GUI for SINUMERIK Operate for NCU is now complete.
    Exit with "WinSCP".

15. Restart SINUMERIK Operate for NCU.

4.6 Testing of the installation on Windows with RPC.OCX

You must install RPC.OCX on the host computer in order to test the installation.

RPC.OCX is a component of the "RpcSinumerikTest.exe" test program.

RPC.OCX is supplied in an additional "RPC_SINUMERIK_OCX_SETUP.zip" file, but it is not automatically installed.

Parameters

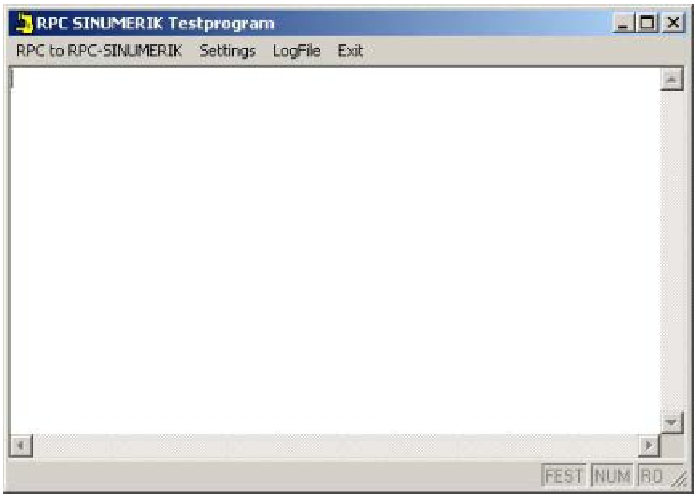
Parameter	Description
"Host" tab	
• HostID	Enter the HostID "H1" here.
• HostPort	Specification of the port number of the host
• HostEnabled	Select check box: Check mark set: yes (true) No check mark: no (false)
• Number of Machines	Specification of the machine number You toggle between the configuration data with the "<" and ">" buttons.
"Machine" tab	
• Machine	Specification of the machine number
• MachineID	Two variants of the specification of the ID number of the machine • String "M1" - OR - • The name that was used in the configuration of the Create MyInterface Client.
• MachinePort	Specification of the port number of the machine
• MachineIP	Specification of the IP address of the machine • IP address of the MCIS RPC Client (i.e. own IP address, no computer name)

Settings for RPC SINUMERIK Test

1. Start the RPC SINUMERIK Test program.

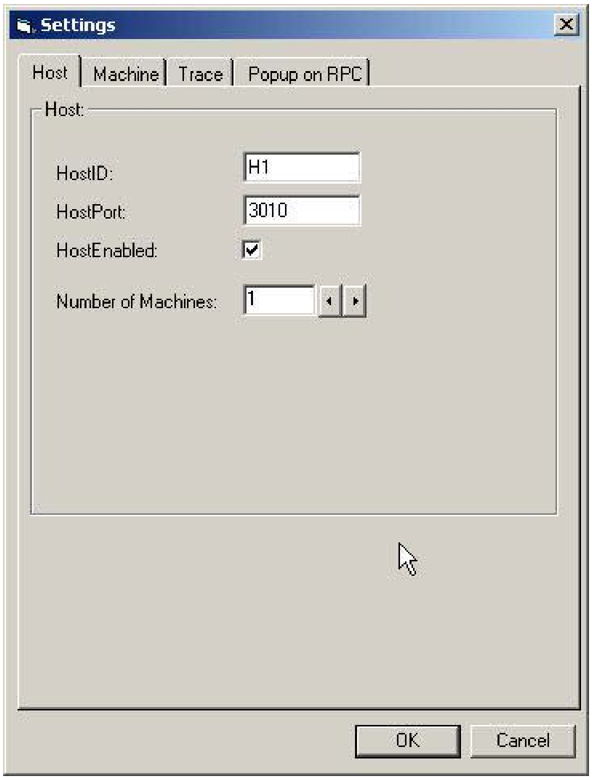
   Select "Start" -> "Programs" -> "RPC SINUMERIK" -> "RpcSinumerikTest".

2. The "RPC SINUMERIK Test program" window opens.
   Click on "Settings" in the menu bar.



3. The "Settings" window opens.
   Open the "Host" tab and enter the information.

4. Confirm your entries by clicking the "OK" button.



Host:

HostID: H1

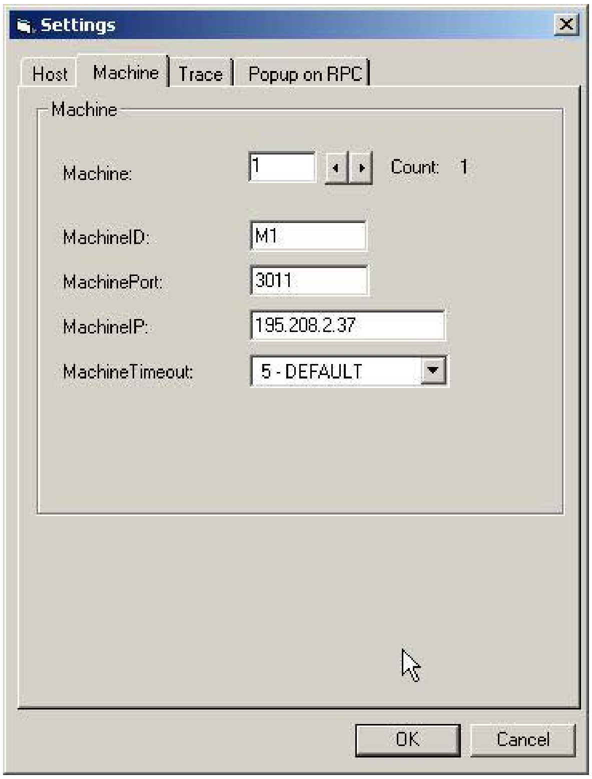
HostPort: 3010

HostEnabled: ☑

Number of Machines: 1

5. Open the "Machine" tab and enter the information.

   Confirm your entries by clicking the "OK" button.



Machine: 1 Count: 1

MachineID: M1

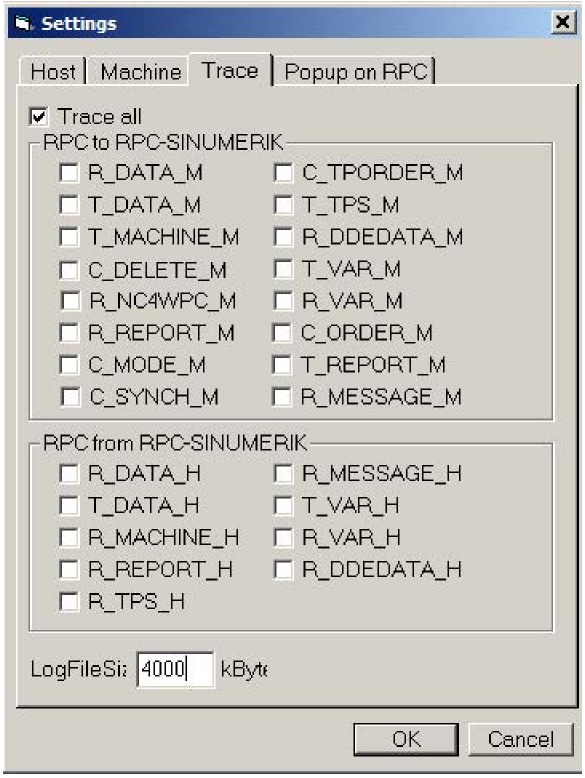
MachinePort: 3011

MachineIP: 195.208.2.37

MachineTimeout: 5 - DEFAULT

6. Open the "Trace" tab and enter the information.

   Confirm your entries by clicking the "OK" button.



☑ Trace all

RPC to RPC-SINUMERIK

☐ R_DATA_M
☐ T_DATA_M
☐ T_MACHINE_M
☐ C_DELETE_M
☐ R_NC4WPC_M
☐ R_REPORT_M
☐ C_MODE_M
☐ C_SYNCH_M
☐ C_TPORDER_M
☐ T_TPS_M
☐ R_DDEDATA_M
☐ T_VAR_M
☐ R_VAR_M
☐ C_ORDER_M
☐ T_REPORT_M
☐ R_MESSAGE_M

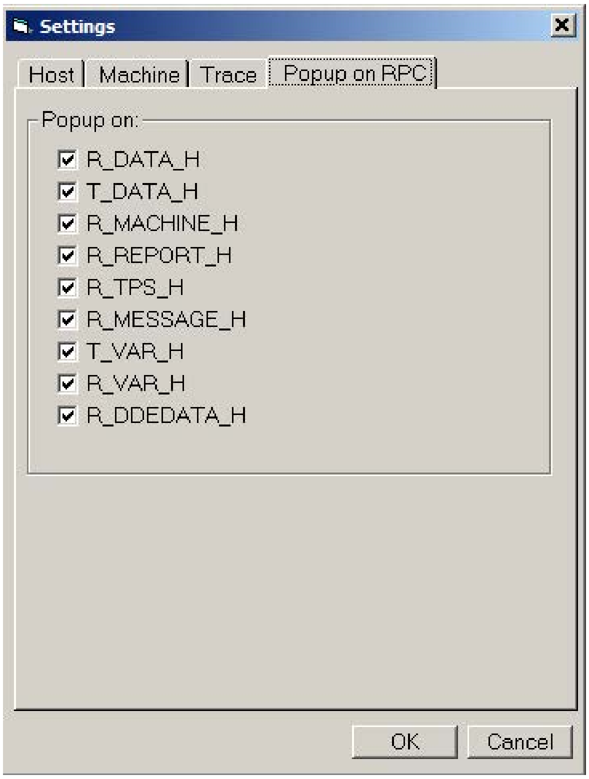
RPC from RPC-SINUMERIK

☐ R_DATA_H
☐ T_DATA_H
☐ R_MACHINE_H
☐ R_REPORT_H
☐ R_TPS_H
☐ R_MESSAGE_H
☐ T_VAR_H
☐ R_VAR_H
☐ R_DDEDATA_H

LogFileSi: 4000 kByte

7. Open the "Popup on RPC" tab and enter the information.

   Confirm your entries by clicking the "OK" button.



Popup on:

- ☑ R_DATA_H
- ☑ T_DATA_H
- ☑ R_MACHINE_H
- ☑ R_REPORT_H
- ☑ R_TPS_H
- ☑ R_MESSAGE_H
- ☑ T_VAR_H
- ☑ R_VAR_H
- ☑ R_DDEDATA_H

Operating modes and states 5

5.1 Operating modes

Overview

The SINUMERIK controller has the following operating modes: AUTO, MDA, JOG and TEACH IN. The computer link also requires its own operating modes. The functional behavior of RPC SINUMERIK, including its interaction with the PLC, depends on the computer link operating mode.

A separate dialog is offered on the controller for operator control and display of the computer link operating mode.

The computer link operating modes are:

- Unmanned host computer mode
- Manned host computer mode
- Manual mode
- Special mode
- Offline

Unmanned/manned host computer mode

In the two host computer modes (unmanned/manned), NC programs specified by the host computer may, in combination with automatic mode, be started from the PLC. The unmanned and manned host computer modes are distinguished to enable a different response to malfunctions.

If a malfunction occurs in manned production, the workpiece holder remains in the machine. The operator must eliminate the malfunction before the machining process can be continued.

If a malfunction occurs in unmanned production, it is possible to terminate machining of the workpiece holder in order to continue with another workpiece holder.

Manual mode

The NC programs are specified by the host computer. The NC program is selected using the computer link; the selected NC program is displayed in the header of the operator interface. The operator must start the program.

Note

Automatic material flow

In the unmanned/manned computer coupling FLR mode - and the manual mode - material flow on the machine is automatic. The transport system automatically delivers workpiece holders and retrieves them again after machining.

Special mode

In special mode, there is no automatic transport of workpiece holders, the host computer does not transfer program assignments and NC programs are not started automatically. The machine is controlled by the operator; otherwise, full communication takes place with the host computer. Special mode is typically used for testing NC programs or producing unplanned workpieces manually.

Offline

Offline indicates an interruption in the connection between the host computer and the controller. No communication takes place between the host computer and SINUMERIK. The controller switches to offline when it detects an interruption in the connection to the host computer. Likewise, the host computer designates a machine as offline in its status data and in the plant display as soon as it detects an interruption in the connection.

After a switch to offline, a synchronization of the machine must be carried out on the host computer. If an operating mode switchover – either a controller or computer link operating mode – occurs on the machine, this must be reported to the host computer with the RPC call `R_MACHINE_H()`.

5.2 States

Status: Machine

The machine can assume the following states:

- Restart: After a restart of the SINUMERIK controller
- Inactive: No machining in progress
- Active: Machining in progress
- Fault: Machining interrupted
- Components switched off

Status: Docking positions

The host computer must recognize the states of the docking positions so that it can issue transport orders on the transport system. The following states are possible:

- Enabled
- Disabled for TPS
- Fault

Status: Workpiece holder

The host computer needs the workpiece holder states for determination of transport orders.

The workpiece holder can assume the following states:

- Not machined - no program assignment: An NC program has not yet been assigned for machining, i.e. machining is not yet possible.
- Not machined - program assigned:
  - In progress
  - Finished
  - Finished with error
  - Only for buffering:
    The workpiece holder will only be buffered on the machine, no machining will take place.

Tool data 6

Tool data

Because not all data of a tool is always needed, three versions are available.

- You can preset three data set versions.

  These can be set to match the particular application.
- Only complete areas can be configured as the data scope for each tool. This means that for each version there is a list (for RPC in the registry and for CMI in file "mcisrpcsl.ini), in which the areas are listed, that are transferred for this version. The file format for the tool data corresponds to the data backup format of the NC840D (punched tape/ASCII format acc. to /BA/ ; such as in `_N_TOx_TOA` or `_N_TOx_INI`).

Tool data areas

The tool data is stored in different areas in the NC.

The areas are designated as follows:

Table 6- 1 Tool data areas

Area	Tool identifier
General tool data	$TC_TPx[y]
User-related tool data (OEM data)	$TC_TPCx[y]
Cutting edge data	$TC_DPx[y,z]
User-related tool cutting data (OEM data)	$TC_DPCx[y,z]
Tool monitoring data	$TC_MOPx[y,z]
User-related tool monitoring data (OEM data)	$TC_MOPCx[y,z]

x: A running value for each area that is used to produce a unique name for the system variable.

y: T number

z: Cutting edge number

Magazine location data area

Table 6- 2 Magazine location data areas

Area	Tool identifier
Magazine and location number	$TC_MPP6[y,z]=x

x: T number

y: Magazine number

z: Location number

Tool and magazine location data

Table 6- 3 Tool and magazine location data

Tool identifier	Description
$TC_TP1[3]= 2	Duplo number
$TC_TP2[3]= "4711"	ID number
$TC_TP3[3]=1	Size to left in half locations
$TC_TP4[3]=1	Size to right in half locations
$TC_TP5[3]=1	Size upwards in half locations
$TC_TP6[3]=1	Size downwards in half locations
$TC_TP7[3]=2	Magazine location type
$TC_TP8[3]=131	Status
$TC_TP9[3]=0	Tool monitoring method
$TC_TP10[3]=2	Tool replacement strategy
$TC_TP11[3]=0	Tool information
$TC_DP1[3,1]=0	Edge parameter 1
$TC_DP2[3,1]=0	Edge parameter 2
$TC_DP3[3,1]=0	Edge parameter 3
$TC_DP ...	...
$TC_DP24[3,1]=0	Edge parameter 24
$TC_DP25[3,1]=0	Edge parameter 25
*$TC_MPP6[1,5]=3	T number

* $TC_MPP6 indicates which tool (T number) is in the stated magazine and location. In this case, the tool with T number 3 is in magazine 1 in location 5.

Reading out TC_MOP parameters (single tool)

In the configuration program, you can specify that the cutting edge-related tool monitoring data is to be output when a single tool is read out. A single tool is then read out. The `TC_MOP11` parameter holds the target tool life and the `TC_MOP13` parameter holds the target workpiece count. If you open the read out file, however, these two target values are contained in parameters `TC_MOP5` and `TC_MOP6`.

These two parameters are intended for the actual and target wear values.

Solution:

Table 6- 4 Monitoring data for each tool cutting edge

Description	Parameter	Software version
P1 = Prewarning limit for tool life in minutes	`$TC_MOP1`	
P2 = Remaining tool life in minutes	`$TC_MOP2`	
P3 = Prewarning limit for workpiece count	`$TC_MOP3`	
P4 = Remaining workpiece count	`$TC_MOP4`	
P5 = Target tool life	`$TC_MOP11`	
P6 = Target workpiece count	`$TC_MOP13`	
P7 = Prewarning limit for wear	`$TC_MOP5`	from SW 5.1
P8 = Remaining wear (actual value)	`$TC_MOP6`	from SW 5.1
P9 = Target wear value	`$TC_MOP15`	from SW 5.1

The following applies: In general, RPC returns a consecutive list of 9 parameters in `TC_MOP`.

There are 9 parameters, but they are called `MOP1-6`, `MOP11`, `MOP13` and `MOP15`.

Communication between the host computer and SINUMERIK

7

7.1 Remote Procedure Call

Communication between the host computer and the SINUMERIK controller makes use of the concept of remote procedure calls (RPC) for the transfer of smaller volumes of data. During communication via RPC, the communication partner is requested to execute a function identified by the procedure name with the parameters (data) contained in the call.

You can find a information the Interface Definition Language IDL (program-internal specification) of the calls used in the following in the appendix in the following section: Interface Definition Language (IDL) (Page 279)

For the procedures described in the following, the names are structured according to the following concept:

Structure of procedure names

The procedure names have three components:

1. Command identifier (first character)
2. Data/function identifier
3. Receiver identifier (last character)

Command identifier

The command identifier appears in the first name character:

C	Command call	Command
R	Request to receive data	Receive
T	Request to transmit data	Transmit

Example:

```
T_MACHINE_M()
```

Data/function identifier

The identifier designates the type of data that is being requested or transferred or the function for which the data is intended.

Examples:

```
T_MACHINE_M(),
```

```
R_NC4WPC_M()
```

Receiver identifier

The receiver identifier designates the addresses that is to execute the function:

H	Receiver is the host computer	(Host)
M	Receiver is the machine	(Machine)

Parameters

Host

Identifier for the host computer with max. 16 characters. When several machines with multiple host computers are used, Host uniquely identifies the host computer that data is to be exchanged with.

Machine

Identifier for a machine (max. 16 characters). All existing machines must be uniquely identifiable in the network.

OrderNum

Order number: The number can be optionally used if RPC requests and their responses have to be uniquely assigned to one another.

Hotlinks

Hotlinks are always signaled with OrderNum "0".

Note

Parameters of the "String" type

With parameters of the "String" type, the string must be delimited by "\0".

In Visual C++, "\0" and "" represent an empty string; in Visual Basic, "" produces an empty string.

The permissible maximum string lengths are given for the individual parameters.

7.2 Communication processes

Requirement

For error-free communication between the host computer and one or more machines, the host computer must know the communication partner from which it is to process RPCs. The machine names of these machines (clients) must be stored in the host computer.

Parameters

Parameters are data that is transferred as a component of the RPC. Information on the meaning of the parameters for the host computer and machine is given in a table block that is written with each RPC.

Acknowledgment

As with local procedures, the return value is used for positive or negative acknowledgment of RPCs. For requests that are processed asynchronously, this acknowledgment can only acknowledge the receipt of the request. After completion of processing or when an error occurs during processing, the communication partner must be informed with a corresponding RPC message.

Error messages are displayed on the controller with the help of the alarm server. In the case of error-free calls, return value = 0.

In the case of an error, you can obtain a list of the return values in the Appendix in the following Chapter: Error numbers (Page 348)

Note

Host computer software

The host computer software must return the called functions as quickly as possible because the part of RPC SINUMERIK that sends the RPC waits during this call and, thus, no further processing occurs. Within the function called in the host computer, the data contained in the RPC should be copied to a buffer area and the function should then be returned.

Only afterwards should the actual processing in the host computer take place.

7.3 Machine status data

7.3.1 Sending machine status data

Function

Called function

```
R_MACHINE_H(          Host,
                      Machine,
                      OrderNum,
                      MachineMode,
                      MachineStatus,
                      NCProgramm,
                      ClampCubeSide,
                      DockPos,
                      DockPosStatus,
                      WPC,
                      WPCStatus,
                      Resint1,
                      Resint2,
                      Resbyte)
```

Direction of transfer: SINUMERIK → Host computer

Meaning

Send machine status data to the host computer.

Data

Table 7- 1 Parameters of the machine status data transfer

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)

Parameter	Description	Format
MachineMode	Operating mode • SINUMERIK operating mode: – 1: Automatic – 2: MDA – 4: JOG – 8: TEACH IN • Computer link operating mode: – 100: Unmanned host computer mode – 200: Manned host computer mode – 300: Manual mode – 400: Special mode – 500: When bit 6=1 in the RPC SINUMERIK mode – 600: When bit 7=1 in the RPC SINUMERIK mode • PLC operation – 1000: no DB12 ****	4 bytes (long int)
MachineStatus	Machine status 0: Restart 1: Inactive 2: Active 3: Fault 4: Components switched off	4 bytes (long int)
NCProgramm	Current NC program*	128 bytes (string)
ClampCubeSide	Side of clamp cube**	4 bytes (long int)
DockPos[3]	Docking position number The docking position number corresponds to the index in the docking position list of the interface DB beginning with 1. Docking position number = 0 means "not occupied", see also Chapter: AUTOHOTSPOT	3 x 4 bytes (long int)
DockPosStatus[3]	Docking position status 0: Enabled 1: Disabled for TPS 2: Fault	3 x 4 bytes (long int)
WPCStatus[3]	Workpiece holder status 1: Not machined, no program assignment 2: Not machined, program assigned 4: Prepare program selection 8: Program selection complete 16: In progress 32: Finished 64: Finished with error 128: For testing	3 x 4 bytes (long int)

Parameter	Description	Format
Resint1***	Reserve 1	4 bytes (long int)
Resint2***	Reserve 2	4 bytes (long int)
Resbyte****	Reserve 3	8 bytes (string)

* NC program identifier of the currently running NC program

** Side (of clamp cube) that is currently being processed

*** Resint1 and 2 are placed on the DB interface of the PLC, values are entered there by the PLC, they are transferred to the host computer. These values are not relevant for the computer link, they are only transferred to the host computer.

****Resbyte: The channel number is displayed here.
In RPC from version 2.0 SP2 HF4
Available in CMI from version 2.6 SP1 HF4

***** Available from RPC 02.00.02.03 and higher

Notes on use

- The SINUMERIK controller must issue this RPC at each status change on the machine. For this purpose, the RPC SINUMERIK computer link software determines the current data and issues the call.
- The action is triggered by the PLC by the setting of a certain bit on the DB interface.
- The host computer can also trigger this action with the `T_MACHINE_M` command (Request machine status data, see below).

Note

- If more than three docking positions must be described, a separate machine assignment file must be transferred. See also section Overview (Page 122)
- In order to report the operating modes of both parties (SIN and host computer) in one variable (MachineMode), the sum can be transferred as a value (e.g. 201: Manned host computer mode =200 and SIN Automatic =1).
- The RPC SINUMERIK computer link software does not perform a plausibility check on the reported operating modes.

7.3.2 Request machine status data

Function

Called function:

T_MACHINE_H(Host,
Machine,
OrderNum)

Direction of transfer: Host computer → SINUMERIK

Meaning

Table 7- 2 Parameters of the machine status data request

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)

Note on use

The host computer can use this RPC call to request the machine status data, for example, for synchronization. The controller then sends the data to the host computer with the `R_MACHINE_H()` command.

Example

```
T_MACHINE_M("FLR1","BAZ3",0);
```

7.4 Production dialog

Overview

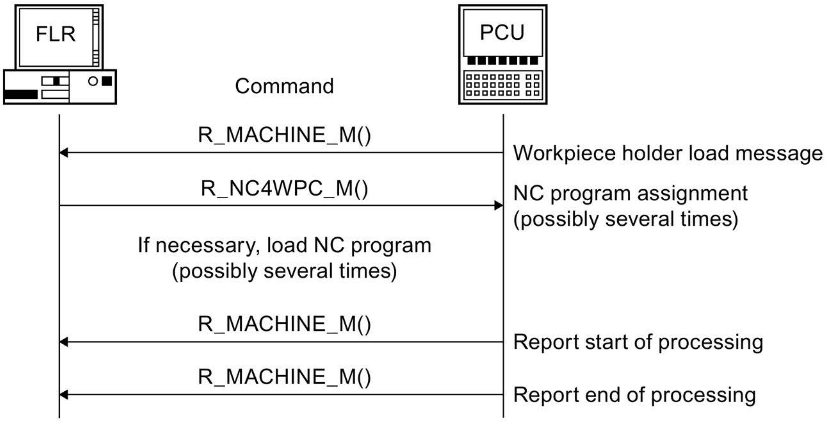
Based on an example, you will see in the following how the host computer and the PLC have to interact within the framework of this dialog?

Once the workpiece holder arrives in the machine, the SINUMERIK controller sends the machine status data to the host computer, initiated by the PLC. Based on the data, the host computer recognizes which workpiece holder has arrived in the machine and transfers the program assignment for this workpiece holder.

For workpiece holders with clamp cube, one NC program is assigned for each side. RPC SINUMERIK saves these program assignments.

Each program assignment consists of workpiece holder, side and NC program. The next NC program is transferred and selected in each case. Afterwards, the PLC may start the NC program (in manned and unmanned host computer mode). The machine reports the NC start and later the NC end with the machine status data.

Below you see a normal error-free sequence:



If an error occurs during a sequence, corresponding messages are displayed, see also section: Message from SINUMERIK to the host computer (Page 94).

Program assignment

Called function:

```
R_NC4WPC_M(           Host,
                      Machine,
                      OrderNum,
                      WPC,
                      NCProg,
                      Date,
                      NCPLength,
                      ClampCubeSide,
                      TpFlag,
                      NCExtern,
                      Resint1,
                      Resint2,
                      Resbyte)
```

Direction of transfer: Host computer > SINUMERIK

Meaning

Instruct the machine which program is to be activated.

Data

Table 7- 3 Parameters of program assignment

<table>
<tr><th>Parameter</th><th colspan="3">Description</th><th>Format</th></tr>
<tr><td>Host</td><td colspan="3">Name of the host computer</td><td>16 bytes (string)</td></tr>
<tr><td>Machine</td><td colspan="3">Machine name</td><td>16 bytes (string)</td></tr>
<tr><td>OrderNum</td><td colspan="3">Order number</td><td>4 bytes (long int)</td></tr>
<tr><td>WPC</td><td colspan="3">Workpiece holder designation</td><td>6 bytes (string)</td></tr>
<tr><td>NCProg</td><td colspan="3">NC program
NC program in the form:
"\mpf.dir\Zylinderkopf.mpf"</td><td>128 bytes (string)</td></tr>
<tr><td>Date</td><td colspan="3">Date of the last change to the NC program (Unix time)</td><td>4 bytes (long int)</td></tr>
<tr><td>NCPLength</td><td colspan="3">NC program size in bytes</td><td>4 bytes (long int)</td></tr>
<tr><td>ClampCubeSide</td><td colspan="3">Side of clamp cube</td><td>4 bytes (long int)</td></tr>
<tr><td>TpFlag</td><td colspan="3">Transport flag
= 0: No follow-up operation
= 1: Follow-up operation
= 9: Only for buffering</td><td>4 bytes (long int)</td></tr>
<tr><td rowspan="6">NCExtern</td><td colspan="3">Process from external source</td><td rowspan="6">4 bytes (long int)</td></tr>
<tr><td>Bit 1</td><td>Bit 2</td><td>Function</td></tr>
<tr><td>0</td><td>0</td><td>Process NC program on NC</td></tr>
<tr><td>0</td><td>1</td><td>Process NC program from external source (decimal 1)</td></tr>
<tr><td>1</td><td>0</td><td>Process workpiece on NC (decimal 2)</td></tr>
<tr><td>1</td><td>1</td><td>Process workpiece from external (decimal 3)</td></tr>
</table>

Parameter	Description	Format
Resint1	Reserve 1	4 bytes (long int)
Resint2	Reserve 2	4 bytes (long int)
Resbyte	Reserve 3	8 bytes (string)

Notes on use

- This RPC may occur multiple times for a workpiece holder, if each side of a clamp cube is machined with a separate NC program. The sides are machined in the order that the program assignments are reported from the host computer to RPC SINUMERIK.
- For all program assignments of a workpiece holder except the last one, the transport flag "1 = Follow-up operation" must be set. If the follow-up operation flag is set for a side, the workpiece holder remains at the machining station after the end of the NC program. Because this flag is not set for the last side, it is possible to recognize from this that no more machining will follow and the workpiece holder can be moved out of the machining station.
- If a workpiece holder is located in the machine only for buffering purposes, this can be reported with transport flag "9 = Only for buffering". In this case, an NC program is not specified.

Example

```
R_NC4WPC_M("FLR1", "BAZ3", 0, "WPC05", "\mpf.dir\Kw15.mpf", 862826400, 3210, 1, 0, 0,
0, 0, "\0");
```

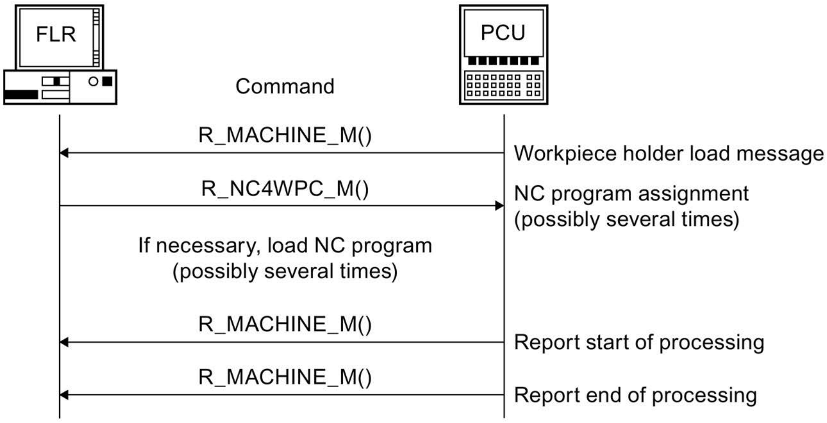
7.5 Application example: Production dialog

Overview

Based on an example, you will see in the following how the host computer and the PLC have to interact within the framework of this dialog?

Once the workpiece holder arrives in the machine, the SINUMERIK controller sends the machine status data to the host computer, initiated by the PLC. Based on the data, the host computer recognizes which workpiece holder has arrived in the machine and transfers the program assignment for this workpiece holder.

Each program assignment consists of workpiece holder, side and NC program. The next NC program is transferred and selected in each case. Afterwards, the PLC may start the NC program (in manned and unmanned host computer mode). The machine reports the NC start and later the NC end with the machine status data.

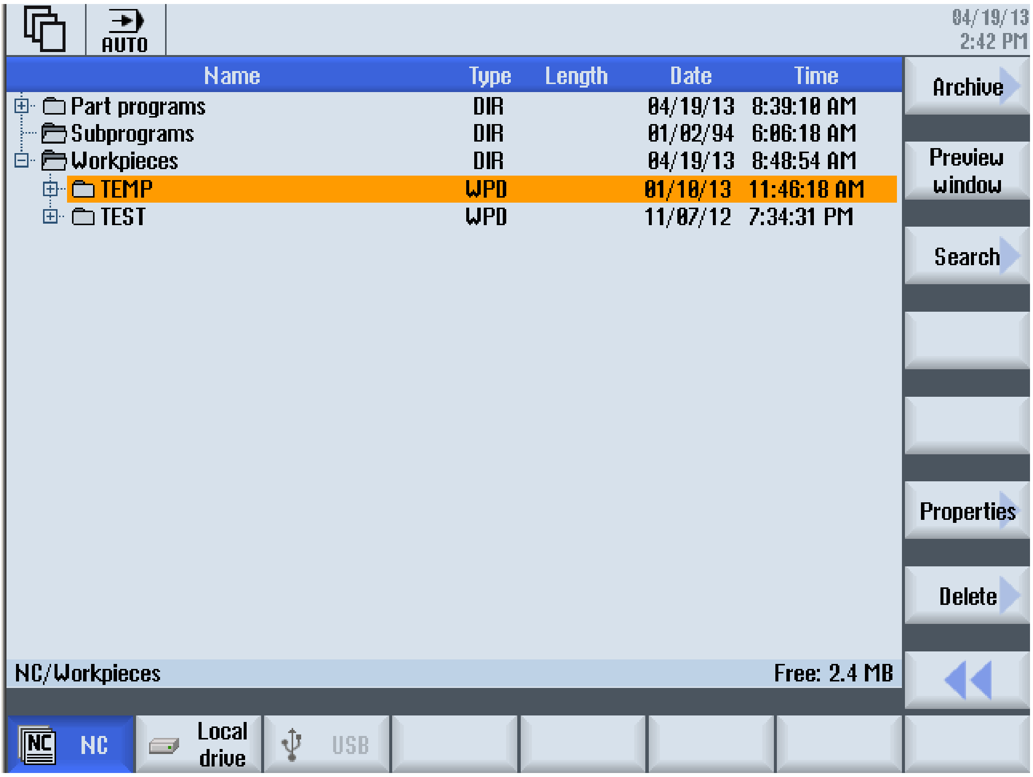


- NC program used: Prog_1.mpf
- Name of the workpiece holder: AAAA

Below you see the needed bytes in DB12:

DB12	
DBB0	Bit 0: set to "1" by the PLC if the workpiece holder status is to be changed by the PLC.
DBB1	Change trigger, e.g. 1 > 2 > 4 > 8 etc., change by the PCL and from RPC
DBB21	Workpiece holder status
DBB22-27	Workpiece holder, flush left

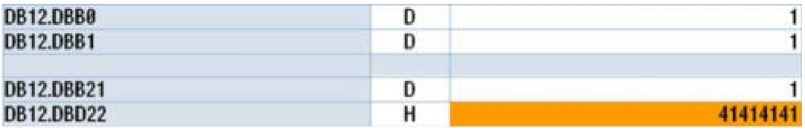
1. The NC program "1.mpf" is not yet available on the controller:



Name	Type	Length	Date	Time
Part programs	DIR		04/19/13	8:39:10 AM
Subprograms	DIR		01/02/94	6:06:10 AM
Workpieces	DIR		04/19/13	8:40:54 AM
TEMP	WPD		01/10/13	11:46:10 AM
TEST	WPD		11/07/12	7:34:31 PM

NC/Workpieces Free: 2.4 MB

2. PLC sets the WPC status to 1, pallet arrival:
   RPC then sends `R_MACHINE_H` to the host computer.



DB12.DBB0	D	1
DB12.DBB1	D	1
DB12.DBB21	D	1
DB12.DBD22	H	41414141

3. RPC then sends `R_MACHINE_H` to the host computer and acknowledges DB12.DBB0:
   08.03.2013 09:26:12 R_MACHINE_H

```
OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
>"" ClampCubeSide->0 DockPos1->1 DockPos2->0 DockPos3->0 DockPosStatus1->0
DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA" WPC2->"" WPC3->"" WPCStatus1->1
WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0 ResByte->""
```

4. The host computer makes the program assignment with `R_NC4WPC_M`:
   08.03.2013 09:26:19 R_NC4WPC_M Ret=0

```
OrderNum->12005 Machine->"M1" Host->"H1" WPC->"AAAA" NCProg-
>"\wks.dir\RPCTEST.wpd\Prog_1.MPF" DateVal->0 NCLength->0 ClampCubeSide->0
TpFlage->0 NCExtern->0 ResInt1->0 ResInt2->0 ResByte->"0"
```

5. RPC determines that the program PROG_1.MPF is needed and calls for it on the host computer with `T_DATA_H` an:
   08.03.2013 09:26:19 T_DATA_H

```
OrderNum->0 Machine->"M1" Host->"H1" SFkt->1 Name1-
>"\wks.dir\RPCTEST.wpd\Prog_1.MPF" Name2->""
```

6. RPC changes the WPC status from 1 to 2:
   08.03.2013 09:26:19 R_MACHINE_H

```
OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
>"" ClampCubeSide->0 DockPos1->1 DockPos2->0 DockPos3->0 DockPosStatus1->0
DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA" WPC2->"" WPC3->"" WPCStatus1->2
WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0 ResByte->""
```

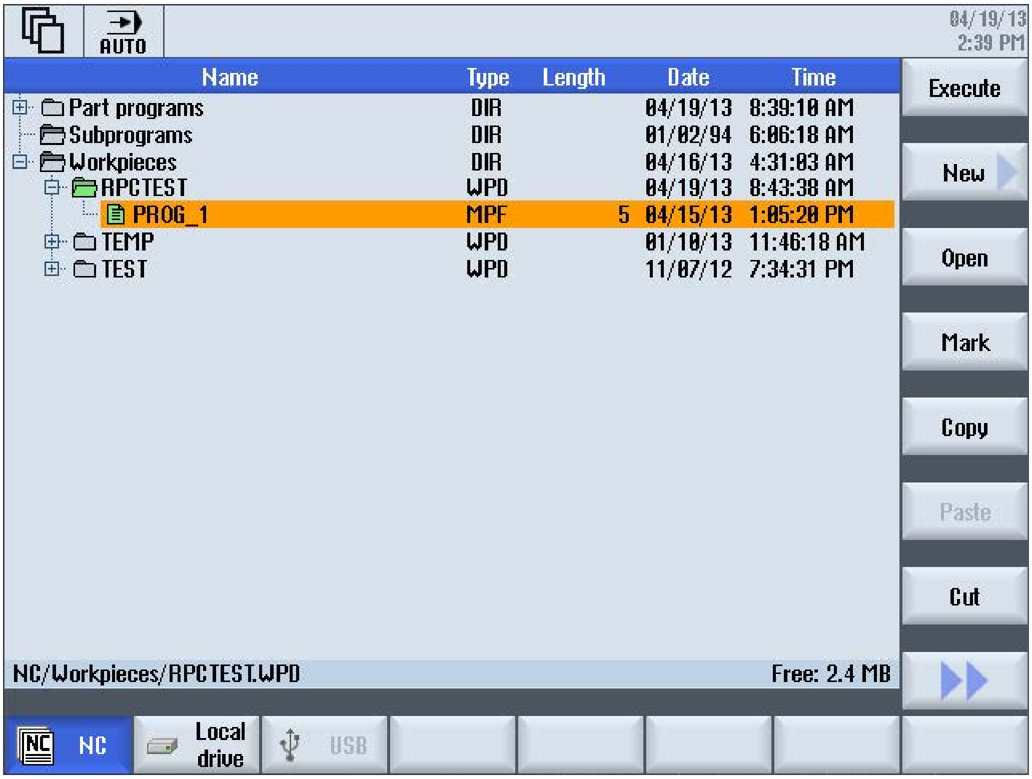
7. The host computer transfers the NC program to the controller:
   08.03.2013 09:26:31 R_DATA_M Ret=0

```
OrderNum->1111 Machine->"M1" Host->"H1" SFkt->1 Name1-
>"\wks.dir\RPCTEST.wpd\Prog_1.MPF" Name2->"V:\putdir\program.MPF" Date->0
LastFile->1
```

8. RPC has received the NC program (Type 5):
   08.03.2013 09:26:31 R_REPORT_H

```
OrderNum->1111 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 1 0 0 0 0 0 0
0 0 0 Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0
0 ResInt1->0 ResInt2->0 ResByte->""
```

9. The NC program is now available and loaded on the controller.

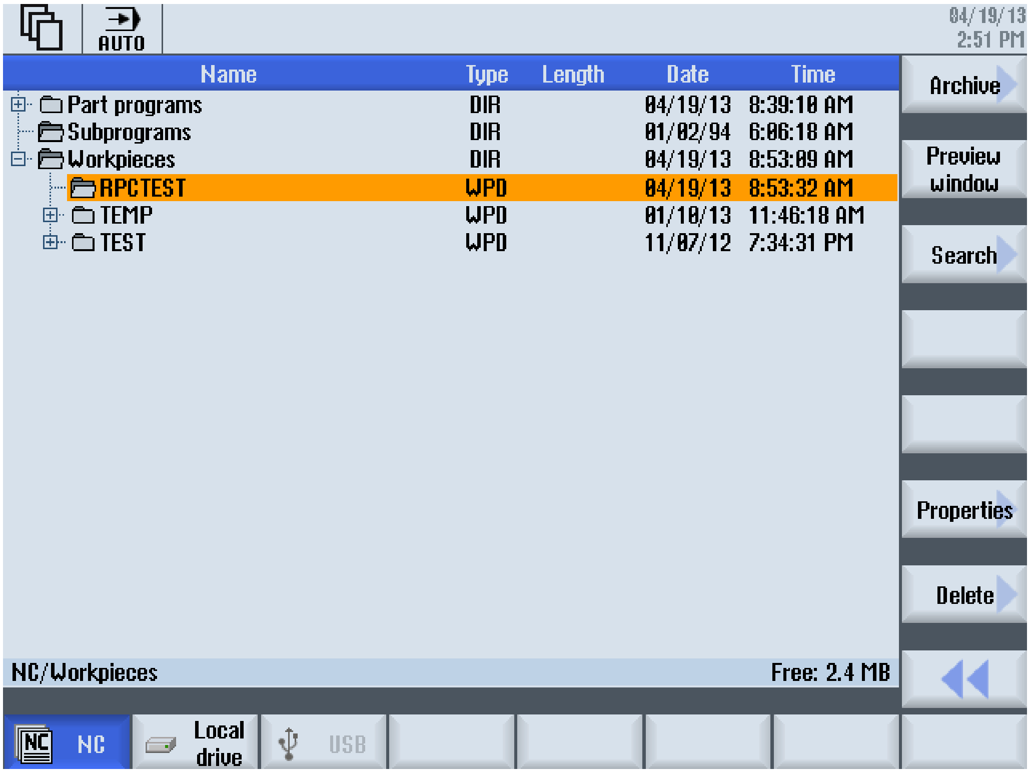


Name	Type	Length	Date	Time
Part programs	DIR		04/19/13	8:39:10 AM
Subprograms	DIR		01/02/94	6:06:18 AM
Workpieces	DIR		04/16/13	4:31:03 AM
RPCTEST	WPD		04/19/13	8:43:38 AM
PROG_1	MPF	5	04/15/13	1:05:20 PM
TEMP	WPD		01/10/13	11:46:18 AM
TEST	WPD		11/07/12	7:34:31 PM

NC/Workpieces/RPCTEST.WPD Free: 2.4 MB

10. PLC sets WPC status 4:
    08.03.2013 09:26:56 R_MACHINE_H
    ```
    OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
    >"" ClampCubeSide->0 DockPos1->1 DockPos2->0 DockPos3->0 DockPosStatus1->0
    DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA" WPC2->"" WPC3->"" WPCStatus1->4
    WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0 ResByte->""
    ```
11. RPC selects the NC program and changes the WPC status to 8:
    08.03.2013 09:26:57 R_MACHINE_H
    ```
    OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
    >"\wks.dir\RPCTEST.wpd\Prog_1.MPF" ClampCubeSide->0 DockPos1->1 DockPos2->0
    DockPos3->0 DockPosStatus1->0 DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA"
    WPC2->"" WPC3->"" WPCStatus1->8 WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0
    ResByte->""
    ```
12. PLC starts the NC program and changes the WPC status to 16 (In progress):
    08.03.2013 09:27:43 R_MACHINE_H
    ```
    OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
    >"\wks.dir\RPCTEST.wpd\Prog_1.MPF" ClampCubeSide->0 DockPos1->1 DockPos2->0
    DockPos3->0 DockPosStatus1->0 DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA"
    WPC2->"" WPC3->"" WPCStatus1->16 WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0
    ResByte->""
    ```
13. After the end of the NC program, PLC sets WPC status 32 (Finished):
    08.03.2013 09:27:59 R_MACHINE_H
    ```
    OrderNum->0 Machine->"M1" Host->"H1" MachineMode->400 MachineStatus->1 NCProgramm-
    >"\wks.dir\RPCTEST.wpd\Prog_1.MPF" ClampCubeSide->0 DockPos1->1 DockPos2->0
    DockPos3->0 DockPosStatus1->0 DockPosStatus2->0 DockPosStatus3->0 WPC1->"AAAA"
    WPC2->"" WPC3->"" WPCStatus1->32 WPCStatus2->0 WPCStatus3->0 ResInt1->0 ResInt2->0
    ResByte->""
    ```

14. The NC program Progr_1.mpf is deleted by RPC.



Name	Type	Length	Date	Time
Part programs	DIR		04/19/13	8:39:10 AM
Subprograms	DIR		01/02/94	6:06:18 AM
Workpieces	DIR		04/19/13	8:53:09 AM
RPCTEST	WPD		04/19/13	8:53:32 AM
TEMP	WPD		01/10/13	11:46:18 AM
TEST	WPD		11/07/12	7:34:31 PM

NC/Workpieces   Free: 2.4 MB

15. The production dialog is now complete. The workpiece holder status remains set to 32. The next production then begins again with the message of the workpiece holder (WPC status = 1).
    The production dialog can be canceled by the PLC with workpiece holder status 64.

7.6 Messages

7.6.1 Message from SINUMERIK to the host computer

Function

Called function

```
R_REPORT_H(            Host,
                       Machine,
                       OrderNum,
                       Typ,
                       Number,
                       Time,
                       Flag,
                       Resint1,
                       Resint2,
                       Resbyte)
```

Direction of transfer: SINUMERIK → Host computer

Meaning

Deliver message to the host computer.

Data

Table 7- 4 Parameters of the message from SINUMERIK to host computer

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
Type	Message type 1: Alarms 2: Operator interruption 3: Operating messages 4: Error message of the RPC SINUMERIK computer link software 5: Positive acknowledgment for `R_DATA_M()`, `R_DDEDATA_M()` and `R_VAR_M()` 6: Positive acknowledgment for C_ORDER_M()	4 bytes (long int)
Number[10]	Message number field • Field for up to 10 alarms. • Unneeded fields must be assigned 0 • For other message types, only Number[0] is assigned	10 x 4 bytes (long int)

Parameter	Description	Format
Time[10]	Time stamp field - Field for up to 10 entries. - Unneeded fields must be assigned 0	10 x 4 bytes (long int)
Flag[10]	Incoming/outgoing flag, field for up to 10 entries C: Incoming message, no machine standstill S: Incoming message, machine standstill G: Outgoing message L: All messages have gone out For incoming messages, the following distinction must be made: results in machine standstill/does not result in machine standstill.	10 x 1 byte
Resint1	Reserve 1	4 bytes (long int)
Resint2	Reserve 2	4 bytes (long int)
Resbyte	Reserve 3	8 bytes (string)

With `R_DATA_M()`, `R_DDEDATA_M()`, `R_VAR_H()` and `C_ORDER_M()` processing cannot occur synchronously during the call. For this reason, RPC SINUMERIK must send a positive acknowledgment to the host computer once processing is complete in order to inform the host computer about the completion. In the case of a positive acknowledgment to `R_DATA_M`, the SFkt of `R_DATA_M` is used as the "error number" and serves to assign the acknowledgment. In the case of a positive acknowledgment to `C_ORDER_M()`, the SFkt is also entered as the error number, even though the type is 6. For `R_DDEDATA_M`, 1000 is reported as the "error number" and for `R_VAR_M`, 0 is reported as the "error number".

Note

Same index

Number, Time and Flag with the same index belong together.

Notes on use

The RPC `R_REPORT_H()` is used to transfer individual messages or a group of up to 10 alarm messages to the host computer.

Special case: When the last alarm has gone out on the machine, this status is sent to the host computer with `R_REPORT_H()` and the following parameters:

- Type = 1
- Number[0] = 0
- Flag[0] = L

Message filter

RPC SINUMERIK has an integrated message filter that enables the forwarding of certain messages or alarms to the host computer to be suppressed. It is possible to completely suppress alarm types of the alarm server and to suppress ranges of alarm numbers.

The "SCALARMEX.INI" file in the ADD_ON directory is used to configure this. The `[AlarmTypeExclusion]` section is used to exclude alarm types. The `[AlarmRangeExclusion]` section is used to exclude alarm number ranges. The keywords (Type1 to TypeN, or Range1 to RangeN) must be specified without gaps. The read function reads in ascending order and stops at the first non-existing number.

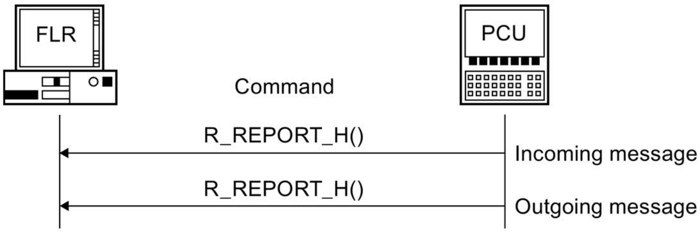
The structure of the "SCALARMEX.INI" file:

```
[AlarmTypeExclusion]

Type1=6

Type2=7

Type3=8

Type4=9

Type5=10

Type6=11

[AlarmRangeExclusion

Range1=100,199

Range2=250,250

Range3=2000,3999

Range4=5000,5050

Range5=6799,6799
```

Sequence for alarms/operator interruptions/operating messages

If machining continues on the machine after the end of alarms or operator interruptions with machine standstill, this must be reported to the host computer with the RPC `R_MACHINE_H()` together with the machine status (on the transport system corresponding with `R_TPS_H()`).

A list of the error messages for message type 4 is contained in the appendix in the following section: Error numbers (Page 348).



Type of the alarm	= 1	POWER ON
	= 2	RESET
	= 3	CANCEL
	= 4	BYNCK
	= 5	NCSTART
	= 6	BAGRESET
	= 7	NCRESET
	= 8	PLCMSG
	= 9	PLC

7.6.2 Message from host computer to SINUMERIK

Function

Called function

```
R_REPORT_M(             Host,
                        Machine,
                        OrderNum,
                        Typ,
                        Number,
                        Resint1,
                        Resint2,
                        Resbyte)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

Error messages of the host computer are sent to RPC SINUMERIK in the controller.

Data

Table 7- 5 Table - Parameters of the message from host computer to SINUMERIK

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
Type	Message type 4: Error message of the host computer to SINUMERIK controller	4 bytes (long int)
Number	Error number	4 bytes (long int)
Time[10]	Time stamp field • Field for up to 10 entries. • Unneeded fields must be assigned 0	10 x 4 bytes (long int)
Resint1	Reserve 1	4 bytes (long int)
Resint2	Reserve 2	4 bytes (long int)
Resbyte	Reserve 3	8 bytes (string)

Notes on use

The error messages are entered in a LOG file on the SINUMERIC controller. In addition, certain error messages trigger error handling in RPC SINUMERIK.

Example

```
R_REPORT_M("FLR1", "BAZ3", 0, 4, -13, 0, 0, "\0");
```

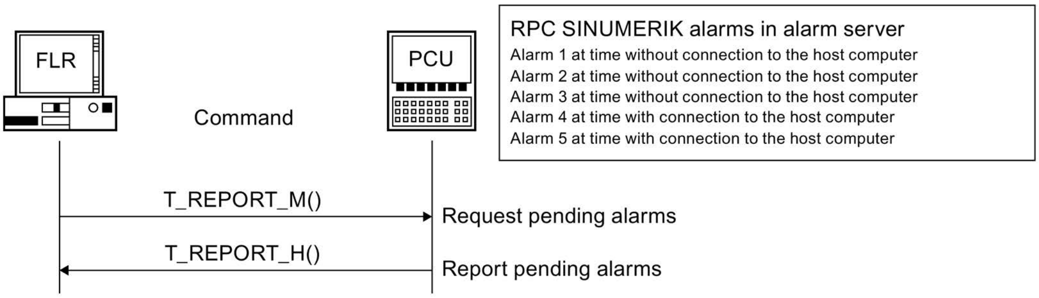
Note

R_REPORT

The RPC call `R_REPORT_M` is not available for CMI.

.

Request pending alarms


FLR
PCU
Command
RPC SINUMERIK alarms in alarm server
Alarm 1 at time without connection to the host computer
Alarm 2 at time without connection to the host computer
Alarm 3 at time without connection to the host computer
Alarm 4 at time with connection to the host computer
Alarm 5 at time with connection to the host computer
T_REPORT_M()
Request pending alarms
T_REPORT_H()
Report pending alarms

Called function

```
T_REPORT_M(             Host,
                        Machine,
                        OrderNum)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

With `T_REPORT_M`, pending alarms and messages in the alarm server can be requested taking the message filter into account. All pending alarms and messages of the alarm server are requested regardless of whether these originated at a time with or without connection to the host computer.

The function therefore allows synchronization for the alarms and messages between the host computer and SINUMERIK.

Otherwise, only alarms that originated at a time with connection to the host computer are forwarded to the host computer.

The alarms and messages requested with `T_REPORT_M` are reported with `R_REPORT_H`.

Data

Table 7- 6 Parameters of the alarm request

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)

7.7 Data dialogs

7.7.1 Overview

Larger data volumes such as NC programs or tool data are transferred as a file. Because the SINUMERIK can only be an FTP client when the File Transfer Protocol is used, the file must always be transferred from RPC SINUMERIK in SINUMERIK.

The host computer uses RPC call `R_DATA_M` to inform the RPC SINUMERIK computer link software in SINUMERIK that a file is ready for processing. The computer link software then fetches the file and processes it. In the other direction, RPC SINUMERIK in SINUMERIK transfers the file and uses RPC call `R_DATA_H` to inform the host computer that a file is ready for processing on the host computer.

Data dialogs relate to the PUT_/GET directories configured with the configuration program "SCCONFIG". Path names must always be specified from the "perspective" of RPC SINUMERIK. Both shared drive names (shares) and UNC notation can be used. When the files are located in the configured directories, the file name can be specified without path in the "Name2" parameter. The path information for the host computer depends on the possibilities of the FTP server.

Program transfer

NC programs are transferred as files. NC programs must be entered in the NC data management after their transfer.

Identification of file contents

The data in the `R_DATA_M()` and `R_DATA_H()` calls includes a subfunction number SFkt, which contains the type of the provided data and, thus, information the data has been correctly entered. A file with tool data must be read tool-by-tool and processed according to the specified subfunction number. After the processing function finishes processing the files, they must be deleted. All old received files must be deleted at the start of the computer link.

The subfunction numbers are the same for the following functions:

- Request
- Accept
- Delete

Table 7- 7 Subfunction numbers: SFkt for data dialogs

SFkt	Function	Remark
1	NC program	Name1 = NC program Name2 = File name with path on the host computer
7	Transfer NC program Function EES, **Only available for CMI**	Parameter required, see SFkt=1
7	Accept transferred data without overwriting existing data **Only available for RPC**	Parameter required, see SFkt=1
8	Transfer NC program to/from PCU 50 / Industrial PC. **Only available for CMI**	Parameter required, see SFkt=1
10	List of existing NC programs	Name1 = Data storage path Name2 = Name of list file 2
14	NC programs in the EES List memory, **Only available for CMI**	Parameter required, see SFkt=10
15	List NC programs on the CF card **Only available for CMI**	Parameter required, see SFkt=10
20	Tool status data of all tools Complete tool data record	Name1 = Empty Name2 = file name with path on the host computer (optional)
21	Tool status data of one tool Variant 1: Selected tool data set	Name1 = tool ID, duplo number Name2 = file name with path on the host computer (optional)
22	Tool status data of one tool Variant 2: Selected tool data set	Name1 = tool ID, duplo number Name2 = file name with path on the host computer (optional)
23	Tool status data of one tool Variant 3: Selected tool data set	Name1 = tool ID, duplo number Name2 = file name with path on the host computer (optional)
24	Data of a tool with adapter number Complete tool data record	Name1 = Adapter number Name2 = File name with path on the host computer
26	Optional/manual loading of a tool Complete tool data set, e. g. for Tool Ident Connection K code 4	Name1 = ID number, duplo number Name2 = File name with path on the host computer
27	W complete tool data set Complete tool data set, e. G. for Tool Ident Connection K code 5	Name1 = ID number, duplo number Name2 = File name with path on the host computer
28	Tool loading from tool pallet	Name1 = Tool pallet number Name2 = name of file with tool status data

SFkt	Function	Remark
29	Unload tool to tool pallet	Name1 = Tool pallet number Name2 = name of file with tool status data

31	Tool status data of one tool
	Variant 1: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

32	Tool status data of one tool
	Variant 2: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

34	Tool status data of one tool
	Variant 3: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

50	Machine assignment data	Name1 = Empty, Name2 = File name with path on the host computer
90	Transfer any file No other processing (optional)	Name1 = Source file name with path Name2 = Target file name with path

7.7.2 Request data as file from SINUMERIK

Function

Called function

```
T_DATA_M(              Host,
                       Machine,
                       OrderNum,
                       SFkt,
                       Name1,
                       Name2)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

Request data as file from SINUMERIK.

Data

Table 7- 8 Parameters for data request from SINUMERIK

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
SFkt	Subfunction number	4 bytes (long int)
Name1	Subfunction numbers: SFkt For name, see also table above Subfunction numbers: SFkt for data dialogs	128 bytes (long int)
Name2	Subfunction numbers: SFkt For name, see also table above Subfunction numbers: SFkt for data dialogs	128 bytes (long int)

Request data as file from SINUMERIK

Example

```
T_DATA_M("FLR1", "BAZ3", 0, 1, "\mpf.dir\ Kw15.mpf", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 10, "\mpf.dir", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 20, "\0", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 21, "Drill10mm,0002", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 22, "Drill10mm,0002", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 23, "Drill10mm,0002", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 50, "\0", "\0");

T_DATA_M("FLR1", "BAZ3", 0, 90, "f:\add_on\mcis_rpc.log", "\0");
```

7.7.3 Request data as file from the host computer

Function

Called function

```
T_DATA_M(              Host,
                       Machine,
                       OrderNum,
                       SFkt,
                       Name1,
                       Name2)
```

Direction of transfer: SINUMERIK → host computer

Meaning

Request data as file from the host computer.

Data

Table 7- 9 Parameters for data request from the host computer

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
SFkt	Subfunction number	4 bytes (long int)
Name1	Subfunction numbers: SFkt For name, see also table above Subfunction numbers: SFkt for data dialogs	128 bytes (long int)
Name2	Subfunction numbers: SFkt For name, see also table above Subfunction numbers: SFkt for data dialogs	128 bytes (long int)

7.8 Accept transferred files

7.8.1 Request to machine: Accept data

Function

Called function

```
R_DATA_M(               Host,
                        Machine,
                        OrderNum,
                        SFkt,
                        Name1,
                        Name2,
                        Date,
                        LastFile)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

The host computer prompts RPC SINUMERIK to retrieve the specified file from the host computer and process it, e.g. enter it in the data management.

Data

Table 7- 10 Parameters for machine acceptance request

<table>
<tr><th>Parameter</th><th colspan="3">Description</th><th>Format</th></tr>
<tr><td>Host</td><td colspan="3">Name of the host computer</td><td>16 bytes (string)</td></tr>
<tr><td>Machine</td><td colspan="3">Machine name</td><td>16 bytes (string)</td></tr>
<tr><td>OrderNum</td><td colspan="3">Order number</td><td>4 bytes (long int)</td></tr>
<tr><td>SFkt</td><td colspan="3">Subfunction number</td><td>4 bytes (long int)</td></tr>
<tr><td>Name1</td><td colspan="3">Subfunction numbers: SFkt
For name see also: Table 7-7 Subfunction numbers: SFkt for data dialogs (Page 101)</td><td>128 bytes (long int)</td></tr>
<tr><td>Name2</td><td colspan="3">Subfunction numbers: SFkt
For name see also: Table 7-7 Subfunction numbers: SFkt for data dialogs (Page 101)</td><td>128 bytes (long int)</td></tr>
<tr><td>Date</td><td colspan="3">Date of the last change
(Unix time, only for NC program file)</td><td>4 bytes (long int)</td></tr>
<tr><td rowspan="5">LastFile</td><td colspan="3">Last file of a workpiece</td><td rowspan="5">4 bytes (long int)</td></tr>
<tr><td>Bit 7</td><td>Bit 0</td><td>Function</td></tr>
<tr><td>0</td><td>0</td><td>Other files will follow</td></tr>
<tr><td>0</td><td>1</td><td>Last file of a workpiece
or individual file</td></tr>
<tr><td>1</td><td>0</td><td>Reset request identifier</td></tr>
</table>

Notes on use

Bit 7 is only relevant if an NC program is requested from the host computer via the SINUMERIK operator interface, but the host computer intentionally sends an NC program with a different name. By default, the RPC SINUMERIK-internal request identifier is only reset when SFkt and Name1 are the same. The host computer can initiate the reset of the request identifier using bit 7.

Example

The file is fetched from the host computer synchronously during R_DATA_M but the processing cannot occur synchronously in the RPC. For this reason, the return value of the RPC can only acknowledge whether or not the file was successfully accepted. Once processing of the file is complete, RPC SINUMERIK sends R_REPORT_H with Type=5 and Error number = SFkt of R_DATA_M as positive acknowledgment.

```
R_DATA_M("FLR1", "BAZ3", 0, 1, "\mpf.dir\ Kw15.mpf", "f:\ncpro\NCKW0815.txt",
862826400, 1);

R_DATA_M("FLR1", "BAZ3", 0, 1, "WKS.DIR\Zylinderkopf.wpd\ Kw15.mpf",
"f:\ncpro\NCKW0815.txt", 862826400, 1);

R_DATA_M("FLR1", "BAZ3", 0, 10, "Main programs", "f:\tmp\NCListe.txt", 0, 1);

R_DATA_M("FLR1", "BAZ3", 0, 26, "Drill10mm,0002", "f:\tmp\wzfile.txt");

R_DATA_M("FLR1", "BAZ3", 0, 27, "TP003", " f:\tmp\tp003.txt ");

R_DATA_M("FLR1", "BAZ3", 0, 28, "TP003", " f:\tmp\tp003.txt ");

R_DATA_M("FLR1", "BAZ3", 0, 1001, "c:\mmc2\oemdata.txt", "c:\tmp\oemdata.txt");

R_DATA_M("FLR1", "BAZ3", 0, 1, "WKS.DIR\Zylinderkopf.wpd\ Kw15.mpf", "NCKW0815.txt",
862826400, 1);
```

7.8.2 Request to host computer: Accept data

Function

Called function

```
R_DATA_H(               Host,
                        Machine,
                        OrderNum,
                        SFkt,
                        Name1,
                        Name2,
                        Date,
                        LastFile)
```

Direction of transfer: SINUMERIK → Host computer

Meaning

The file already transferred with file transfer is to be entered in the data management of the host computer.

Data

Table 7- 11 Parameters for host computer acceptance request

<table>
<tr><th>Parameter</th><th colspan="3">Description</th><th>Format</th></tr>
<tr><td>Host</td><td colspan="3">Name of the host computer</td><td>16 bytes (string)</td></tr>
<tr><td>Machine</td><td colspan="3">Machine name</td><td>16 bytes (string)</td></tr>
<tr><td>OrderNum</td><td colspan="3">Order number</td><td>4 bytes (long int)</td></tr>
<tr><td>SFkt</td><td colspan="3">Subfunction number</td><td>4 bytes (long int)</td></tr>
<tr><td>Name1</td><td colspan="3">Subfunction numbers: SFkt
For name see also: Table 7-7 Subfunction numbers: SFkt for data dialogs (Page 101)</td><td>128 bytes (long int)</td></tr>
<tr><td>Name2</td><td colspan="3">Subfunction numbers: SFkt
For name see also: Table 7-7 Subfunction numbers: SFkt for data dialogs (Page 101)</td><td>128 bytes (long int)</td></tr>
<tr><td>Date</td><td colspan="3">Date of the last change
(Unix time, only for NC program file)</td><td>4 bytes (long int)</td></tr>
<tr><td rowspan="4">LastFile</td><td colspan="3">Last file of a workpiece</td><td rowspan="4">4 bytes (long int)</td></tr>
<tr><td>Bit 7</td><td>Bit 0</td><td>Function</td></tr>
<tr><td>0</td><td>0</td><td>Other files will follow</td></tr>
<tr><td>0</td><td>1</td><td>Last file of a workpiece
or individual file</td></tr>
</table>

7.9 Delete data on the SINUMERIK

Function

Called function

```
C_DELETE_M (           Host,
                       Machine,
                       OrderNum,
                       SFkt,
                       Name1,
                       Name2)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

Delete data from the previous file transfer.

Data

Table 7- 12 Parameters for delete data

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
SFkt	Subfunction number	4 bytes (long int)
Name1	Name of the file to be deleted	128 bytes (long int)
Name2	Additional name	128 bytes (long int)

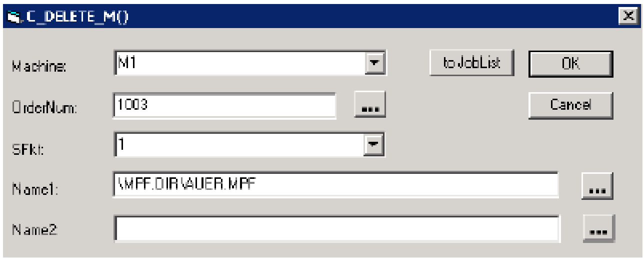
Note

Files in the data management

Only files in the data management can be addressed, `SFkt = 1`, e.g.: `Name1` ="\mpf.dir\AUER.mpf"

Example

```
C_DELETE_M ( "H1", "M1", 1003, 1, "\mpf.dir\ AUER.mpf", "\0");
```



Machine: M1
OrderNum: 1003
SFkt: 1
Name1: \MPF.DIR\AUER.MPF
Name2:

```
R_REPORT_H
OrderNum->1003 Machine->"BAZ1X1" Host->"H1" Type->5
Number( 0 ... 9 ) -> 1 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 0
ResInt1->0 ResInt2->0 ResByte->"" Runtime->0,1721
```

Result:

Type	= 5	Identification positive acknowledgment
Number(0)	= 1	Initiated SFkt

RPC SINUMERIK sends an RPC call `R_REPORT_H Type=4` to the host computer as negative acknowledgment.

Error number:

-250 - NC program cannot be deleted.

7.10 NC programs

The following description of program handling is a special application of the functions described in the following Chapter: Data dialogs (Page 100)

Knowledge of these functions is required.

7.10.1 Request NC program, initiated by host computer

Sequence

1. Function call

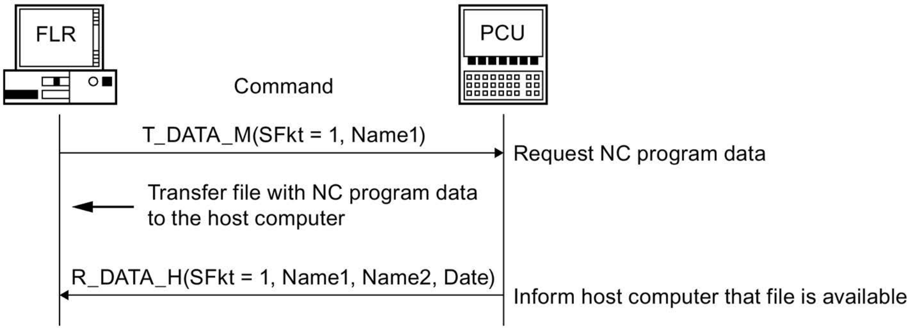
   `T_DATA_M()`
   `SFkt = 1`
   `Name1` = program name in the data management, e.g.: \mpf.dir\Auer2.mpf
   transfer direction, host computer → SINUMERIK
   Significance:
   The host computer requests a certain NC program from SINUMERIK.

2. File transfer

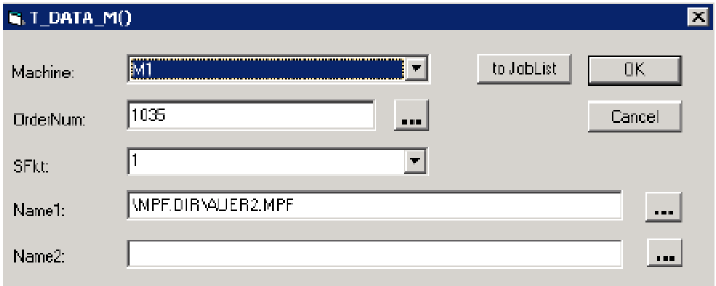
   The file with the requested NC program is transferred.

3. Called function

   `R_DATA_H()`
   `SFkt = 1`
   `Name1` = Program name in the data management
   `Name2` = Name incl. path of file on host computer
   Date = Date of last change
   Direction of transfer SINUMERIK → host computer
   Meaning:
   The SINUMERIK controller requests a certain NC program from the host computer.


FLR
PCU
Command
T_DATA_M(SFkt = 1, Name1)
Request NC program data
Transfer file with NC program data to the host computer
R_DATA_H(SFkt = 1, Name1, Name2, Date)
Inform host computer that file is available

Example



T_DATA_M()

Machine: M1

OrderNum: 1035

SFkt: 1

Name1: \MPF.DIR\AUER2.MPF

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type=5` to the host computer as positive acknowledgment.

```
R_DATA_H
OrderNum->1035 Machine->"M1" Host->"H1" SFkt->1Name1->”\mpf.dir\Auer2.mpf”
Name2->"\\192.168.3.230\RPCHostDirectory\Putdir\AUER2.mpf"datVal-
>1465379838 LastFile->0 runtime->0,0142
```

Result

The NC program is stored in the configured host directory.

7.10.2 Request NC program, initiated by SINUMERIK

Sequence

1. Function call

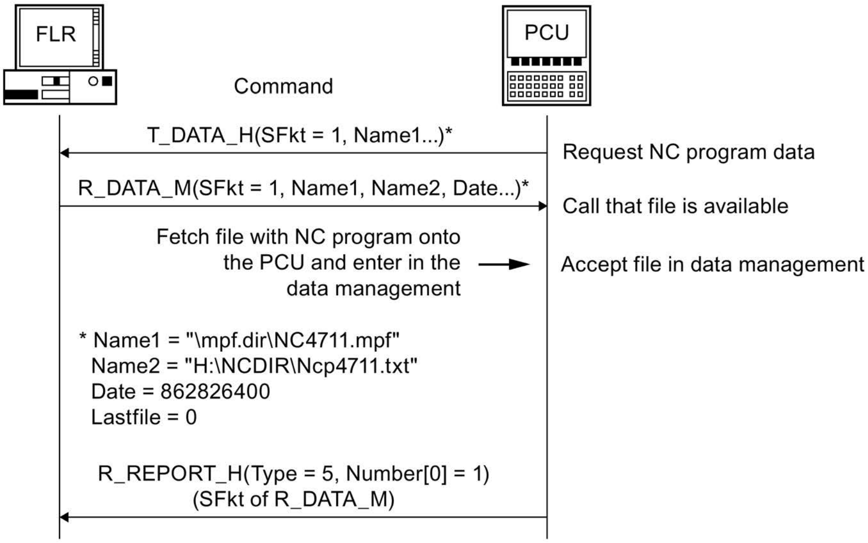
   `T_DATA_H()`
   `SFkt = 1`
   `Name1` = program name in the data management
   Transfer direction: Host computer → SINUMERIK
   Meaning:
   The SINUMERIK controller requests a certain NC program from the host computer.

2. File transfer:

   The file with the requested NC program is transferred.

3. Called function

   `R_DATA_M()`
   `SFkt = 1`
   `Name1` = Program name in the data management
   `Name2` = Name incl. path of file on host computer
   Date = Date of last change
   Direction of transfer: Host computer → SINUMERIK
   Meaning:
   The host computer provides the requested NC program for the SINUMERIK controller.


FLR
PCU
Command
T_DATA_H(SFkt = 1, Name1...)*
Request NC program data
R_DATA_M(SFkt = 1, Name1, Name2, Date...)*
Call that file is available
Fetch file with NC program onto the PCU and enter in the data management
Accept file in data management
* Name1 = "\mpf.dir\NC4711.mpf"
Name2 = "H:\NCDIR\Ncp4711.txt"
Date = 862826400
Lastfile = 0
R_REPORT_H(Type = 5, Number[0] = 1)
(SFkt of R_DATA_M)

Note

Request NC programs individually

Only individual NC programs can be requested and transferred.

The workpiece holder designation must end with "\0" and have a maximum length including "\0" of 6 bytes. For the NC program, the data management path must always be included, e.g.: NCProg = "\mpf.dir\Zylinderkopf.mpf".

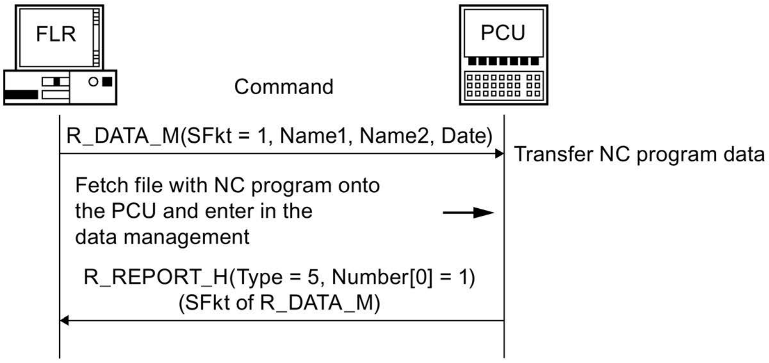
Because the data management in SINUMERIK does not recognize versions, the last date of change and the file size can be specified. If the NC program is contained in the data management of SINUMERIK but with a different size or date of change, RPC SINUMERIK must request the current file from the host computer before the program may be activated for machining.

7.10.3 Transferring the NC program

Sequence

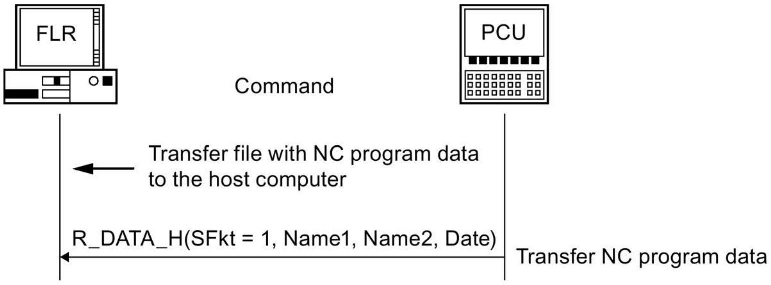
1. Function call

   `R_DATA_M()`
   `SFkt = 1`
   `Name1` = Program name in the data management
   `Name2` = Name incl. path of file on host computer
   Date = Date of last change
   Direction of transfer Host computer → SINUMERIK
   Meaning:
   The host computer provides a certain NC program for the SINUMERIK controller.


FLR
PCU
Command
R_DATA_M(SFkt = 1, Name1, Name2, Date)
Transfer NC program data
Fetch file with NC program onto the PCU and enter in the data management
R_REPORT_H(Type = 5, Number[0] = 1) (SFkt of R_DATA_M)

2. Called function

   `R_DATA_H()`
   `SFkt = 1`
   `Name1` = Program name in the data management
   `Name2` = Name incl. path of file on host computer
   Date = Date of last change
   Direction of transfer SINUMERIK → Host computer
   Meaning:
   The SINUMERIK controller provides a certain NC program to the host computer.


FLR
PCU
Command
Transfer file with NC program data to the host computer
R_DATA_H(SFkt = 1, Name1, Name2, Date)
Transfer NC program data

7.10.4 Delete NC program on the machine

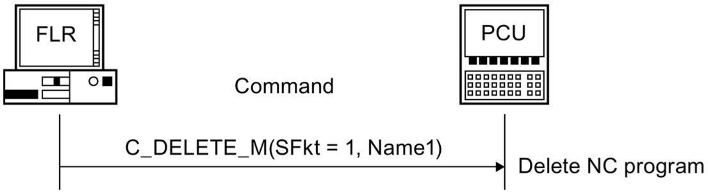
Sequence

1. Function call

   `C_DELETE_M()`
   `SFkt = 1`
   `Name1` = program name in the data management
   Transfer direction: Host computer → SINUMERIK
   Meaning:
   The host computer instructs SINUMERIK to delete the program specified by Name1.

2. Data

   The data to be transferred is described above under `C_DELETE_M()`.
   The following must be assigned:
   `SFkt = 1,`
   `Name1` = "\mpf.dir\zylinderkopf.mpf"
   - OR -
   "\spf.dir\4711.spf"



7.10.5 Request list of existing NC programs, initiated by host computer

Sequence

1. Function call

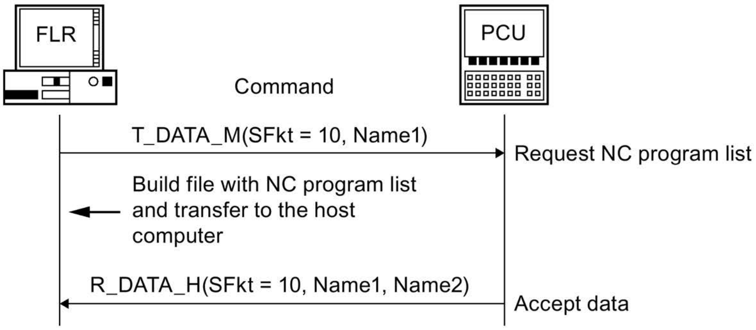
   `T_DATA_M()`
   `SFkt = 10`
   `Name1` = path in the data management, e.g.: \mpf.dir
   Transfer direction, host computer → SINUMERIK
   Meaning:
   The host computer requests the list of existing NC program from SINUMERIK.

2. File transfer

   The file with the requested NC program list is transferred to the host computer.

3. Called function

   `R_DATA_H()`
   `SFkt = 10`
   `Name1` = Path of the data management
   `Name2` = File name with NC program list
   Direction of transfer SINUMERIK → Host computer
   Meaning:
   Transfer of machine to the host computer: List of existing NC programs.

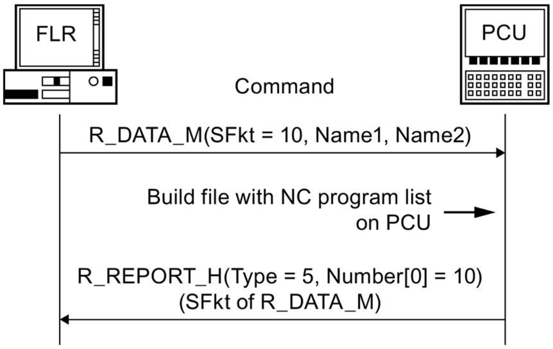

FLR
PCU
Command
T_DATA_M(SFkt = 10, Name1)
Request NC program list
Build file with NC program list and transfer to the host computer
R_DATA_H(SFkt = 10, Name1, Name2)
Accept data

7.10.6 Transfer NC program list

Sequence

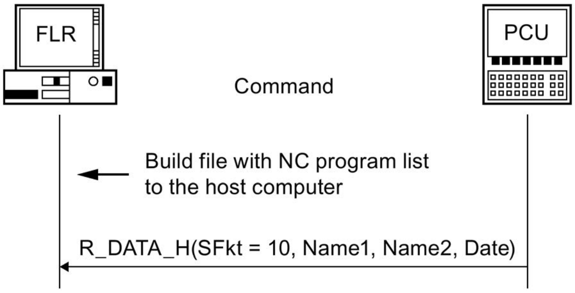
1. Function call

   `R_DATA_M()`
   `SFkt = 10`
   `Name1` = Path of the data management
   `Name2` = file name with NC program list
   Direction of transfer: Host computer → SINUMERIK
   Meaning:
   Transfer of the host computer to SINUMERIK: List of existing programs.


FLR
PCU
Command
R_DATA_M(SFkt = 10, Name1, Name2)
Build file with NC program list on PCU
R_REPORT_H(Type = 5, Number[0] = 10) (SFkt of R_DATA_M)

2. Called function

   `R_DATA_H()`
   `SFkt = 10`
   `Name1` = Path of the data management
   `Name2` = File name with NC program list
   Direction of transfer SINUMERIK → host computer
   Meaning:
   Transfer from SINUMERIK to the host computer: List of existing programs.


FLR
PCU
Command
Build file with NC program list to the host computer
R_DATA_H(SFkt = 10, Name1, Name2, Date)

The file with the NC program list contains a list of files as well as the subdirectories contained in this data management path. The content of these subdirectories is not listed. If needed, it must be determined with a separate request.

Structure of the file returned with R_DATA_H

```
\mpf.dir

Zylinderkopf.MPF,FM,5320,876403708

Kurbelwelle.MPF,FN,8300,862826400
```

Rows

The first row of the file contains the directory that was specified in Name1 and whose content is listed.

The other rows contain a file or subdirectory name with additional information separated by a comma.

Columns

The first column contains the NC program name or the name of the subdirectory.

The second column contains two characters:

- The first character identifies whether a file (F) or subdirectory (D) is involved.
- The second character indicates whether the file is located in the PCU or NC

Example

"Fx" - File

"Dx" - Directory

"xN" - located in the NC or in the NC and PCU

"xM" - located on the PCU.

The third column contains the file size in bytes.

The fourth column contains a decimal number representing the date of the file as UNIX time in seconds since 1970-01-01.

The time for cylinder head 876403715 means 1997-10-09 15:28:35 and the time for crank shaft 862826400 means 1997-05-05 12:00:00.

7.11 Tool dialogs

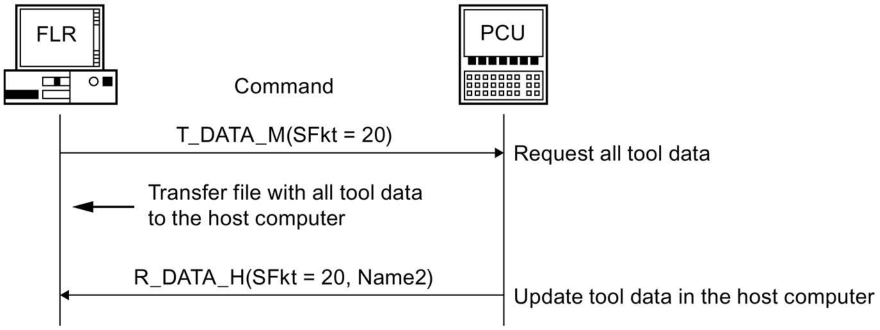
7.11.1 Interrogate the complete tool magazine assignment

Tool data is always transferred as a file.

Function

Table 7- 13 Scan tool magazine assignment

Direction of transfer	Command		Meaning
Host computer → SINUMERIK Host computer ← SINUMERIK	T_DATA_M	(SFkt = 20)	Request all tool data Transfer file with all tool status data to the production host controller Notify the host computer of arrival of the file
Host computer ← SINUMERIK	R_DATA_H	(SFkt = 20, Name1 = Empty, Name2 = File with all tool status data)	


FLR
PCU
Command
T_DATA_M(SFkt = 20)
Request all tool data
Transfer file with all tool data to the host computer
R_DATA_H(SFkt = 20, Name2)
Update tool data in the host computer

7.11.2 Optional/manual loading

Tool data is always transferred as a file.

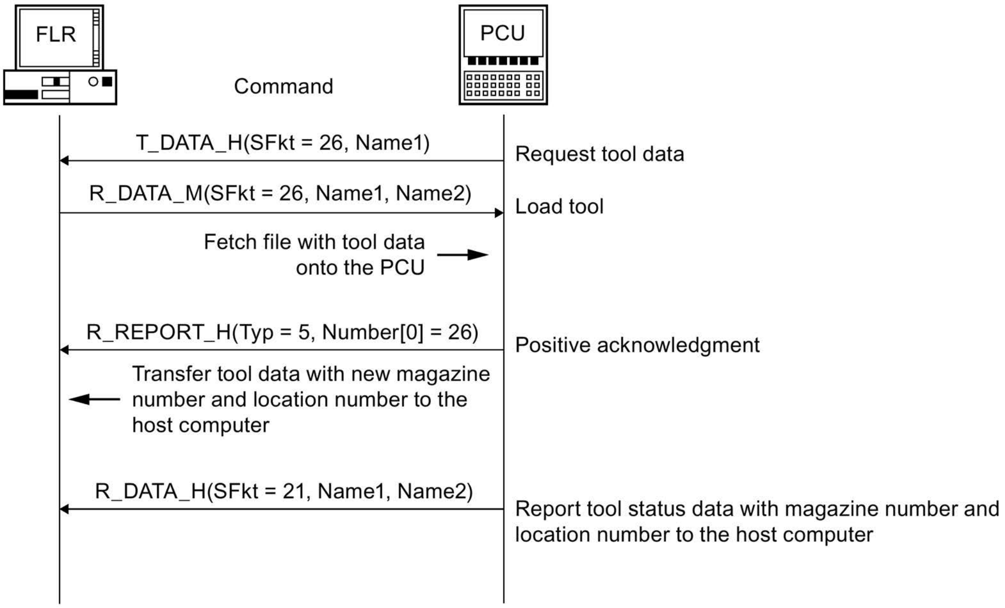
Function

Use, e.g. for Tool Ident Connection K code 4

K code 4 = load tool from host computer

Table 7- 14 Optional/manual loading

Direction of transfer	Command		Meaning
Host computer ← SINUMERIK	T_DATA_H	(SFkt = 26, Name1=ID number, duplo number)	Request tool data
Host computer → SINUMERIK	R_DATA_M	(SFkt = 26, Name1=ID number, duplo number, Name2=File name with tool status data)	Prompt to SINUMERIK to fetch file. Fetch file with tool status data to SINUMERIC.
Host computer ← SINUMERIK	R_DATA_H	(SFkt = 21, Name1=ID number, duplo number, Name2=File name with tool status data)	Load tool File with tool status data with new magazine number and location number to the host computer (this is the loading acknowledgment).


FLR
PCU
Command
T_DATA_H(SFkt = 26, Name1)
Request tool data
R_DATA_M(SFkt = 26, Name1, Name2)
Load tool
Fetch file with tool data onto the PCU
R_REPORT_H(Typ = 5, Number[0] = 26)
Positive acknowledgment
Transfer tool data with new magazine number and location number to the host computer
R_DATA_H(SFkt = 21, Name1, Name2)
Report tool status data with magazine number and location number to the host computer

7.11.3 Optional/manual loading

Tool data is always transferred as a file.

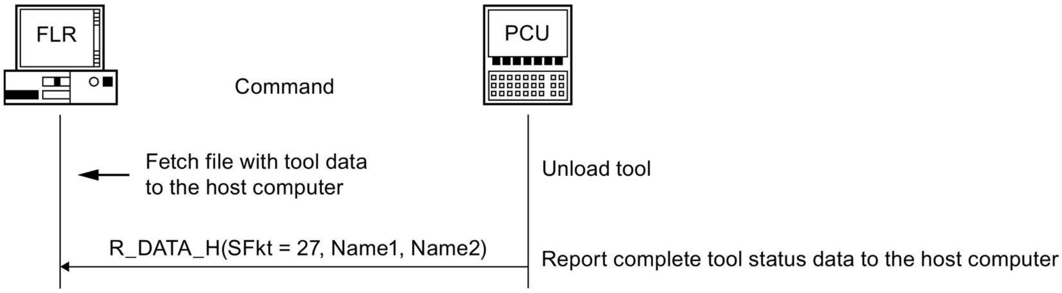
Function

Use, e.g. for Tool Ident Connection K code 5

K code 5 = unload tool from host computer

Table 7- 15 Optional/manual unloading to the host computer

Direction of transfer	Command		Meaning
Host computer ← SINUMERIK	R_DATA_H	(SFkt = 27, Name1=ID number, duplo number, Name2=File name with tool status data)	File with tool status data with new magazine number and location number to host computer (unloading acknowledgment).


FLR
PCU
Command
Fetch file with tool data to the host computer
Unload tool
R_DATA_H(SFkt = 27, Name1, Name2)
Report complete tool status data to the host computer

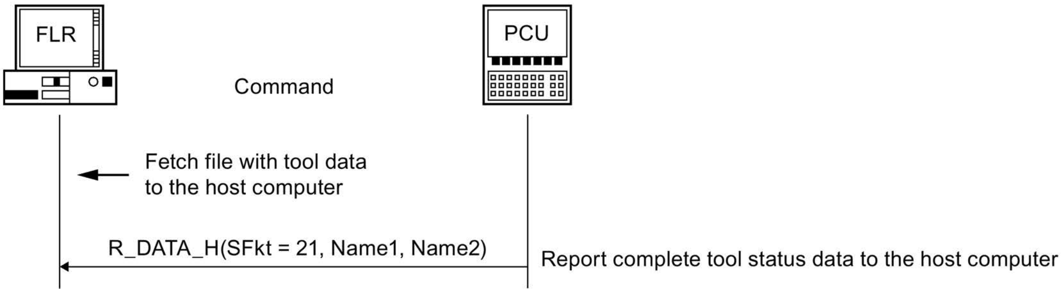
7.11.4 Report tool

Tool data is always transferred as a file.

Function

Table 7- 16 Report tool

Direction of transfer	Command		Meaning
Host computer ← SINUMERIK	R_DATA_H	(SFkt = 21, Name1=ID number, duplo number, Name2=File name with tool status data)	File with tool status data to host computer (report tool)


FLR
PCU
Command
Fetch file with tool data to the host computer
R_DATA_H(SFkt = 21, Name1, Name2)
Report complete tool status data to the host computer

7.12 Machine assignment data

7.12.1 Overview

For machines with more than 3 docking positions, such as a transport system with buffers, the data of all docking positions is transferred in an ASCII file.

Transfer of the data to the host computer is followed by RPCR_ DATA_H SFkt = 50, see also section: Overview (Page 100).

The following is transferred for each docking position:

- Docking position number
- Workpiece holder number
- Workpiece holder status

7.12.2 Structure of the ASCII file

Parameter

The data fields are located in the file, separated by commas, as follows:

DockPos 1, DockPosStatus, WPC, WPCStatus <line feed>

DockPos 2, DockPosStatus, WPC, WPCStatus <line feed>

...

DockPos n, DockPosStatus, WPC, WPCStatus <EOF>

Table 7- 17 Description of file parameters

Parameter	Description	Format
Dockpos	Docking position number	ASCII
DockPosStatus	Docking position status 0: Enabled 1: Disabled for TPS 2: Fault ASCII	ASCII
WPC	Workpiece holder designation	ASCII
WPCStatus	Workpiece holder status 1: Not machined, no program assignment 2: Not machined, program assigned 16: In progress 32: Finished 64: Finished with errors 128: Only for buffering	ASCII

The host computer can request this data with `T_DATA_M(SFkt = 50)`.

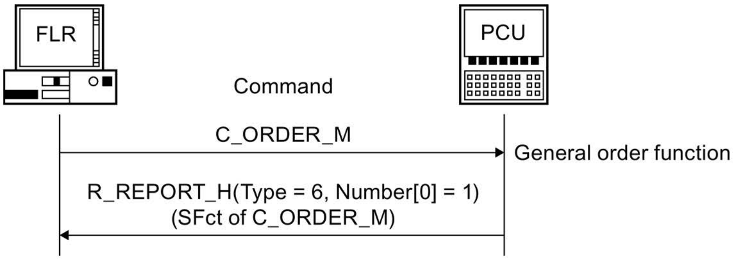
7.13 General order functions

Function

Called function

```
C_ORDER_M(              Host,
                        Machine,
                        OrderNum,
                        SFkt,
                        Name1,
                        Name2,
                        Name3,
                        Name4,
                        Parameter1,
                        Parameter2,
                        Parameter3,
                        Parameter4)
```

Direction of transfer: Host computer → SINUMERIK



Meaning

Once processing is complete, RPC SINUMERIK either sends `R_REPORT_H` with type = 6 and error number = SFkt of `C_ORDER_M` as positive acknowledgment or `R_REPORT_H` with type = 4 and with a corresponding error number as negative acknowledgment.

Table 7- 18 Subfunction numbers: SFkt for the general order function

SFkt	Function	Remark

2	Loading NC programs	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1= Function (0,2,3)

3	Loading and selection of NC programs	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1 = Function (0-3)
		Parameter2 = Channel number

4	Unloading of NC programs	Name1 = NC program
		Name2 = NC name

5	Selection of NC programs	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1 = Function (0.1)
		Parameter2 = Channel number

SFkt	Function	Remark

6	Deselection of NC programs	Name2 = NC name (optional)
		Parameter2 = Channel number

11	Determining the number of NC channels	No parameters necessary
12	Determining the NC program selected in the NC channel	Parameter2 = NC channel number
13	Determining the assigned TO area of all NC channels of the SINUMERIK 840D sl	No parameters necessary

100	Set time/date on SINUMERIK/PLC	Name2 = NC name (optional)
		Parameter1 = UNIX time
		Parameter2 = Date / time SINUMERIK / PLC

200	Set protection level	Name2 = NC name (optional)
		Parameter1 = Protection level

201	Reset protection level	Name2 = NC name (optional)

Note

Use NC name

Der NC name can only be used after the necessary additions are made to the "NETNAMES.INI" and "MMC.INI" files.

Data

Table 7- 19 Parameters for Mode switchover

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
SFkt	Subfunction number	4 bytes (long int)
Name1	*	128 bytes (long int)
Name2	*	128 bytes (long int)
Name3	*	128 bytes (long int)
Name4	*	128 bytes (long int)
Parameter1	*	4 bytes (long int)
Parameter2	*	4 bytes (long int)
Parameter3	*	4 bytes (long int)
Parameter4	*	4 bytes (long int)

* Additional name

See also the above table: Subfunction number: SFkt for the general order function

7.13.1 Loading of NC programs

Function

Called function

```
C_ORDER_M() with
SFkt = 2
```

Name1 =	NC program name
Name2 =	NC name (optional)
Parameter1 =	0: Load individual NC program into the NC 2: Load complete workpiece into the NC 3: Load workpiece into the NC with the exception of the NC program specified in Name1
Direction of transfer:	Host computer → SINUMERIK

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 2, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,0 ,0 ,0);

C_ORDER_M ("FLR1", "BAZ3", 0, 2, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,0 ,0
,0);
```

Meaning

Loading of individual NC programs or workpieces from the SINUMERIK data management into the NC. The NC name (Name2) only has to be specified if the NC program is not to be loaded into the standard NC.

Error number

-262: NC program cannot be loaded

7.13.2 Loading and selection of NC programs

Function

Called function

```
C ORDER M() with SFkt = 3
```

Name1 =	Workpiece / NC program name
Name2 =	NC name (optional)
Parameter1 =	0: Load NC program into NC and select 1: Process individual program from external source 2: Load complete workpiece into NC and select NC program from Name1 3: Load complete tool into NC and select the NC program specified in Name1 for processing from external source
Parameter2 =	Channel number
Direction of transfer:	Host computer → SINUMERIK

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 3, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,2 ,0 ,0);
```

```
C_ORDER_M ("FLR1", "BAZ3", 0, 3, "\WKS.DIR\Zylinderkopf.wpd\Kw15.mpf", "NCU_2", "\0",
"\0",0 ,0 ,0 ,0);
```

Meaning

The function corresponds to SFkt = 2. In addition, the specified program is selected. The NC name (Name2) only has to be specified if the NC program is not to be loaded into the standard NC.

Note

Service `R_VAR_M`

If the NC program is to then be automatically started or if the operator is to receive a display of the NC program, this must be implemented using the RPC SINUMERIK service `R_VAR_M`.

Error number

-262: NC program cannot be loaded.

-263: NC program cannot be selected.

7.13.3 Unloading of NC programs

Function

Called function

```
C_ORDER_M() with
SFkt = 4
```

Name1=	NC program name or workpiece name
Name2=	NC name (optional)
Direction of transfer:	Host computer → SINUMERIK

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",0 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,0 ,0 ,0);
C_ORDER_M ("FLR1", "BAZ3", 0, 4, "\WKS.DIR\Zylinderkopf.wpd\Kw15.mpf", "\0", "\0", "\0",0
,0 ,0 ,0);
```

Meaning

Unloading of NC programs or workpieces from the NC. The NC programs are then located in the PCU data management. The NC name (Name2) only has to be specified if loading into the standard NC is not to occur.

Error number

-264: NC program cannot be unloaded.

7.13.4 Selection of NC programs

Function

Called function

```
C_ORDER_M() with
SFkt = 5
Name1=                 NC program name
Name2=                 NC name (optional)
Parameter1=            0:  Process on NC
                       1:  Process individual program from external source
Parameter2=            Channel number
```

Direction of transfer: Host computer → SINUMERIK

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 5, "\mpf.dir\kw15.mpf", "\0", "\0", "\0",1 ,5 ,0 ,0);
```

```
C_ORDER_M ("FLR1", "BAZ3", 0, 5, "\mpf.dir\kw15.mpf", "NCU_2", "\0", "\0",0 ,1 ,0
,0);
```

Meaning

Selection of NC programs in a certain channel. The function corresponds to SFkt 3 without previous loading of the NC program. The NC name (Name2) only has to be specified if the NC program is not to be loaded into the standard NC.

Error number

-265: NC program cannot be selected.

7.13.5 Deselection of NC programs

Function

Called function

```
C_ORDER_M() with
SFkt = 6
Name2=                 NC name (optional)
Parameter2=            Channel number
```

Direction of transfer: Host computer → SINUMERIK

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 6, "\0", "\0", "\0", "\0",0 ,5 ,0 ,0);
```

```
C_ORDER_M ("FLR1", "BAZ3", 0, 6, "\0", "NCU_2", "\0", "\0",0 ,1 ,0 ,0);
```

Meaning

Deselection of the current NC program. The NC name (Name2) only has to be specified if the NC is not the standard NC.

Error number

-266: NC program cannot be deselected.

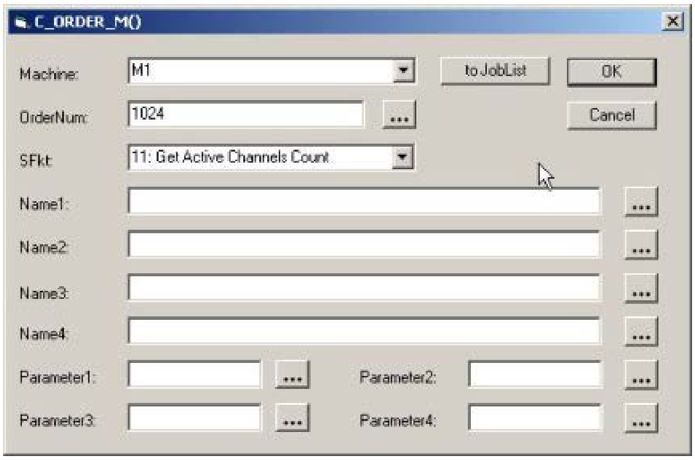
7.13.6 Determine the number of NC channels

The general order function `C_ORDER_M SFkt = 11` is used to determine the number of configured NC channels of SINUMERIK 840D sl.

Function

Table 7- 20 Subfunction number SFkt for the NC channels

SFkt	Function	Remark
11	Determine number of NC channels	No parameters necessary



C_ORDER_M()

Machine: M1

OrderNum: 1024

SFkt: 11: Get Active Channels Count

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->0 Machine->"M1" Host->"H1" Type>6

Number(0..9) -> 11 2 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> 0 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->""
```

Result

```
Type := 6 - Identifier positive acknowledgment

Number(0) := 11 - Initiated SFkt

Number(1) := 2 - Number of NC channels determined
```

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-273 - Number of NC channels could not be determined.

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, then it is mandatory that you use the latest program version!

7.13.7 Determine selected NC program in NC channel

The general order function `C_ORDER_M SFkt = 12` is used to determine the selected NC program of the required NC channel of the SINUMERIK 840D sl.

Function

Table 7- 21 Subfunction number: SFkt for NC channel

SFkt	Function	Remark
12	Determining the selected NC program in the NC channel	Parameter2 = NC channel number

Once processing is complete, RPC SINUMERIK sends an RPC call `R_MACHINE_H` to the host computer as positive acknowledgment.

```
R_MACHINE_H

OrderNum->1025 Machine->"M1" Host->"H1" MachineMode->1 MachineStatus->0

NCProgramm->"\WKS.DIR\TEST.WPD\TEST5.MPF" ClampCubeSide->0

DockPos1->1 DockPos2->0 DockPos3->0 DockPosStatus1->0 DockPosStatus2->0
DockPosStatus3->0

WPC1->"AAAA" WPC2->"" WPC3->"" WPCStatus1->32 WPCStatus2->0 WPCStatus3->0 ResInt1->0

ResInt2->0 ResByte->"2"
```

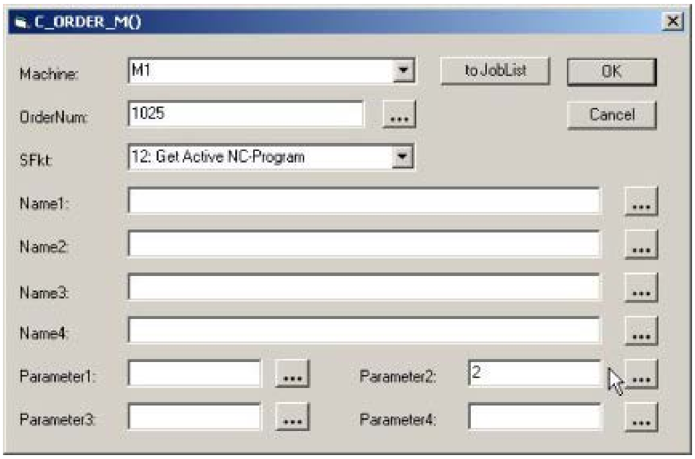
Result

```
NC program := "<NC program>" - NC program selected in the NC channel

ResByte := "<NC channel>" - Number of NC channel
```

RPC SINUMERIK sends an RPC call `R_REPORT_H type = 4` to the host computer as negative acknowledgment.

Error number

-273 - Number of NC channels could not be determined.



C_ORDER_M()

Machine: M1
OrderNum: 1025
SFkt: 12: Get Active NC-Program
Name1:
Name2:
Name3:
Name4:
Parameter1:
Parameter2: 2
Parameter3:
Parameter4:

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, then it is mandatory that you use the latest program version!

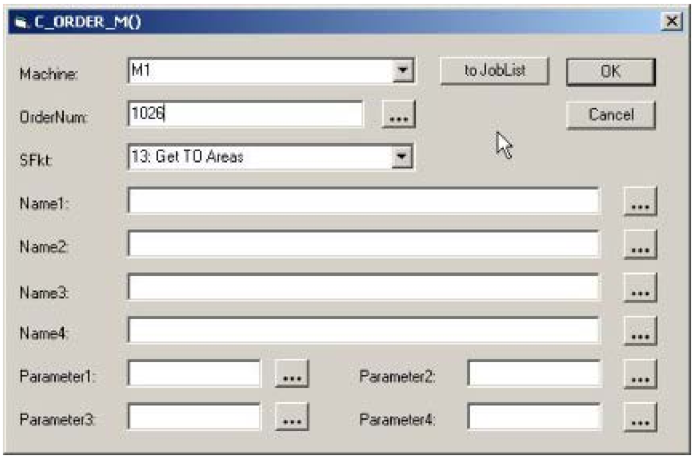
7.13.8 Determine assigned TO area of all NC channels

The general order function `C_ORDER_M SFkt = 13` is used to determine the assigned TO area of all NC channels of the SINUMERIK.

Function

Table 7- 22 Subfunction number: SFkt for TO area

SFkt	Function	Remark
13	Determine assigned TO area of the NC channels	No parameters necessary



Machine: M1

OrderNum: 1026

SFkt: 13: Get TO Areas

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1025 Machine->"M1" Host->"H1" Type->6

Number(0..9) -> 1 1 2 0 0 0 0 0 0 0 Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> 0 0 0 0 0 0 0 0 0 0 ResInt1->0 ResInt2->0 ResByte->""
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0):=` 1 `- Channel 1 has assigned the TO area` 1

`Number(1):=` 1 `- Channel 2 has assigned the TO area` 1

`Number(2):=` 2 `- Channel 3 has assigned the TO area` 2

`Number(3):=` 0 `- Channel 4 not configured`

...

`Number(9):=` 0 `- Channel 10 not configured`

RPC SINUMERIK sends an RPC call `R_REPORT_H type = 4` to the host computer as negative acknowledgment.

Error number

-273 - Number of NC channels could not be determined.

-275 - TO area could not be determined.

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, then it is mandatory that you use the latest program version!

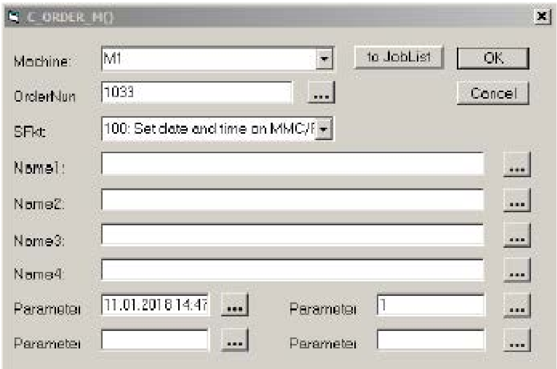
7.13.9 Set time/date on SINUMERIK/PLC

Function

Called function

```
C_ORDER_M() with
SFkt = 100
Name2=                  NC name (optional)
Parameter1=             UNIX time
Parameter2=             0:    Date/time on SINUMERIK
                        1:    Date/time on SINUMERIK and PLC

                        2:    Date/time on PLC
```

Direction of transfer: Host computer → SINUMERIK



Machine: M1
OrderNum 1033
SFkt: 100: Set date and time on MMC/F
Name1:
Name2:
Name3:
Name4:
Parameter 11.01.2018 14:47   Parameter 1
Parameter   Parameter

Meaning

Set date and time on SINUMERIK / PLC

Enter a time in GMT.

The NC name (Name2) only has to be specified if the date or time is not to be changed on the standard NC.

Parameter2 shows the target device the date and time change.

Note

To avoid diverging time stamps in the incoming and outgoing alarms, the time must be synchronized regularly.

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 100, "\0", "\0", "\0", "\0",862826400 ,0 ,0 ,0);

C_ORDER_M ("FLR1", "BAZ3", 0, 100, "\0", "NCU_2", "\0", "\0",862826400 ,1 ,0 ,0);
```

Error number

-270: Time/date cannot be updated.

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, then it is mandatory that you use the latest program version!

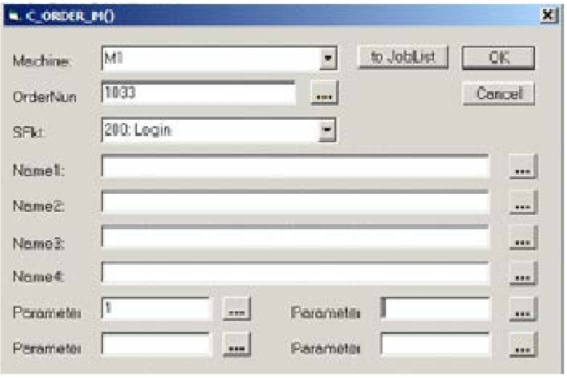
7.13.10 Set protection level

Function

Called function

```
C_ORDER_M() with
SFkt = 200
Name2=                  NC name (optional)
Parameter1=             Protection level:
                        0=  System password
                        1=  Manufacturer password
                        2=  Service password
                        3=  User password
```

Direction of transfer: Host computer → SINUMERIK



C_ORDER_M()

Machine: M1

OrderNum: 1033

SFkt: 200: Login

Name1:

Name2:

Name3:

Name4:

Parameter: 1

Parameter:

Parameter:

Parameter:

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 200, "\0", "\0", "\0", "\0",2 ,0 ,0 ,0);
```

Meaning

Set protection level, e.g.: Protection level 2:

The NC name (Name2) only has to be specified if the NC is not the standard NC.

Certain tasks can be performed only when a specific protection level is set. This call allows the host computer to set a protection level temporarily, e.g. before the data management system is accessed.

Note

Reset protection level again

This protection level is valid for the whole SINUMERIK system!

The original protection level must be set again as soon as possible to protect the system against misuse!

Error number

-271: "Set protection level" has not worked.

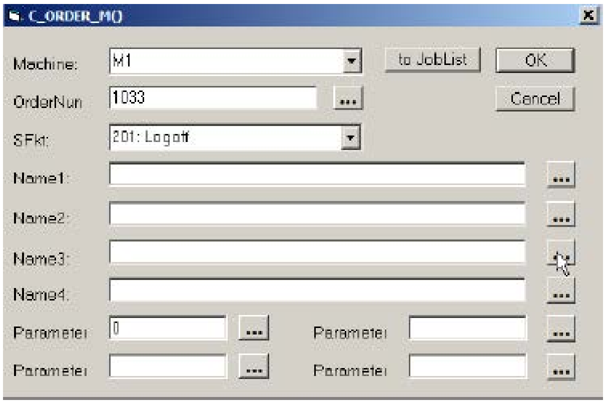
7.13.11 Reset protection level

Function

Called function

```
C_ORDER_M() with
SFkt = 201
Name2=                  NC name (optional)
```

Direction of transfer: Host computer → SINUMERIK



Machine: M1

OrderNum: 1033

SFkt: 201: Logoff

Name1:

Name2:

Name3:

Name4:

Parameter: 0

Parameter:

Parameter:

Parameter:

Example

```
C_ORDER_M ("FLR1", "BAZ3", 0, 201, "\0", "\0", "\0", "\0",0 ,0 ,0 ,0);
```

Meaning

Reset protection level.

Note

Always reset to the selected protection level

A reset is always made to the selected protection level, not to the protection level last set by RPC/CMI.

Error number

-272: "Reset protection level" has not worked.

7.13.12 Determine tool status data in the TO area

The tool status data in the TO area is requested as a file from SINUMERIK using RPC call `T_DATA_M SFkt = 31..33`.

Function

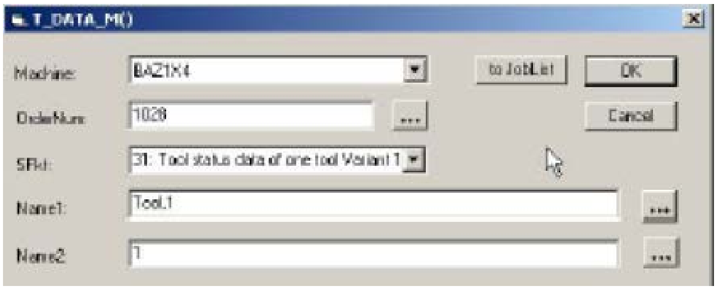
Table 7- 23 Subfunction number: SFkt for tool status data

SFkt	Function	Remark

31	Tool status data of a tool in the TO area as variant 1	Name1 = Tool ID, duplo number
		Name2 = TO area

32	Tool status data of a tool in the TO area as variant 2	Name1 = Tool ID, duplo number
		Name2 = TO area

33	Tool status data of a tool in the TO area as variant 3	Name1 = Tool ID, duplo number
		Name2 = TO area



Machine: BAZ1X4

OrderNum: 1028

SFkt: 31: Tool status data of one tool Variant 1

Name1: Tool.1

Name2: 1

After processing is complete, RPC SINUMERIK transfers the tool data as a file to the host computer. RPC SINUMERIK then sends RPC call `R_DATA_H` SFkt = 31 ... 33 to the host computer, see also Chapter: Overview (Page 100).

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, then it is mandatory that you use the latest program version!

7.13.13 Report tool status data from TO area

With the request from the PLC "Report tool", the PLC initiates that the specified tool is reported to the host computer.

With this function, the PLC indicates the tool status data of a tool from the TO area it has specified as variant 1, 2, or 3. It places the number of the corresponding TO range in data byte DB12.DBB13.

The function is answered with RPC call `R_DATA_H` SFkt = 31, 32, or 33
Name1 = <tool_identifier, duplo_number> Name2 = <path and name of the tool file>.

If the function was not able to derive the expected tool status data, this function is returned with RPC call `R_REPORT_H` type = 4 number[0] = <detailed_error>

Note

- The meaning of the global data in data block 12, data byte 13 has been changed.
- The previous meaning 'Number of program assignments' has been eliminated.
- The new meaning 'Number TO range' has been introduced.

DB12 global data

The data elements listed below are specified by the PLC and define the tool to be reported.

Table 7- 24 DB12 global data for tool data

Data element	Short name	Data type	Access from	Offset
...				
Identifier for tool data	DataTyp	Byte	PLC	5
Magazine number	MagNum	Word	PLC	6 - 7
Location number	PlaceNum	Word	PLC	8 - 9
T number	TNum	Word	PLC	10 - 11
Tool area	TOArea	Byte	PLC	13
...				

Note

Data element tool area

If the value 21, 22 or 23 is specified in data element "Identifier for tool data" and the value 0 is transferred by the PLC in data element "Tool area", the tool to be reported is determined in the tool area of the first NC channel.

If the value <> 0 is transferred by the PLC in data element "Tool area", the tool to be reported is determined in the specified "Tool area" data element.

This function "Report tool" with DB12 must be programmed by the OEM in the PLC.

7.14 Mode switchover

7.14.1 Overview

Function

Called function

```
C_MODE_M(                   Host,
                            Machine,
                            OrderNum,
                            Mode)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

Command to the machine to switch to a certain operating mode.

Data

Table 7- 25 Parameters for Mode switchover

Parameter	Description		Format
Host	Name of the host computer		16 bytes (string)
Machine	Machine name		16 bytes (string)
OrderNum	Order number		4 bytes (long int)
Mode	Operating mode		4 bytes (long int)
	1:	Special mode ON	
	2:	Special mode OFF	
	3:	Switch off components	
	4:	Switch on components	
	5:	In request from RPC SINUMERIK Set bit 4 = 1	
	6:	In request from RPC SINUMERIK Set bit 4 = 0	
	7:	In request from RPC SINUMERIK Set bit 5 = 1	
	8:	In request from RPC SINUMERIK Set bit 5 = 0	

Mode 5 to 8 can be used to switch special functions on and off on a project-specific basis.

Example

```
C_MODE_M ("FLR1", "BAZ3", 0, 3);
```

7.14.2 Special mode

Function

In special mode, workpiece holders are not delivered to the machine by means of automatic material flow, and the host computer does not assign a program in response to the workpiece holder arrival message. Workpiece holders can be delivered by means of manual transport, and the operator must select and start the NC program himself.

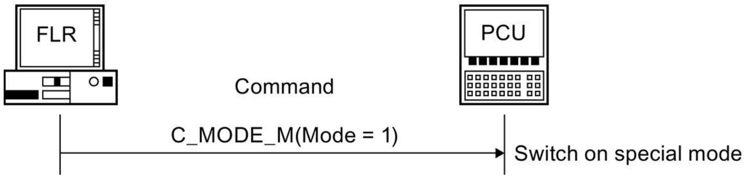
Workpiece holders that were delivered by automatic material flow will also be further transported automatically in special mode.

Workpiece holders that were delivered by manual transport must be transported away manually.

Special mode for a machine may be activated at any time on the host computer. An active machining process on the machine is finished in the normal way.

As soon as special mode is activated, the workpiece holder to be machined in special mode may be delivered with manual transport. The machine does not have to be run until empty.

Mode = 1 Activate special mode



7.14.3 Switching off/switching on components

Function

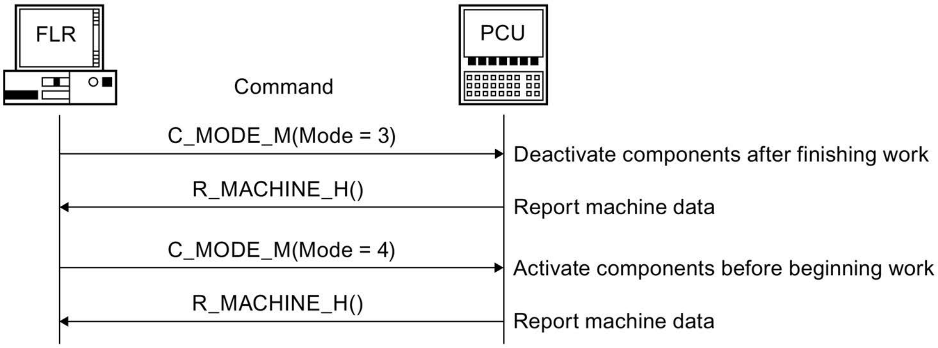
To enable drives or other plant components to be switched off at the end of work by the host computer, a corresponding request is needed. Likewise, a switch-on request is needed to start work.

When the host computer requests the switch-off – at a particular time or only after the last workpiece holder is machined – is irrelevant for the computer link. It only forwards the request to the PLC via the DB interface. The PLC must verify that machining is not in progress before it switches off the components.

Once the components are switched off, this must be reported to the host computer with R_MACHINE_H (). The same applies when switching on the components again.

Mode = 3 Switch off components

Mode = 4 Switch on components



7.15 Synchronization

7.15.1 Overview

Synchronization means the supplying of current data to the host computer so that the system image in the computer matches reality.

Synchronization is necessary following a restart of the computer or machine or after an interruption in the connection.

During synchronization, the machine may not start a new machining operation. Active machining operations are not affected by the synchronization.

Function

The host computer requests machine status data from the machine with `T_MACHINE_M()` and machine assignment data with `T_DATA_M(SFkt = 50)` and transfers the program assignment `R_NC4WPC_M()` for all sides for the workpiece holders whose machining is not yet finished.

If the host computer is also responsible for tool provision/balancing, synchronization of the machine after an (extended) interruption in the connection should be followed by a scan of all tools, so that the host computer has the current tool data again.

The tool scan does not occur automatically. The host computer must request it when needed using `T_DATA_M(SFkt =20).`

7.15.2 Starting/exiting synchronization

Function

Called function

```
C_SYNCH_M(             Host,
                       Machine,
                       OrderNum,
                       SynchFlag)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

The host computer starts or ends a synchronization operation.

Data

Table 7- 26 Parameters of the synchronization

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
SynchFlag	Start /end identifier 1: Start 0: End	4 bytes (long int)

Notes on use

The following section explains the interaction of the sequences of synchronization.

Example

```
C_SYNCH_M ("FLR1", "BAZ3", 0, 1);
```

7.15.3 Synchronization sequence

Sequence

Host computer → SINUMERIK	`C_SYNCH_M()`	SynchFlag = 1	//Start
Host computer → SINUMERIK	`T_MACHINE_M()`		
Host computer ← SINUMERIK	`R_MACHINE_H()`		

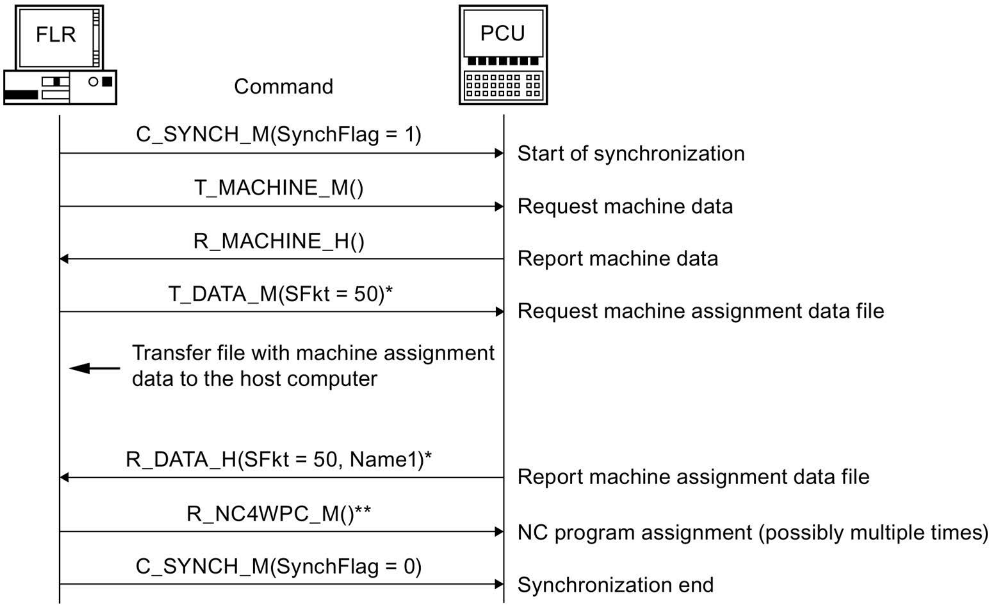
For machines with more than 3 docking positions, such as a transport system, the sequence continues with:

Host computer → SINUMERIK	`T_DATA_M()`	SFkt = 50
Host computer ← SINUMERIK	Transfer machine assignment data file	

Host computer ← SINUMERIK	`R_DATA_H()`	SFkt = 50
		Name1 = File name

For all workpiece holders whose machining is not yet complete and all their sides, the sequence continues with (once each):

Host computer → SINUMERIK	Program assignment `R_NC4WPC_M()`		
Host computer → SINUMERIK	`C_SYNCH_M()`	SynchFlag = 0	//End


FLR
PCU
Command
C_SYNCH_M(SynchFlag = 1)
Start of synchronization
T_MACHINE_M()
Request machine data
R_MACHINE_H()
Report machine data
T_DATA_M(SFkt = 50)*
Request machine assignment data file
Transfer file with machine assignment data to the host computer
R_DATA_H(SFkt = 50, Name1)*
Report machine assignment data file
R_NC4WPC_M()**
NC program assignment (possibly multiple times)
C_SYNCH_M(SynchFlag = 0)
Synchronization end

Steps * : For machines with more than 3 docking positions (e.g. transport system, only for RPC)
Steps ** : For all workpiece holders whose machining has not yet been completed and all their sides (once each)

Configuring data transfer/variable services 8

8.1 Configurable data transfer/variable service

RPC SINUMERIK can read and write any project-specific data from the PLC and NC using the variable services. The is done based on the variable service of the PCU OEM package, using the NC DDE server. The host computer can use `T_VAR_M()` to request data, which is then determined by RPC SINUMERIK and forwarded to the host computer using `R_VAR_H()`. Likewise, the host computer can transfer data using `R_VAR_M()`, which is then written by RPC SINUMERIK.

In the file "SCVARSET.INI" in the "ADD_ON" directory, variable sets can be defined that can be used for read and write operations. When the mode is configured accordingly, changes in these variables are automatically reported to the host computer by RPC SINUMERIK with `R_VAR_H()`.

The following definitions apply to the "SCVARSET.INI" file:

- Maximum of 10 sets can be defined
- Maximum of 10 variables in a set can be defined
- Maximum of 50 variables within all sets can be defined
- Maximum of 10 hotlinks within all sets can be defined

As long as these limits are observed, you can combine as required.

When multiple NCUs are connected to a SINUMERIK controller, RPC SINUMERIK optionally allows the desired NCU to be configured in the variable set. However, this only applies to variable sets that are defined in the "SCVARSET.INI" file and not when the data is described directly with the VarDescr parameter of `T_VAR_M()` or `R_VAR_M()`!

Structure of the "SCVASRSET.INI" file:

Each variable set is a section in the file "SCVASRSET.INI" - and comprises the following elements:

- Section names in square parentheses
- Access mode
- Host name (optional)
- NCU connection (optional)
- Definition of variables

If no host name is configured, after data changes `R_VAR_H()` is sent to all hosts configured for this RPC.

If no NCU connection is configured, the default connection is used.

The name of the variable set and the host name may not exceed 16 characters. NCU connections must match the configurations in the "NETNAMES.INI" file.

Table 8- 1 Access modes

0	No hotlink is set up, the variables of the set are only read at the request of the host computer (`T_VAR_M`).
1*	A hot link is set up for each variable in the set.
2*	Only a hotlink to the first variable of the set is set up.
3*	With handshake. Same as 2, only a hotlink to the first variable of the set is set up. After a hotlink, once all variables have been read, the first variable is reset to 0 (handshake). The variable 1 may not define a field, only a single variable. After the variable is reset to 0, a time delay is required before it can be set again. If the variable is set again too soon, its status change might not be detected.

* In the case of a hotlink, all variables of the corresponding variable set are read and sent to the host computer with `R_VAR_H`.

Table 8- 2 File "SCVASRSET.INI" for RPC SINUMERIK variable service

```
;Name of the variable set
[Measured values]
;Access mode
Mode=2
;Host name (optional)
Host=H1
;NCU connection (optional)
Connection=NCU_1
VAR01=/Plc/DataBlock/Word[c50,0]
VAR02=/Plc/Datablock/Byte[c50,1,20]("!l%d,")
VAR03=/Channel/Parameter/R[5]
[Set02]
Mode=3
Host=FLR2
Var01=/Plc/Datablock/Byte[c50,1]
Var02=/Plc/Datablock/Byte[c50,2]
Var03=/Plc/Datablock/Byte[c50,3,20]("!l%d,")
Var04=/Plc/Datablock/Byte[c50,4]
Var05=/Plc/Datablock/Byte[c50,5,20]("!l%d,")
Var06=/Plc/Datablock/Byte[c50,6]
Var07=/Plc/Datablock/Byte[c50,7,20]("!l%d,")
Var08=/Plc/Datablock/Byte[c50,8]
Var09=/Plc/Datablock/Byte[c50,9,20]("!l%d,")
Var10=/Plc/Datablock/Byte[c50,10]
[Set03]
Mode=0
Var01=/Plc/Datablock/Byte[c51,0,10]("!l%d,")
Var02=/Plc/Datablock/Byte[c51,30,50]("!l%d,")
```

Note

RPC: Test program "DDETEST.EXE"

The SINUMERIK Operate operating software contains a test tool called "DDETEST.EXE". It can be used to test variable access while the NCDDE server is running. In this way, test all variables that you wish to use in the "SCVARSET.INI" file.

You can find the help files "Btss_gr.hlp" for German and "Btss_uk.hlp" for English, for example in the NC-Varselector toolbox of SINUMERIK, or in the OEM programming package. These help files contain all the information required for variable access.

Access to data using $ variables, e.g. `$TC_TP1` (T number), is not implemented with this service.

Note

CMI: "slcaptest.exe" test program

The SINUMERIK Operate operating software contains a test tool called "SLCapTest.exe". It can be used to test variable access while the SLCAP server is running. In this way, test all variables that you wish to use in the "SCVARSET.INI" file.

You can find the help files "Btss_gr.hlp" for German and "Btss_uk.hlp" for English, for example in the NC-Varselector toolbox of SINUMERIK, or in the OEM programming package. These help files contain all the information required for variable access.

Access to data using $ variables, e.g. `$TC_TP1` (T number), is not implemented with this service.

See also

OEM interface HMI <=> NC/PLC (Page 284)

8.2 Transferring data

8.2.1 Variable data transfer to the machine

Function

Called function

```
R_VAR_M(             Host,
                     Machine,
                     OrderNum,
                     VarMode,
                     VarSet,
                     VarDescr,
                     VarData)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

Transfer data from the host computer to RPC SINUMERIK, which is then written to the PLC or NC.

With the `R_VAR_M` call, the name of a variable set from the "SCVARSET.INI" file can be specified for VarSet, or the variable descriptions can be specified for `VarDescr`.

After the data is written, RPC SINUMERIK sends `R_REPORT_H` with `Type = 5` and Error number = 0 as positive acknowledgment for the sender of `R_VAR_M`.

Note

Limit the number of parameters

An error can occur if too much data is transferred for field access between NC and HMI.

Determine the limits for each machine and configure the parameters with several `R_VAR-M` calls.

Data

Table 8- 3 Parameters for variable data transfer to the machine

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
VarMode	Variable mode	4 bytes (long int)
VarSet	Name of variable set	16 bytes (string)
VarDescr *	Data description	1024 bytes (string)
VarData *	User data If more than 10 kB data is written, then the scan/interrogation should be split up between two RPC calls `T_VAR_M()`.	10 KB (string)

* The definition of the separator to separate the values of individual NC variables or to separate individual values for field access operations, is defined as follows when writing variables:

The pipe symbol („|") must be used to separate values of NC variables.

A combination of several NC variables and at the same time array access operations is not possible.

The separator symbol is defined as follows to separate the values for array access operations:

The first character of the data is a separator character, if the first character is not a letter, not a number, not a sign and (-/+) - and not a pipe ("|").

Permissible separators (compatibility): !"#$%&'()*,./:;<=>?@[\]^_`{}~

Recommended separators: ; :

Example, NC variable: VarDescr = „/plc/datablock/byte[c12, 21]|/plc/datablock/byte[c12, 0]"

VarData = "4|1"

VarDescr = "/plc/datablock/byte[c12, 16, #4]"

VarData = ":1:2:3:4"

Example 1

`SET03` is defined in the "SCVARSET.INI" file:

```
Set03]

Mode=1

Host=FLR2

Var01=/Plc/Datablock/Byte[c51,0,10]("!l%d,")

Var02=/Plc/Datablock/Byte[c51,30,50]("!l%d,")

R_VAR_M ("FLR1", "BAZ3", 0, 0, "Set03", "\0", "33|50");
```

Example 2

```
R_VAR_M ("FLR1", "BAZ3", 0, 0, "\0",

 "/Plc/DataBlock/Word[c50,20]|/Plc/DataBlock/Word[c60,30]|", "33|50");
```

8.2.2 Variable data transfer to the host computer

Function

Called function

```
R_VAR_H(              Host,
                      Machine,
                      OrderNum,
                      VarMode,
                      VarSet,
                      VarDescr,
                      VarData)
```

Direction of transfer: Host computer ← SINUMERIK

Meaning

RPC SINUMERIK transfers data read from the PLC or NC to the host computer.

Data

Table 8- 4 Parameters for variable data transfer to the host computer

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
VarMode	Variable mode	4 bytes (long int)
VarSet	Name of variable set	16 bytes (string)
VarDescr *	Data description	1024 bytes (string)
VarData *	User data If more than 10 kB data is requested, then the scan/interrogation should be split up between two RPC calls `T_VAR_M()`.	10 KB (string)

* For both the data description and the user data, the individual variables are separated by the "Pipe" symbol ('|' character code 124).
Variable sets that are defined in the "SCVARSET.INI" file are reported.

Example

```
R_VAR_H ("FLR1", "BAZ3", 0, 0, "Set02", "\0", "33|50");
```

8.3 Requesting data

8.3.1 Request variable data from the machine

Function

Called function

```
R_VAR_M(              Host,
                      Machine,
                      OrderNum,
                      VarMode,
                      VarSet,
                      VarDescr)
```

Direction of transfer: Host computer → SINUMERIK

Meaning

The host computer prompts RPC SINUMERIK to read data from the PLC or NC. The data is then returned with `R_VAR_H()`.

Only one variable set that is defined in the "SCVARSET.INI" file may be specified in `VarSet`.

`VarDescr` is not evaluated.

Data

Table 8- 5 Parameters for request variable data from the machine

Parameter	Description	Format
Host	Name of the host computer	16 bytes (string)
Machine	Machine name	16 bytes (string)
OrderNum	Order number	4 bytes (long int)
VarMode	Variable mode	4 bytes (long int)
VarSet	Name of variable set	16 bytes (string)
VarDescr *	Data description	1024 bytes (string)

* For both the data description and the user data, the individual variables are separated by the "Pipe" symbol ('|' character code 124).

Example

```
T_VAR_M ("FLR1", "BAZ3", 0, 0, "Set02", "\0", "\0");
```

8.3.2 Request variable data from the host computer

Function

Called function

```
R_VAR_H(             Host,
                     Machine,
                     OrderNum,
                     VarMode,
                     VarSet,
                     VarDescr,
                     VarData)
```

Direction of transfer: Host computer ← SINUMERIK

Meaning

RPC SINUMERIK requests data from the host computer. The data is then supplied to RPC SINUMERIK with `R_VAR_M()`.

Note

This RPC is not used by RPC/CMI SINUMERIK at the present time.

---

9 Summary of the RPC calls

Function calls

Table 9- 1 From the host computer to SINUMERIK

Call	Description	Valid in the product
`T_MACHINE_M()`	Forward initiation from "server" to client	RPC/CMI
`T_TPS_M()`	Forward initiation from "server" to client	RPC
`T_REPORT_M()`	Request pending alarms	RPC/CMI
`R_NC4WPC_M()`	Production dialog, initiation to client	RPC/CMI
`R_REPORT_M()`	Transfer report to machine	RPC/CMI
`C_DELETE_M()`	Delete data in the data management	RPC/CMI
`C_MODE_M()`	Set bits in the interface	RPC/CMI
`C_SYNCH_M()`	Set/reset bit in the interface	RPC
`C_TPORDER_M()`	Enter in list, initiative to client	RPC
`C_ORDER_M()`	General order function	RPC/CMI
`T_DATA_M()`	Request data, initiated by host computer	RPC/CMI
`R_DATA_M()`	Request data, initiated by host computer	RPC/CMI
`T_VAR_M()`	Request variable data from the machine	RPC/CMI
`R_VAR_M()`	Send variable data to the machine	RPC/CMI
`R_DDEDATA_M()`	Data transfer to an OEM application	RPC
`R_MESSAGE_M()`	Message to SINUMERIK	RPC

9.1 Function calls from SINUMERIK to the host computer

Function calls

Table 9- 2 From SINUMERIK to host computer

Call	Description	Valid in the product
`T_MACHINE_H()`	Offer machine status data to the host computer	RPC/CMI
`T_TPS_H()`	Offer transport system status data to the host computer	RPC
`T_REPORT_H()`	Transfer messages to the host computer	RPC/CMI
`T_DATA_H()`	Request data as file from the host computer	RPC/CMI
`R_DATA_H()`	Request to host computer: Accept data	RPC/CMI
`T_VAR_H()`	Request variable data from the host computer	RPC
`R_VAR_H()`	Variable data transfer to the host computer	RPC/CMI
`R_DDEDATA_H()`	Data transfer from an OEM application to the host computer	RPC
`R_MESSAGE_H()`	Message to the host computer	RPC/CMI

Note

Direction of the SFkts

Not all subfunction numbers (SFkt) are possible in both directions

Subfunction numbers

Table 9- 3 SFkt for data dialogs

SFkt	Function	Remark
1	NC program	Name1 = NC program Name2 = File name with path on the host computer
7	Transfer NC program Function EES, **only for CMI**	Parameter required, see SFkt=1
7	Accept transferred data without overwriting existing data **only for RPC**	Parameter required, see SFkt=1
8	Transfer NC program to/from PCU 50 / Industrial PC, **only for CMI**	Parameter required, see SFkt=1
10	List of existing NC programs	Name1 = Data storage path Name2 = Name of list file 2
14	NC programs in the EES List memory, **only for CMI**	Parameter required, see SFkt=10

SFkt	Function	Remark
15	List NC programs on the CF card **only for CMI**	Parameter required, see SFkt=10
20	Tool status data of all tools Complete tool data record	Name1 = Empty Name2 = file name with path on the host computer (optional)
21	Tool status data of one tool Variant 1: Complete tool data record	Name1 = tool ID, duplo number Name2 = file name with path on the host computer (optional)
22	Tool status data of one tool Variant 2: Reduced tool data record	Name1 = tool ID, duplo number Name2 = file name with path on the host computer (optional)
23	Tool status data of one tool Variant 3: Reduced tool data record	Name1 = ID number, duplo number Name2 = File name with path on the host computer
24	Data of a tool with adapter number Complete tool data record	Name1 = Adapter number Name2 = File name with path on the host computer
26	Optional/manual loading of a tool Complete tool data set, e. g. for Tool Ident Connection K code 4	Name1 = ID number, duplo number Name2 = File name with path on the host computer
27	Complete tool data set, e. G. for Tool Ident Connection K code 5	Name1 = ID number, duplo number Name2 = File name with path on the host computer
28	Tool loading from tool pallet	Name1 = Tool pallet number Name2 = name of file with tool status data
29	Unload tool to tool pallet	Name1 = Tool pallet number Name2 = name of file with tool status data

31	Tool status data of one tool
	Variant 1: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

32	Tool status data of one tool
	Variant 2: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

34	Tool status data of one tool
	Variant 3: Selected tool data set
		Name1 = tool ID, duplo number
		Name2 = TO area

50	Machine assignment data	Name1 = Empty Name2 = File name with path on the host computer
90	Transfer any file No other processing (optional)	Name1 = Source file name with path Name2 = Target file name with path

Table 9- 4 SFkt for the general order function

SFkt	Function	Remark

2	Load NC program	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1 = function (0, 2, 3)

3	Loading and selection of NC programs	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1 = function (0 - 3)
		Parameter2 = Channel number

4	Unloading of NC programs	Name1 = NC program
		Name2 = NC name

5	Selection of NC programs	Name1 = NC program name
		Name2 = NC name (optional)
		Parameter1 = Function (0, 1)
		Parameter2 = Channel number

6	Deselection of NC programs	Name2 = NC name (optional)
		Parameter2 = Channel number

11	Determining the number of NC channels	No parameters necessary
12	Determining the NC program selected in the NC channel	Parameter2 = NC channel number
13	Determining the assigned TO area of all NC channels of the Sinumerik 840D sl	No parameters necessary

100	Set time/date on SINUMERIK/PLC	Name2 = NC name (optional)
		Parameter1 = UNIX time
		Parameter2 = Date / time SINUMERIK/PLC

200	Set protection level	Name2 = NC name (optional)
		Parameter1 = Protection level

201	Reset protection level	Name2 = NC name (optional)

SINUMERIK-OCX 10

10.1 Overview

Introduction

The SINUMERIK-OCX development packaged represents an add-on product for the RPC SINUMERIK computer link.

The RPC SINUMERIK computer link provides an interface for communication between a SINUMERIK controller and a higher-level host computer. Communication between the host computer and the machine control system takes place in this setup using RPCs (Remote Procedure Call).

Because the RPCs represent a platform-neutral standard, the RPC SINUMERIK interface can be used by MS Windows as well as by UNIX, Linux and other systems.

Range of application

Generally, RPCs can only be used in the C/C++ programming language.

With the use of SINUMERIK-OCX, however, the RPC SINUMERIK interface is accessible for a whole range of widely established Windows development systems without the need for C/C++ programming.

All development systems that are capable of integrating 32-bit ActiveX components are supported.

These include:

- MS Visual Basic V4.0 and higher (32-bit)
- MS Visual J++ 6.0
- Internet Explorer V4.0 and higher
- WinDev and many other development systems

With MCIS RPC 2.0.2.4, Windows 7 as well as both 32-bit and 64-bit host applications are now also supported.

For this purpose, an additional (new) version of the SINUMERIK-OCX as well as an assembly for .NET has been created.

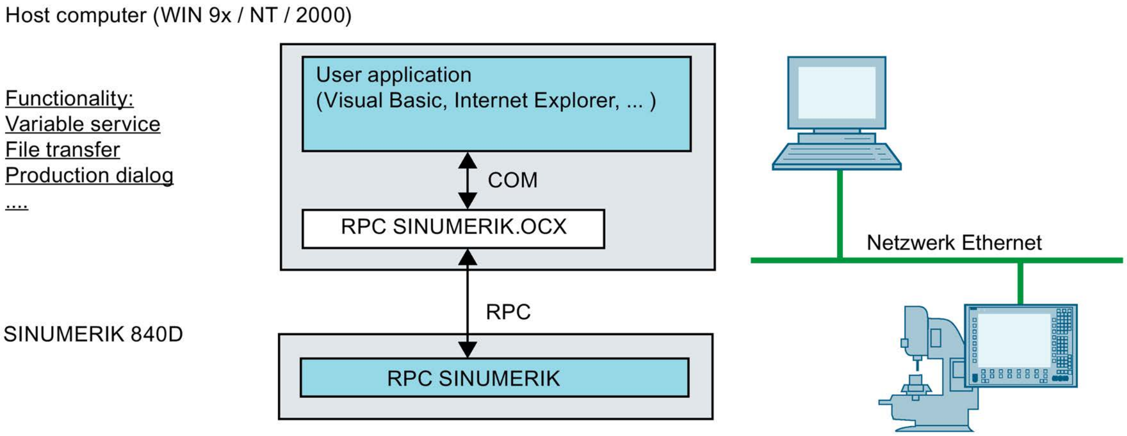
Just as before, it is also possible to communicate with MCIS RPC directly from C++. In this case, it is not necessary to make a distinction between Windows XP and Windows 7 or 32 bit/64 bit.

The "RPC_SINUMERIK_OCX_SETUP.zip" file contains all of the mentioned versions.

Function

The SINUMERIK OCX encapsulates the RPC interface of RPC SINUMERIK in COM calls (Component Object Model).

The mode of functioning of the individual RPCs is not changed by the use of SINUMERIK OCX.



Operating systems

The SINUMERIK OCX can be used on WIN 9x/NT/2000/XP/2003 computers with installed TCP/IP network.

One or more SINUMERIK controllers with RPC SINUMERIK are required as communication partners.

Languages

RPC SINUMERIK generally supports all languages which are covered by the ASCII character set or for which a language package has been installed on the SINUMERIK. Languages that use an extended character set (e.g. Chinese, Russian, etc.) are processed by SINUMERIK in the DBCS format. In this case, data are displayed correctly on the PCU if the application on the host computer sends the data in DBCS format.

10.2 Installing SINUMERIK-OCX

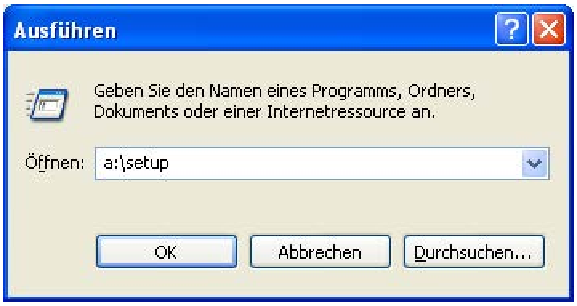
Directories

The following directories are arranged below the target directory:

Directory	Description
Doc	Directory with this documentation
Bin	Directory with the file "rpcSinumerik.OCX" and the application "RpcSinumerikTest.EXE"
RPC SINUMERIK Test	Directory with the source code of the application RPC SINUMERIK Test
Examples	Directory with examples for the use of the SINUMERIK-OCX

Procedure

1. Double-click on "setup.exe".

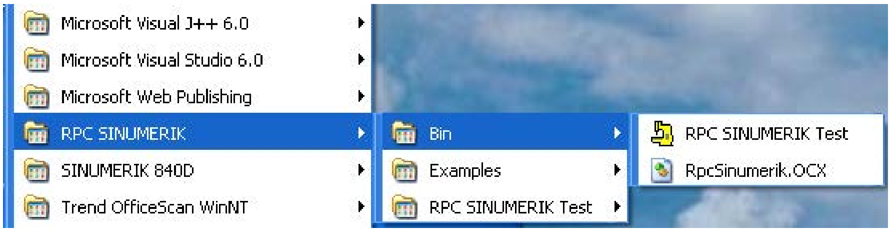


Ausführen

Geben Sie den Namen eines Programms, Ordners, Dokuments oder einer Internetressource an.

Öffnen: a:\setup

2. The above-mentioned directories are created.
3. In addition, Microsoft Visual Basic 6.0 (SP3) runtime system is installed in <Windows>\System - Directory, if not already installed.
   The setup also installs the .NET Framework 4.0 and the VC 2010 runtime on the target system.
4. Open the files that have been installed via the start menu.



Test and examples

After installation, the RPC SINUMERIK interface can be tested with the RPC SINUMERIK Test application using SINUMERIK-OCX.

Requirements for the use of the RPC SINUMERIK Test application can be read in the following Chapter: Test application RPC SINUMERIK Test (Page 166).

After a successful test of the connection to RPC SINUMERIK, the examples described in the following section can be used, see Chapter: Examples for the use of SINUMERIK-OCX (Page 178).

10.3 SINUMERIK-OCX component

10.3.1 Installing SINUMERIK-OCX

Introduction

SINUMERIK-OCX is implemented as a 32-bit ActiveX/COM component. It can be used in all 32-bit Windows development systems that allow the integration of these components.

A separate instance of the SINUMERIK-OCX component must be created in the user application for each connection to the machine control system.

Installation

When the SINUMERIK-OCX development packages are installed, the "RpcSinumerik.OCX" file is stored in <Windows>\System directory and registered in the Windows© Registry.

You can find a additional information on installation in the following section: Installing variants (Page 193).

Other computers

If you want to use the SINUMERIK-OCX component on other computers, it must also be installed there.

This can be done by installation programs, such as InstallShield, or with following manual steps.

1. Copy the "RpcSinumerik.OCS" file from the C:\Programs\Siemens\MCIS\RPC SINUMERIK\bin directory to the <Windows>\System directory of the computer.
2. Register SINUMERIK-OCX in the Windows registry with the following command:
   ```
   Regsvr32 <Windows>\System\RpcSinumerik.OCX
   ```
3. Further, there must be a TCP/IP connection to the RPC SINUMERIK control system.

10.3.2 SINUMERIK-OCX attributes

SINUMERIK-OCX has the following attributes that you use to configure the connection.

Attributes

Table 10- 1 Attributes of SINUMERIK-OCX components

Attributes	Description	Example
MachineID	Name of the machine in the RPC SINUMERIK configuration. User-selectable designation of the machine. This information is transferred in each RPC for identification of the communication partners.	M1
MachineIP	IP network address of the machine control system. This information can be obtained from the Windows network installation. A static IP address assignment is required. In the SINUMERIK-OCX, the network name of the respective computer can be used here as an alternative.	195.2.208.233
MachinePort	Additional information for addressing an application within a computer in the TCP/IP communication. You can freely select the port number in the range 1000 to 64000. Use of port number 3011 is recommended for the machine control systems. This information is designated in the RPC SINUMERIK configuration with Machine-EndPoint.	3011
MachineTimeout	This value is used to influence the timing when sending RPCs to RPC SINUMERIK. If an RPC cannot be delivered to RPC SINUMERIK, e.g. because the controller is not switched on, the time until the RPC call is aborted with an error is defined by the TimeOut value. The attribute may assume a value between 0 and 9. These values are not time values. Rather, they are relative values that are defined in the Microsoft RPC system. 0 - Min TimeOut 5 - Default TimeOut 9 - Max. TimeOut Use the default TimeOUT value (5).	5
HostID	Name of the host computer in the RPC SINUMERIK configuration. User-selectable designation of the host computer. This information is transferred in each RPC for identification of the communication partners.	H1
HostPort	Additional information for addressing an application within a computer in the TCP/IP communication. You can freely select the port number in the range 1000 to 64000. Use of port number 3010 is recommended for the host computer. This information is designated in the RPC SINUMERIK configuration with Host-EndPoint.	3010

Attributes	Description	Example
HostEnabled	This attribute indicates the readiness of the component to receive RPCs The readiness to receive RPCs can be achieved by setting this attribute to True in the user application. The readiness to receive RPCs is also set when at least one RPC has been successfully transferred to RPC SINUMERIK.	True
The attributes apply to all instances. A change in one instance causes a change in all instances.		

The following attributes apply to all instances of the SINUMERIK-OCX within an application (EXE):

- HostID
- HostPort
- HostEnabled

A change in one instance causes a change in all instances.

10.3.3 Methods for sending RPCs to RPC SINUMERIK

The SINUMERIK-OCX encapsulates the RPC interface of RPC SINUMERIK in COM calls (Component Object Model).

An RPC to RPC SINUMERIK is initiated by calling an identically named method to SINUMERIK-OCX.

Example of function

For example, the `RPC Ret = T_MACHINE_M(Host, Machine, OrderNum)` is mapped to one of the instances of the SINUMERIK-OCX using method `T_MACHINE_M`.

```
Ret = Machine1.T_MACHINE_M(OrderNum)
```

The first two parameters – `Host and Machine` – are omitted for all these methods. These parameters are taken from the `HostID` and `MachineID` attributes of the respective instance.

The return values supplied by methods are described in the following section: Error handling (Page 164).

Note

Empty string in different programming languages

Empty strings must be represented in different ways for different programming languages.

- Basic: ""
- C: "\0"

10.3.4 Receiving RPCs from RPC SINUMERIK

Requirement

Enable readiness to receive:

The SINUMERIK-OCX is ready to receive RPCs from RPC SINUMERIK when at least one RPC has been successfully sent to the machine.

- OR -

when the following attribute is set: HostEnabled to True

Function

The RPCs from RPC SINUMERIK are transferred to the application as events of the respective instance of the SINUMERIK-OCX component.

For example, the RPC from RPC SINUMERIK `T_DATA_H ( Host, Machine, OrderNum, SFkt, Name1, Name2 )` is transferred to the application as event `TxDATAxH ( OrderNum, SFkt, Name1, Name2 )`.

Because some development systems (e.g.: Visual Basic) do not permit the "_" character in the name of events, the "_" character in the names of the RPC have been replaced by "x".

10.3.5 Error handling

The error numbers that are supplied by the methods of the SINUMERIK-OCX can be subdivided into two categories:

- Error messages of the Microsoft RPC system
- Error messages of RPC SINUMERIK

Error messages of the Microsoft RPC system

These errors occur when the transfer of an RPC to RPC SINUMERIK or the enabling of the RPC server by the SINUMERIK-OCX cannot be performed. Errors of this category are in the range from 1700 to 1938 (RPC Status Codes).

The errors of this group that occur in practice are described in table "Typical error situations".

For a full description, refer to the Microsoft documentation:

http://msdn.microsoft.com/library/ under Platform SDK > Networking and Directory services > Remote Procedure Calls (RPC)

Error messages of RPC SINUMERIK

These errors occur when an RPC could be properly transferred to RPC SINUMERIK, but correct processing was not possible due to the content of the RPC or the current status of RPC SINUMERIK.

Errors of this category are in the range of negative numbers.

You can read a full description in the appendix in the following section: Error numbers (Page 348)

The following error situations can occur:

Typical error situations

Table 10- 2 Error situations

Error situation	Return value of an RPC method
Incorrect value of attribute: MachineIP	1722 (RPC_S_SERVER_UNAVAILABLE)
Incorrect value of attribute: MachinePort	1722 (RPC_S_SERVER_UNAVAILABLE)
Incorrect value of attribute: MachineID	-100 (ERR_WRONG_MACHINE)
Incorrect value of attribute: HostID	-110 (ERR_WRONG_HOST)
Incorrect value of attribute: HostPort	0, but no response from RPC SINUMERIK An entry is simultaneously created in MCIS_RPCERR.LOG on the controller.
Readiness to receive of the SINUMERIK-OCX component cannot be activated.	1720 (RPC_S_CANT_CREATE_ENDPOINT) This error usually occurs when another application on the same computer is already using the port specified in the HostPort attribute. For example: An attempt is made to start an application that uses SINUMERIK-OCX while the RPC SINUMERIK Test application – with the same HostPort value – is running.
Interruption of the TCP/IP connection	1722 or 1726 (after approx. 20 seconds) After the connection is restored, the RPC calls return 0 again.

10.4 Test application RPC SINUMERIK Test

Requirement

Events are not triggered by the SINUMERIK-OCX within the development environment of Visual Basic if Visual Basic is interrupted by a breakpoint.

The following requirements must be met in order to use RPC SINUMERIK Test:

1. At least one SINUMERIK controller with installed RPC SINUMERIK package must be available.
2. The TCP/IP protocol must be installed on the host computer (Windows PC).
3. There must be a network connection between the control systems and the host computer.

Overview

The RPC SINUMERIK Test application allows you to send or receive individual RPCs of the RPC SINUMERIK interface in the dialog. A connection can be established to multiple controllers.

The application was created with Visual Basic 6.0.

The source code of the application is supplied in the following directory: Siemens\MCIS\RPC SINUMERIK\RPC SINUMERIK Test.

See also section: Sending RPCs to RPC SINUMERIK (Page 174), source code of the RPC SINUMERIK Test application

Languages

RPC SINUMERIK generally supports all languages which are covered by the ASCII character set or for which a language package has been installed on the SINUMERIK.

Languages that use an extended character set (e.g. Chinese, Russian, etc.) are processed by SINUMERIK in the DBCS format. In this case, data are displayed correctly on the PCU if the application on the host computer sends the data in DBCS format.

Configuration

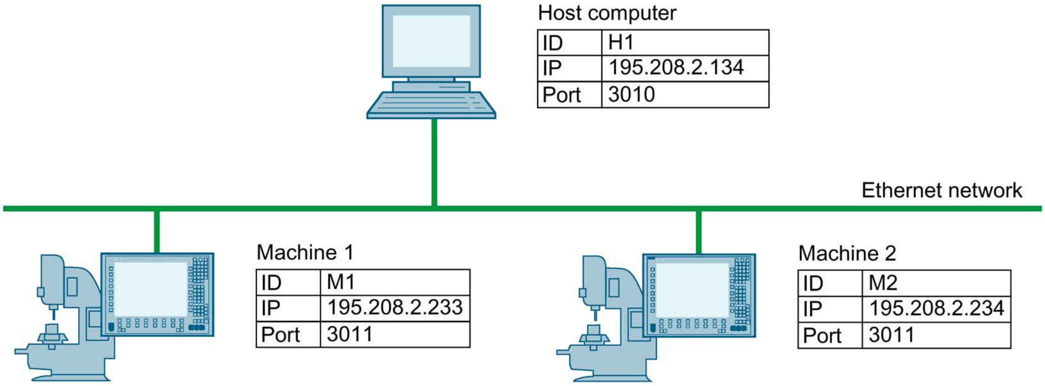
The following information is needed for each of the computers involved:

Table 10- 3 RPC SINUMERIK Test

Information	Description
ID	User-selectable designation of the communication partners. This information is transferred in each RPC for identification of the communication partners.
IP	IP network address of the host computer or controller. This information can be obtained from the Windows network installation. A static IP address assignment is required. In the SINUMERIK-OCX, the network name of the respective computer can be used here as an alternative.
Port	Additional information for addressing an application within a computer in the TCP/IP communication. You can freely select the port number in the range 1000 to 64000. Use of port number 3010 is recommended for the host computer and port number 3011 for the control systems. This information is designated in the RPC SINUMERIK configuration with End-Point.

Example

The configuration is shown using the example of the following network architecture:



Host computer

ID	H1
IP	195.208.2.134
Port	3010

Machine 1

ID	M1
IP	195.208.2.233
Port	3011

Machine 2

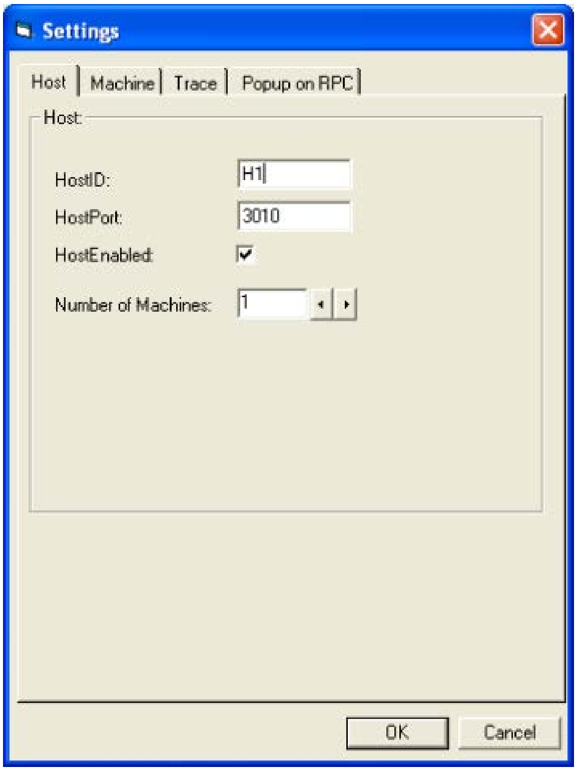
ID	M2
IP	195.208.2.234
Port	3011

The network architecture shown yields the following entries in the configuration windows of RPC SINUMERIK Test.

Procedure

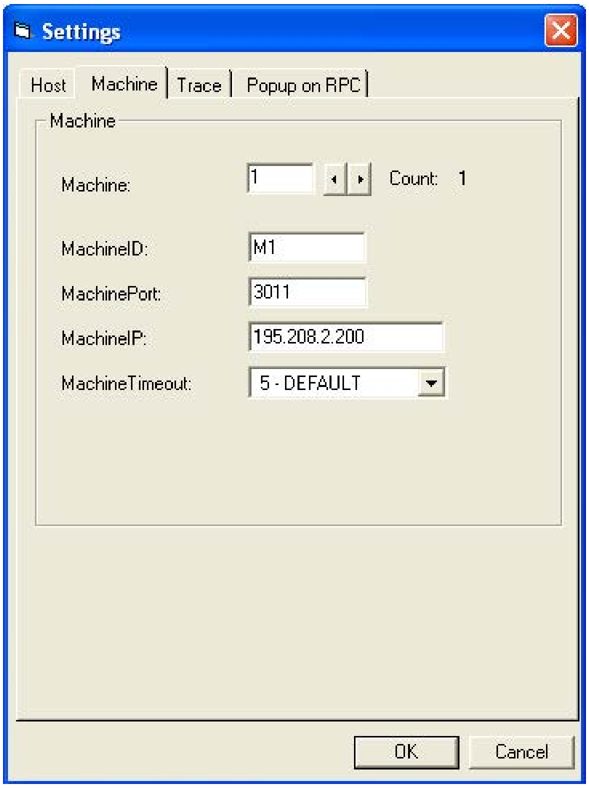
1. Call up the RPC SINUMERIK Test program.
2. Open the "Settings" tab.

3. The "Settings" window opens.
   Click the "Host" tab and enter the appropriate information.



HostID: H1
HostPort: 3010
HostEnabled: ☑
Number of Machines: 1

4. Open the "Machine" tab and enter the appropriate information for Machine 1.


Settings

Host | Machine | Trace | Popup on RPC

Machine

Machine: 1 Count: 1

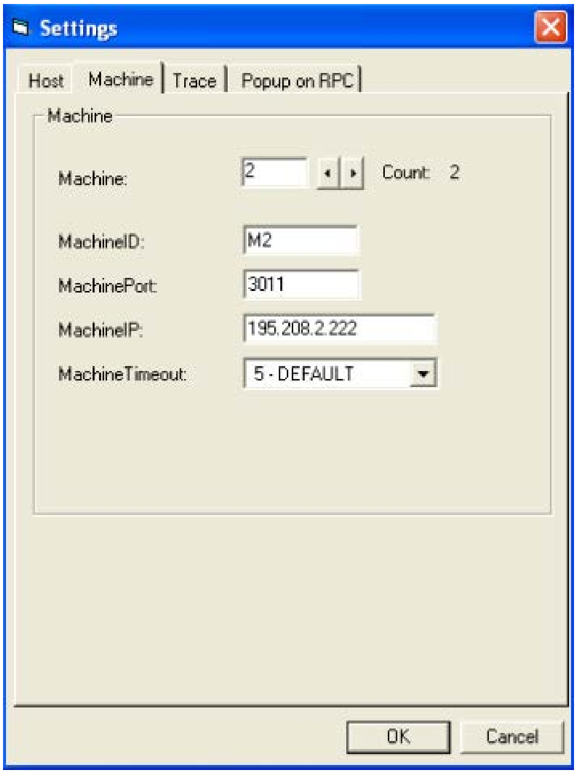
MachineID: M1

MachinePort: 3011

MachineIP: 195.208.2.200

MachineTimeout: 5 - DEFAULT

Enter the appropriate information for Machine 2.


Settings

Host | Machine | Trace | Popup on RPC

Machine

Machine: 2 Count: 2

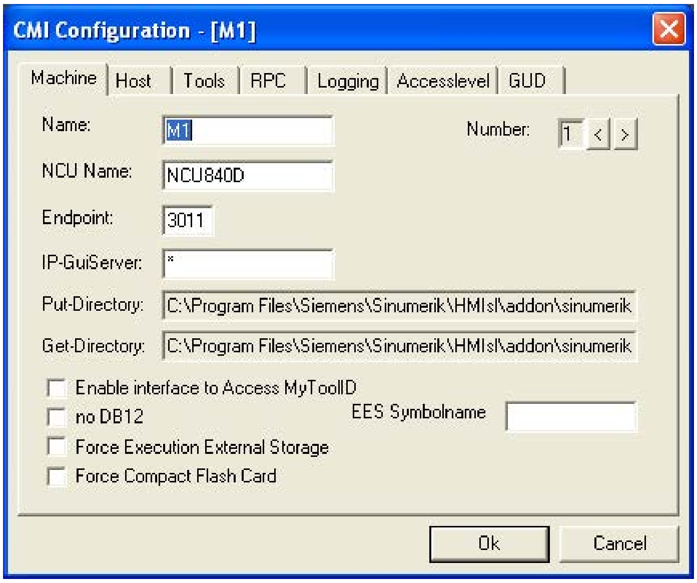
MachineID: M2

MachinePort: 3011

MachineIP: 195.208.2.222

MachineTimeout: 5 - DEFAULT

5. Open the program Create MyInterface Configuration.
   Enter the following information for Machine 1 in line with the network architecture shown.



Name: M1

Number: 1

NCU Name: NCU840D

Endpoint: 3011

IP-GuiServer: *

Put-Directory: C:\Program Files\Siemens\Sinumerik\HMIsl\addon\sinumerik

Get-Directory: C:\Program Files\Siemens\Sinumerik\HMIsl\addon\sinumerik

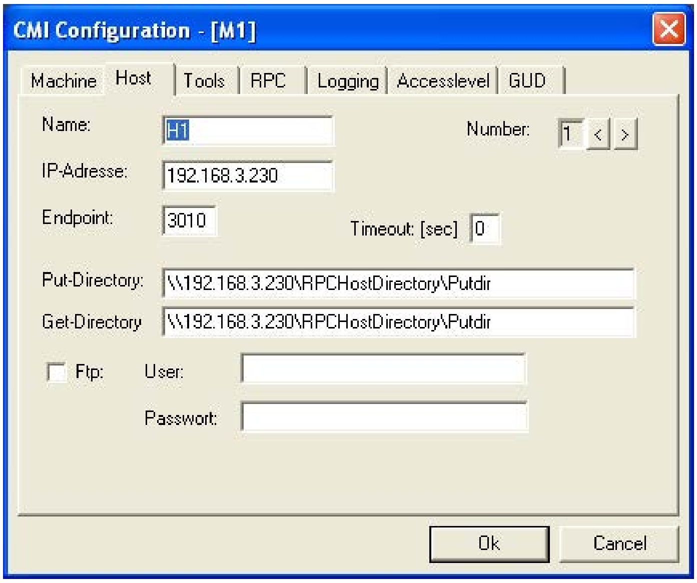
☐ Enable interface to Access MyToolID

☐ no DB12

EES Symbolname

☐ Force Execution External Storage

☐ Force Compact Flash Card



Name: H1

Number: 1

IP-Adresse: 192.168.3.230

Endpoint: 3010

Timeout: [sec] 0

Put-Directory: \\192.168.3.230\RPCHostDirectory\Putdir

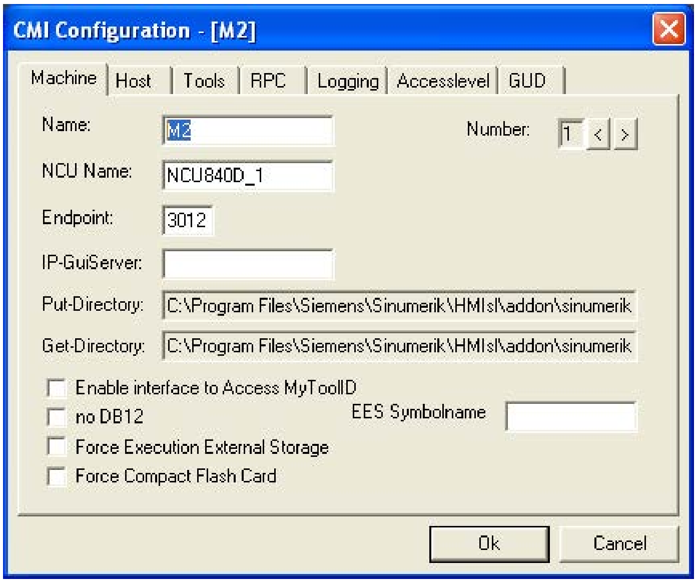
Get-Directory \\192.168.3.230\RPCHostDirectory\Putdir

☐ Ftp:

User:

Passwort:

Enter the following information for Machine 2 in line with the network architecture shown.



Name: M2

Number: 1

NCU Name: NCU840D_1

Endpoint: 3012

IP-GuiServer:

Put-Directory: C:\Program Files\Siemens\Sinumerik\HMIsl\addon\sinumerik

Get-Directory: C:\Program Files\Siemens\Sinumerik\HMIsl\addon\sinumerik

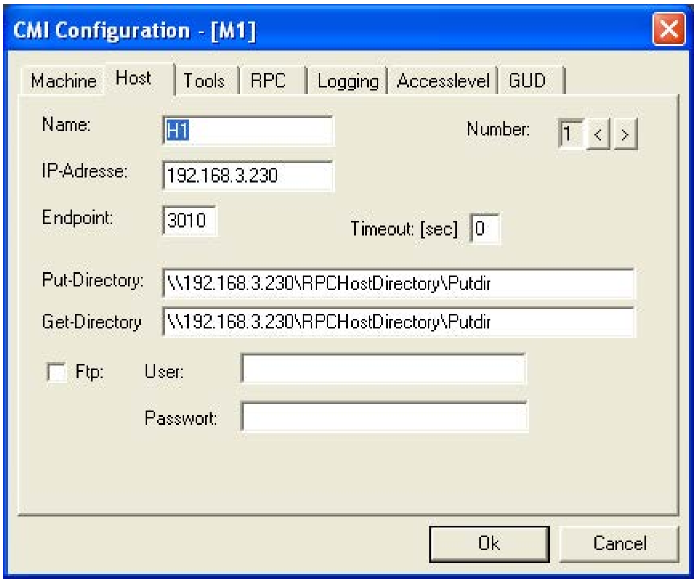
☐ Enable interface to Access MyToolID

☐ no DB12

EES Symbolname

☐ Force Execution External Storage

☐ Force Compact Flash Card



Name: H1

Number: 1

IP-Adresse: 192.168.3.230

Endpoint: 3010

Timeout: [sec] 0

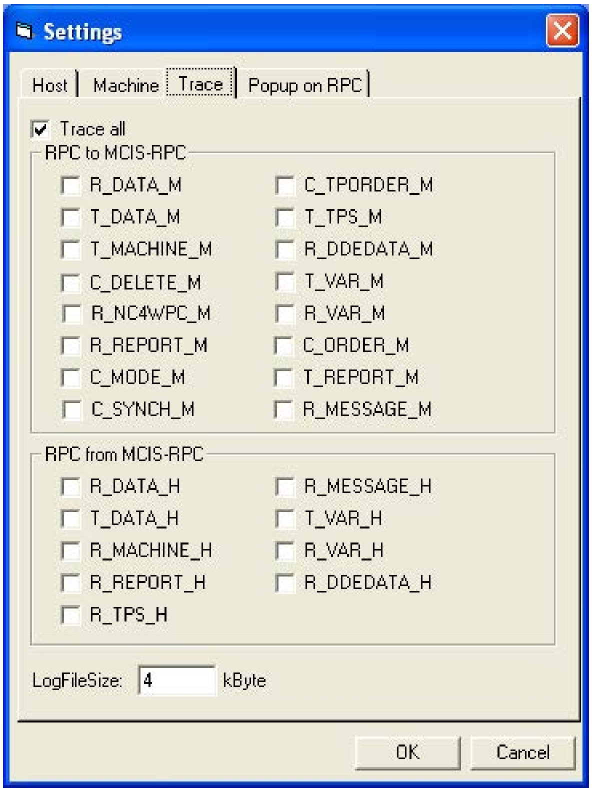
Put-Directory: \\192.168.3.230\RPCHostDirectory\Putdir

Get-Directory \\192.168.3.230\RPCHostDirectory\Putdir

☐ Ftp: User:

Passwort:

6. Open the "Trace" tab and select which RPCs will be logged.
   The log outputs are made on the screen and in the "RpcSinumerikTest.LOG" file.
   Enter the maximum size of the file in the "LogFileSize:" input field.



☑ Trace all

RPC to MCIS-RPC

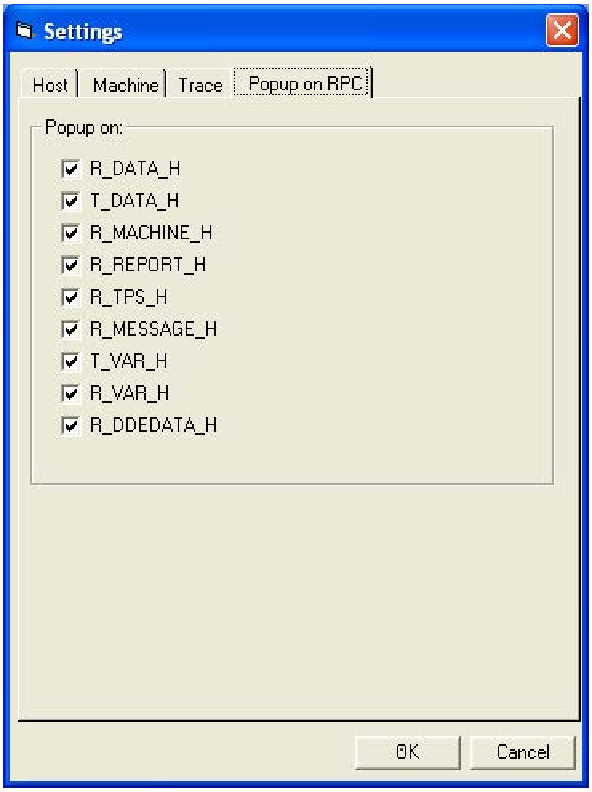
□ R_DATA_M □ C_TPORDER_M
□ T_DATA_M □ T_TPS_M
□ T_MACHINE_M □ R_DDEDATA_M
□ C_DELETE_M □ T_VAR_M
□ R_NC4WPC_M □ R_VAR_M
□ R_REPORT_M □ C_ORDER_M
□ C_MODE_M □ T_REPORT_M
□ C_SYNCH_M □ R_MESSAGE_M

RPC from MCIS-RPC

□ R_DATA_H □ R_MESSAGE_H
□ T_DATA_H □ T_VAR_H
□ R_MACHINE_H □ R_VAR_H
□ R_REPORT_H □ R_DDEDATA_H
□ R_TPS_H

LogFileSize: 4 kByte

7. Open the "Popup on RPC" tab to define the reaction to received PRCs.
   The default reaction to a received RPC is to display a dialog window with all data of the RPC.



Popup on:

☑ R_DATA_H
☑ T_DATA_H
☑ R_MACHINE_H
☑ R_REPORT_H
☑ R_TPS_H
☑ R_MESSAGE_H
☑ T_VAR_H
☑ R_VAR_H
☑ R_DDEDATA_H

8. Click the "OK" button to save the inputs.

10.5 Sending/receiving RPCs to/from RPC SINUMERIK

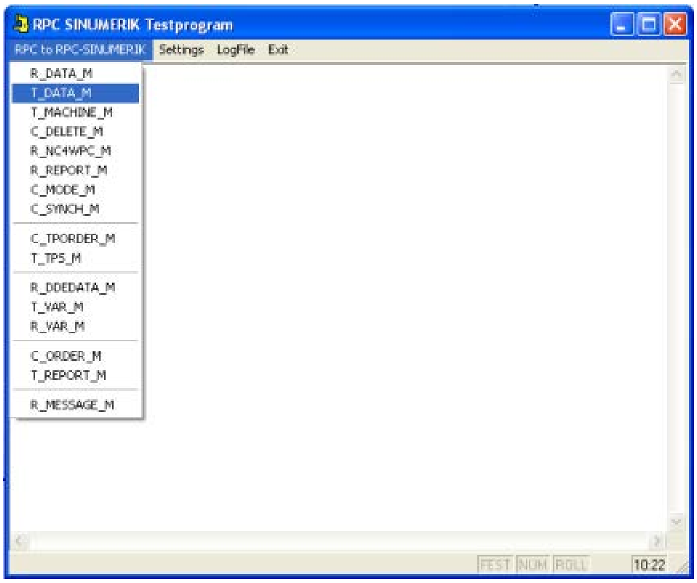
10.5.1 Sending RPCs to RPC SINUMERIK

The RPC SINUMERIK Test application enables you to send all defined RPCs to RPC SINUMERIK via a dialog window. You open the dialog windows using the "RPC to RPC SINUMERIK" menu.

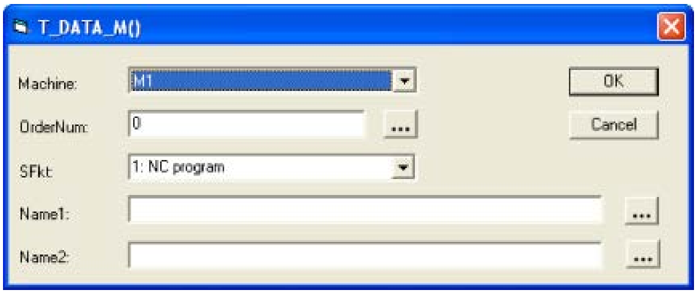
The input fields and the selection from the drop-down lists have the same names as the corresponding RPC parameters.
For the parameters, refer to the Chapters under: Summary of the RPC calls (Page 153)

Procedure

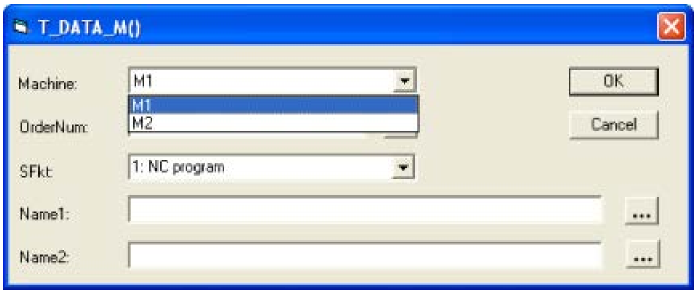
1. Open the "RPC SINUMERIK Test" program.
2. Open the "RPC to RPC SINUMERIK" menu and select e.g. `T_DATA_M`.



3. The "T_DATA_M()" window opens.
   From the "Machine:" drop-down list, select the controller to which the RPC will be sent. The drop-down list contains the communication partners defined in the configuration.

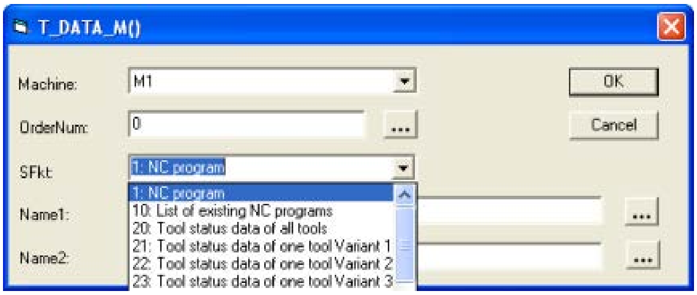


Machine: M1
OrderNum: 0
SFkt: 1: NC program
Name1:
Name2:



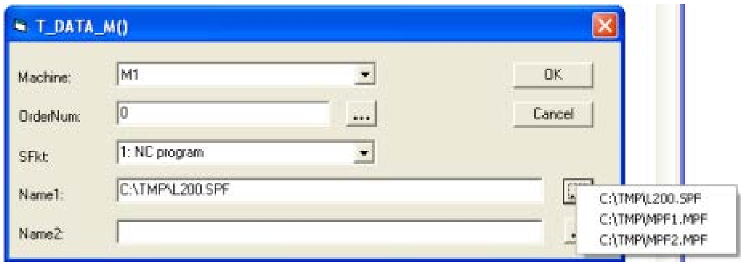
Machine: M1
M1
M2
OrderNum:
SFkt: 1: NC program
Name1:
Name2:

Select the desired predefined values from the "SFkt:" drop-down list.



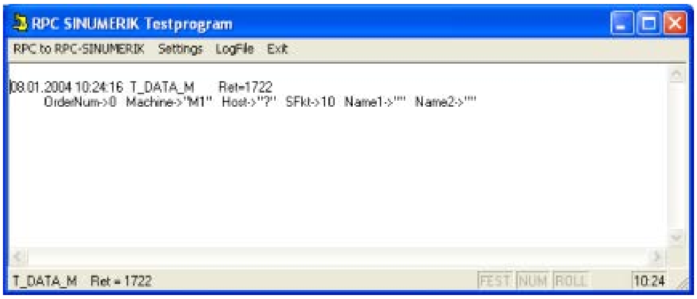
Machine: M1
OrderNum: 0
SFkt: 1: NC program
1: NC program
10: List of existing NC programs
20: Tool status data of all tools
21: Tool status data of one tool Variant 1
22: Tool status data of one tool Variant 2
23: Tool status data of one tool Variant 3
Name1:
Name2:

Click the "..." button to select directories that were already used earlier.



Machine: M1
OrderNum: 0
SFkt: 1: NC program
Name1: C:\TMP\L200.SPF
Name2:
C:\TMP\L200.SPF
C:\TMP\MPF1.MPF
C:\TMP\MPF2.MPF

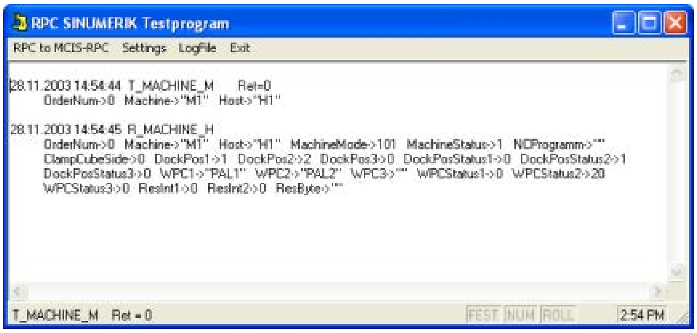
4. Click the "OK" button to send the RPC.
   All sent RPCs are logged. The log output is made on the screen in the main window of the RPC SINUMERIK Test application and in the "RpcSinumerikTest.LOG" log file.
   The return values are also logged and displayed in the status bar for each sent RPC. (Ret=0).

```
08.01.2004 10:24:16  T_DATA_M     Ret=1722
     OrderNum->0  Machine->"M1"  Host->"?"  SFkt->10  Name1->""  Name2->""
```

   T_DATA_M Ret = 1722

5. To test the basic connection to RPC SINUMERIK, the RPC `T_MACHINE_M` can be sent. RPC SINUMERIK then responds with the RPC: `R_MACHINE_H`

```
28.11.2003 14:54:44  T_MACHINE_M     Ret=0
     OrderNum->0  Machine->"M1"  Host->"H1"

28.11.2003 14:54:45  R_MACHINE_H
     OrderNum->0  Machine->"M1"  Host->"H1"  MachineMode->101  MachineStatus->1  NCProgramm->""
     ClampCubeSide->0  DockPos1->1  DockPos2->2  DockPos3->0  DockPosStatus1->0  DockPosStatus2->1
     DockPosStatus3->0  WPC1->"PAL1"  WPC2->"PAL2"  WPC3->""  WPCStatus1->0  WPCStatus2->20
     WPCStatus3->0  ResInt1->0  ResInt2->0  ResByte->""
```

   T_MACHINE_M Ret = 0

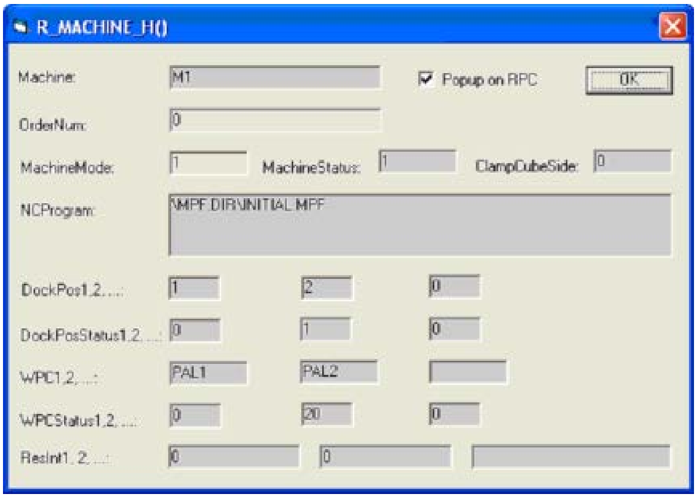
10.5.2 Receiving RPCs from RPC SINUMERIK

Overview

Once the communication partners are configured, the RPC SINUMERIK Test application can already receive RPCs from RPC SINUMERIK when the HostEnabled option is selected.

If the "HostEnabled" option is not activated in the configuration, RPCs are only received from RPC SINUMERIK when at least one RPC has been successfully transferred to RPC SINUMERIK.

The received RPCs are logged. The log output is made on the screen in the main window of the RPC SINUMERIK Test application and in the "RpcSinumerikTest.LOG" log file. In addition, a dialog window is displayed for each RPC with all parameters of the RPC.



R_MACHINE_H()

Machine: M1 ☑ Popup on RPC OK

OrderNum: 0

MachineMode: 1 MachineStatus: 1 ClampCubeSide: 0

NCProgram: \MPF.DIR\INITIAL.MPF

DockPos1,2,...:	1	2	0
DockPosStatus1,2,...:	0	1	0
WPC1,2,...:	PAL1	PAL2	
WPCStatus1,2,...:	0	20	0
ResInt1, 2, ...:	0	0	

The output fields have the same names as the corresponding RPC parameters in the RPC SINUMERIK documentation. For the meaning of these parameters, refer to the sections under: Summary of the RPC calls (Page 153)

The display of the dialog windows for the received RPCs can be activated or deactivated using the "Popup on RPC" check box in the respective window

- OR -

in the configuration ("Popup on RPC" tab).

10.5.3 Source code of the RPC SINUMERIK Test application

Overview

The source code of the application is stored by the setup program in the following directory:

Siemens\MCIS\RPC SINUMERIK\RPC SINUMERIK Test.

The application was created with the Microsoft Visual Basic 6.0 development system.

The application consists of the following modules:

Table 10- 4 Modules of the RPC SINUMERIK Test application

Modules	Description
RpcSinumerikTest.vbp	Visual Basic project file
RpcSinumerikTest.frm **RpcSinumerikTest.frx**	Main window of the application
RpcSinumerikConfig.frm **RpcSinumerikConfig.frx**	Configuration window
History.frm	Selection of the input values
Logen.bas	Logging functions
Util.bas	Auxiliary functions
R_DATA_H.frm **...**	Input or display windows for the corresponding RPCs

10.6 Examples for the use of SINUMERIK-OCX

10.6.1 Introduction

All examples presented in this section use the network configuration shown previously in the following Chapter: Test application RPC SINUMERIK Test (Page 166).

To adapt the examples to your network configuration, the IP addresses must be changed as appropriate in the source code of the examples.

Requirement

RPC SINUMERIK must be installed on the machine control systems and it must be possible to establish a network connection.

Note

C++ applications

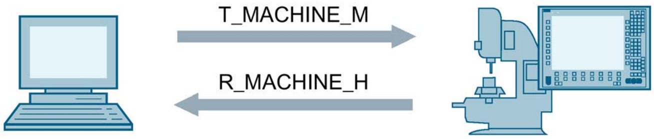
The string terminator "\0" in the examples of the interface description for the host computer is only needed for applications that are created with C++.

10.6.2 Example 1: Machine status query (Visual Basic)

Example of a Visual Basic application, which sends the RPC `T_MACHINE_M()` to the machine - and receives the subsequent response in the form of RPC `R_MACHINE_H()` from RPC SINUMERIK.

The use of the RPCs is described in the RPC SINUMERIK documentation in the following Chapters: Sending machine status data (Page 82).

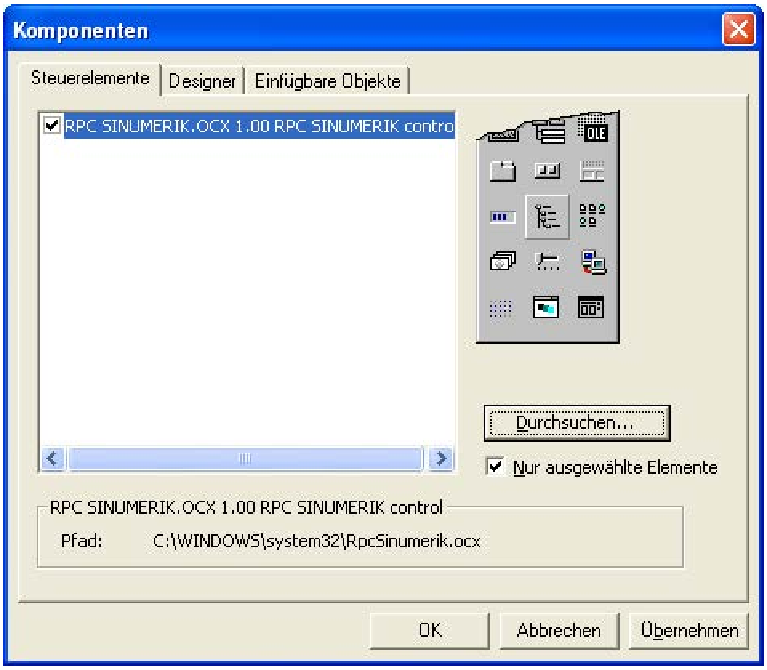
All steps that are needed within the Visual Basic development environment to create the application are shown.



Integrating SINUMERIK-OCX components into Visual Basic 6.0

Before the SINUMERIK-OCX component can be used within Visual Basic, it must be made known there.

This is done using menu: Project > Components.



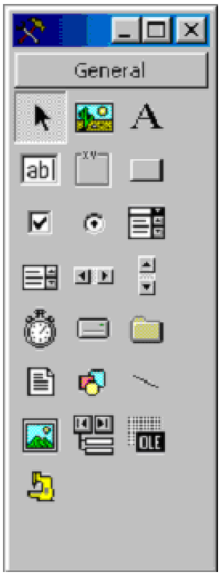
☑ RPC SINUMERIK.OCX 1.00 RPC SINUMERIK contro

☑ Nur ausgewählte Elemente

RPC SINUMERIK.OCX 1.00 RPC SINUMERIK control

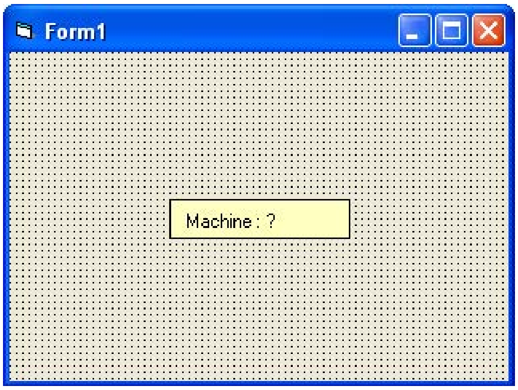
Pfad: C:\WINDOWS\system32\RpcSinumerik.ocx

After that, the symbol (yellow machine) of the SINUMERIK-OCX appears in the Toolbox window.

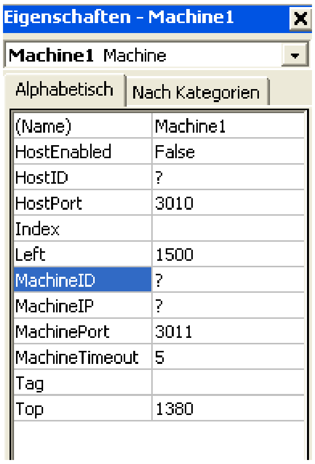


Next, you place SINUMERIK-OCX in a form using drag-and-drop. The component is visible within the development environment.

However, it is no longer visible at runtime of a finished application.



You can save the attributes (properties) in the "Properties" window. Enter the IP address of the machine control system corresponding to your particular network configuration.



Eigenschaften - Machine1

Machine1 Machine

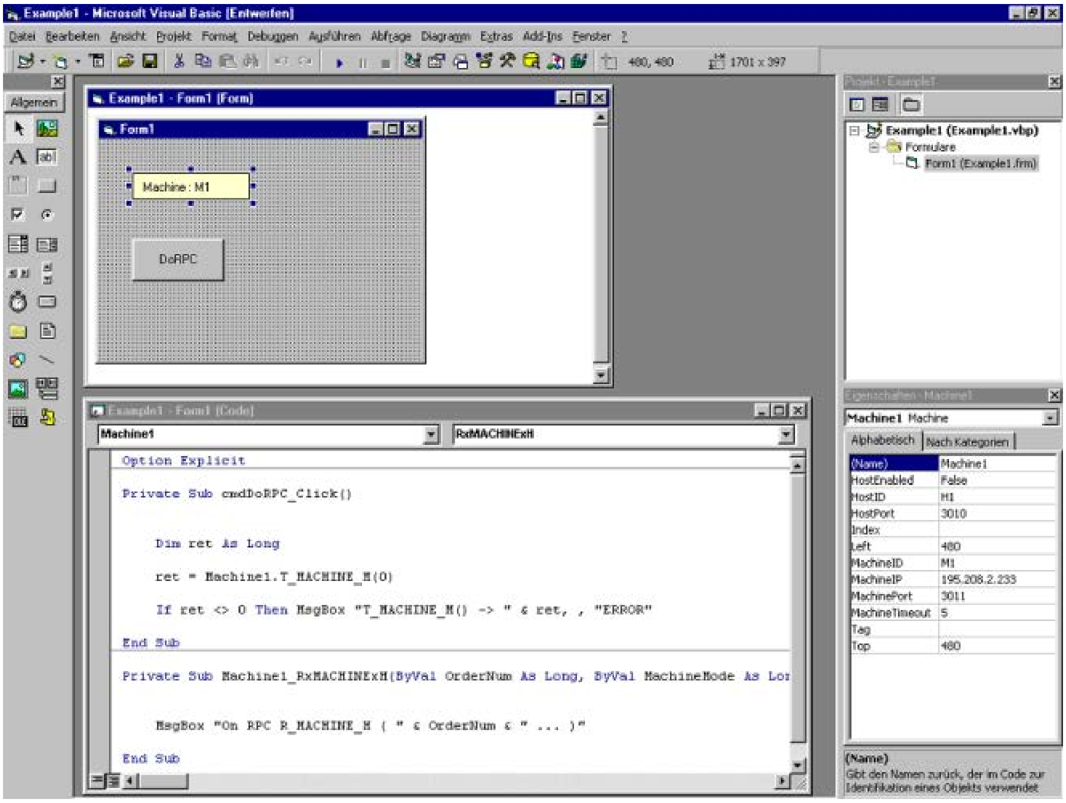
Alphabetisch | Nach Kategorien

(Name)	Machine1
HostEnabled	False
HostID	?
HostPort	3010
Index	
Left	1500
MachineID	?
MachineIP	?
MachinePort	3011
MachineTimeout	5
Tag	
Top	1380

With the change of the "MachineID" attribute, the display of the SINUMERIK OCX in the form changes.

Click the "DoRPC" button to start sending the `T_MACHINE_M`.

The RPC SINUMERIK application responds with the RPC `R_MACHINE_H`. An information field is displayed in the example application in reaction to this.



```
Option Explicit

Private Sub cmdDoRPC_Click()

    Dim ret As Long

    ret = Machine1.T_MACHINE_M(0)

    If ret <> 0 Then MsgBox "T_MACHINE_M() -> " & ret, , "ERROR"

End Sub

Private Sub Machine1_RxMACHINExH(ByVal OrderNum As Long, ByVal MachineMode As Lor

    MsgBox "On RPC R_MACHINE_M ( " & OrderNum & " ... )"

End Sub
```

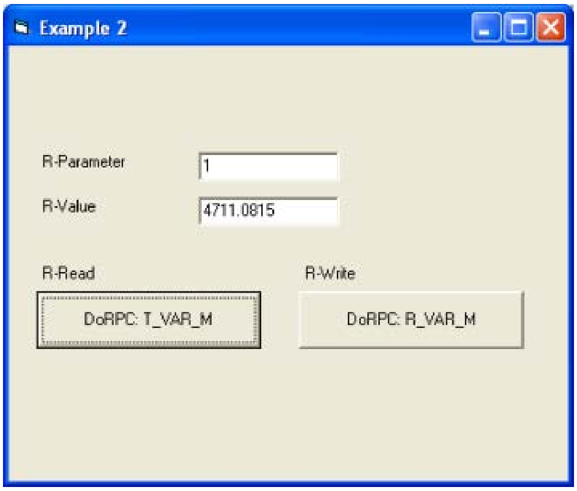
(Name)	Machine1
HostEnabled	False
HostID	M1
HostPort	3010
Index	
Left	480
MachineID	M1
MachineIP	195.208.2.233
MachinePort	3011
MachineTimeout	5
Tag	
Top	480

(Name)
Gibt den Namen zurück, der im Code zur Identifikation eines Objekts verwendet

10.6.3 Example 2: Reading and writing of R parameters (Visual Basic)

Introduction

In this example, the use of the RPC SINUMERIK variable service is explained using the R parameters as example. You can read a full description of the variable service in the following Chapter: Configurable data transfer/variable service (Page 145)



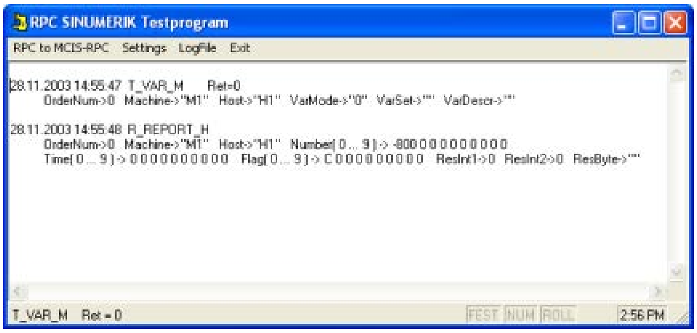
R-Parameter 1

R-Value 4711.0815

In order to activate the variable service of RPC SINUMERIK, at least one variable set must be defined in the c:\add_on\scvarset.ini file on the control system.

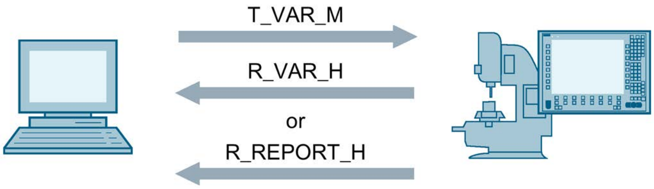
Changes in this file only take effect after a restart of the controller.

If this is not the case, RPC SINUMERIK responds with the RPC `R_REPRT_H()` and the error -800.



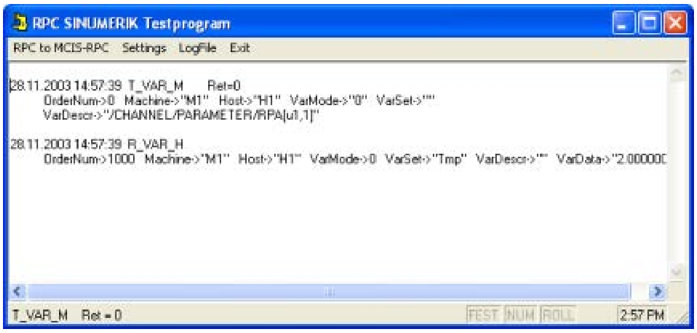
```
28.11.2003 14:55:47  T_VAR_M      Ret=0
      OrderNum->0  Machine->"M1"  Host->"H1"  VarMode->"0"  VarSet->""  VarDescr->""

28.11.2003 14:55:48  R_REPORT_H
      OrderNum->0  Machine->"M1"  Host->"H1"  Number[ 0 ... 9 ]-> -800 0 0 0 0 0 0 0 0 0
      Time[ 0 ... 9 ]-> 0 0 0 0 0 0 0 0 0 0   Flag[ 0 ... 9 ]-> C 0 0 0 0 0 0 0 0 0   ResInt1->0  ResInt2->0  ResByte->""
```

Flowchart for reading of R parameters



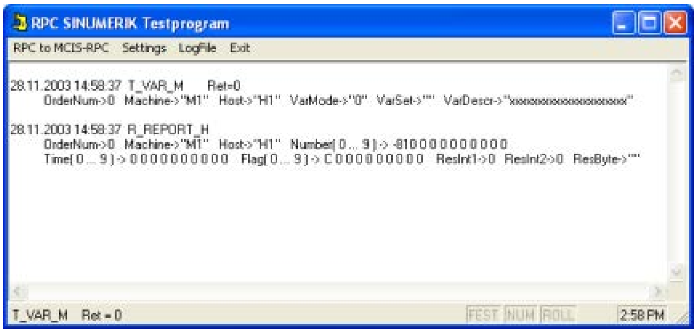
The Visual Basic application requests the value of an R parameter with the RPC `T_VAR_M()`.

RPC SINUMERIK supplies the current value of the R parameter with the RPC `R_VAR_H()`.



```
28.11.2003 14:57:39  T_VAR_M      Ret=0
      OrderNum->0  Machine->"M1"  Host->"H1"  VarMode->"0"  VarSet->""
      VarDescr->"/CHANNEL/PARAMETER/RPA[u1,1]"

28.11.2003 14:57:39  R_VAR_H
      OrderNum->1000  Machine->"M1"  Host->"H1"  VarMode->0  VarSet->"Tmp"  VarDescr->""  VarData->"2.00000C
```

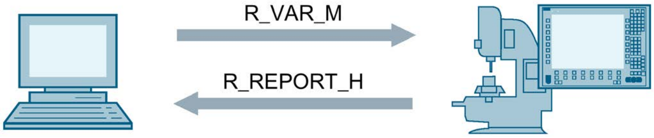
If an error occurs, RPC SINUMERIK responds with the RPC `R_REPORT_M()`.



```
28.11.2003 14:58:37 T_VAR_M      Ret=0
      OrderNum->0  Machine->"M1"  Host->"H1"  VarMode->"0"  VarSet->""  VarDescr->"xxxxxxxxxxxxxxxxxxx"

28.11.2003 14:58:37 R_REPORT_H
      OrderNum->0  Machine->"M1"  Host->"H1"  Number( 0 ... 9 )-> -81 0 0 0 0 0 0 0 0 0
      Time( 0 ... 9 )-> 0 0 0 0 0 0 0 0 0 0  Flag( 0 ... 9 )-> C 0 0 0 0 0 0 0 0 0  ResInt1->0  ResInt2->0  ResByte->""
```

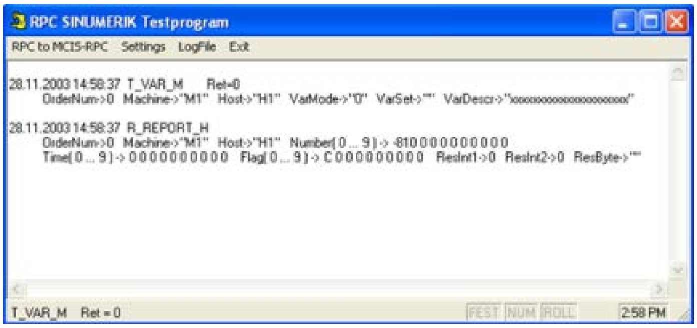
```
T_VAR_M   Ret = 0                                        FEST NUM ROLL   2:58 PM
```

Flowchart for the writing of R parameters



The Visual Basic application transfers the data of an R parameter with the RPC: `R_VAR_M()` to RPC SINUMERIK.

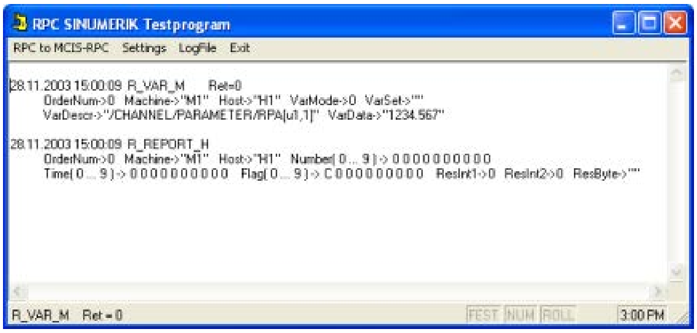
RPC SINUMERIK confirms the write operation with the RPC: `R_REPORT_H()`.



```
28.11.2003 14:58:37 T_VAR_M      Ret=0
      OrderNum->0  Machine->"M1"  Host->"H1"  VarMode->"0"  VarSet->""  VarDescr->"xxxxxxxxxxxxxxxxxxx"

28.11.2003 14:58:37 R_REPORT_H
      OrderNum->0  Machine->"M1"  Host->"H1"  Number( 0 ... 9 )-> -81 0 0 0 0 0 0 0 0 0
      Time( 0 ... 9 )-> 0 0 0 0 0 0 0 0 0 0  Flag( 0 ... 9 )-> C 0 0 0 0 0 0 0 0 0  ResInt1->0  ResInt2->0  ResByte->""
```

```
T_VAR_M   Ret = 0                                        FEST NUM ROLL   2:58 PM
```

In the case of an error, the RPC `R_REPORT_H()` is also sent from RPC SINUMERIK. However, parameter number(0) contains the error code.



```
28.11.2003 15:00:09 R_VAR_M      Ret=0
      OrderNum->0  Machine->"M1"  Host->"H1"  VarMode->0  VarSet->""
      VarDescr->"/CHANNEL/PARAMETER/RPA[u1,1]"  VarData->"1234.567"

28.11.2003 15:00:09 R_REPORT_H
      OrderNum->0  Machine->"M1"  Host->"H1"  Number( 0 ... 9 )-> 0 0 0 0 0 0 0 0 0 0
      Time( 0 ... 9 )-> 0 0 0 0 0 0 0 0 0 0  Flag( 0 ... 9 )-> C 0 0 0 0 0 0 0 0 0  ResInt1->0  ResInt2->0  ResByte->""
```

```
R_VAR_M   Ret = 0                                        FEST NUM ROLL   3:00 PM
```

Visual Basic source code

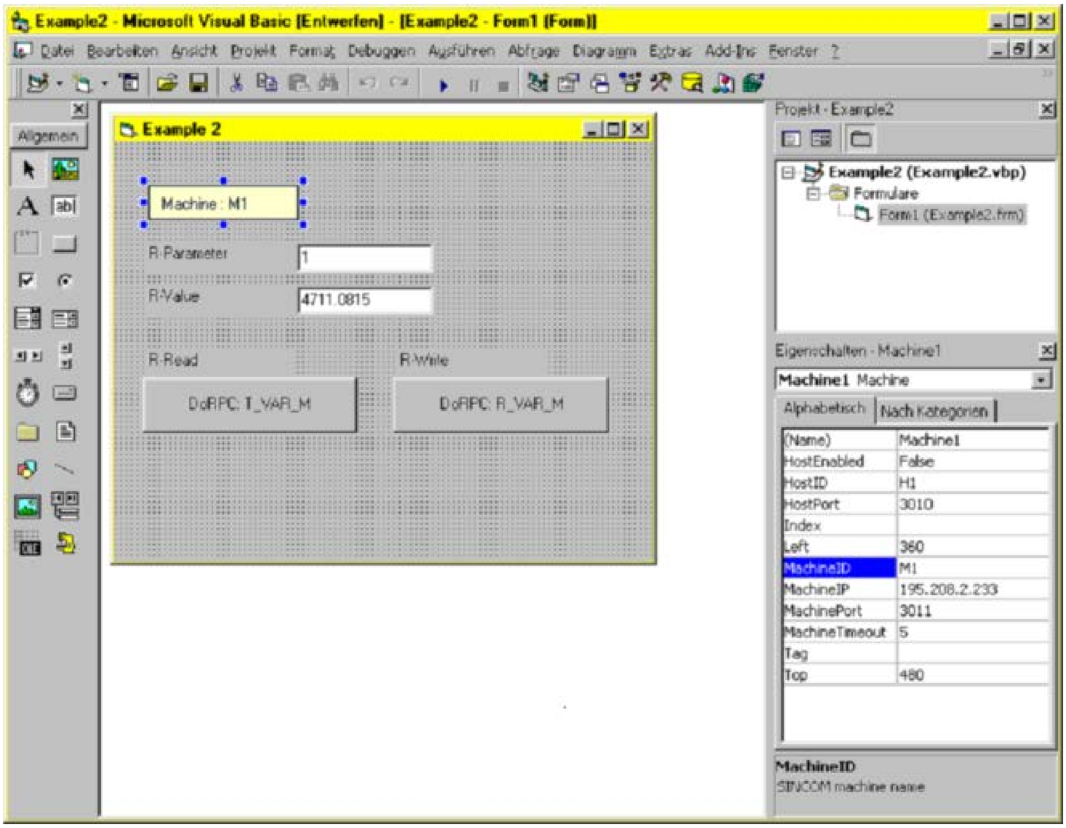
SINUMERIK-OCX is integrated as already described in the following Chapter: Example 1: Machine status query (Visual Basic) (Page 178).

The number of the R parameter for the reading and writing is specified using the "R-Parameter" input field.

The following occurs in the "R-Value" input field:

- When reading – The current value of the R parameter is displayed.
- When writing – the new value will be entered.

The RPCs `T_VAR_M()` and `R_VAR_M()` are sent using the corresponding buttons.



Machine : M1

R-Parameter: 1

R-Value: 4711.0815

Machine1 Machine

(Name)	Machine1
HostEnabled	False
HostID	H1
HostPort	3010
Index	
Left	360
MachineID	M1
MachineIP	195.208.2.233
MachinePort	3011
MachineTimeout	5
Tag	
Top	480

MachineID
SINCOM machine name

File: \Examples\Example2\Example2.frm

```
Option Explicit
Private Sub cmdR_VAR_M_Click()
  'write R parameter
  Dim ret      As Long
  Dim RParam   As Long   ' R parameter number
  Dim RItem    As String ' item for access
  Dim RValue   As String
  RParam = Val(txtRParam.Text)
  RItem = "/Channel/Parameter/R[" & RParam & "]"
  RValue = txtRValue.Text
  ret = Machine1.R_VAR_M(0, 0, "", RItem, RValue)
  If ret <> 0 Then MsgBox "R_VAR_M() -> " & ret, , "ERROR"
End Sub
Private Sub Machine1_RxVARxH(ByVal OrderNum As Long, ByVal VarMode As Long, _ ByVal VarSet
As String, ByVal VarDescr As String, ByVal VarData As String)
  'show R parameter in the form
  txtRValue.Text = VarData
End Sub
Private Sub Machine1_RxREPORTxH(ByVal OrderNum As Long, ByVal Typ As Long, _ ByVal Number
As Variant, ByVal Time As Variant, ByVal Flag As Variant, _ ByVal ResInt1 As Long, ByVal
ResInt2 As Long, ByVal ResByte As String)
  If Number(0) <> 0 Then
  MsgBox "On RPC R_REPEOR_H ( ... Number(0)->" & Number(0) & " )"
  End If
End Sub
```

10.6.4 Example 3 - Active reading of R parameters (Internet Explorer)

Introduction

In this example, active reading of R parameters using MS Internet Explorer is demonstrated.

The active reading function (also called hotlink) enables the SINUMERIK-OCX to be immediately informed by RPC SINUMERIK of each change in the data of a variable set.

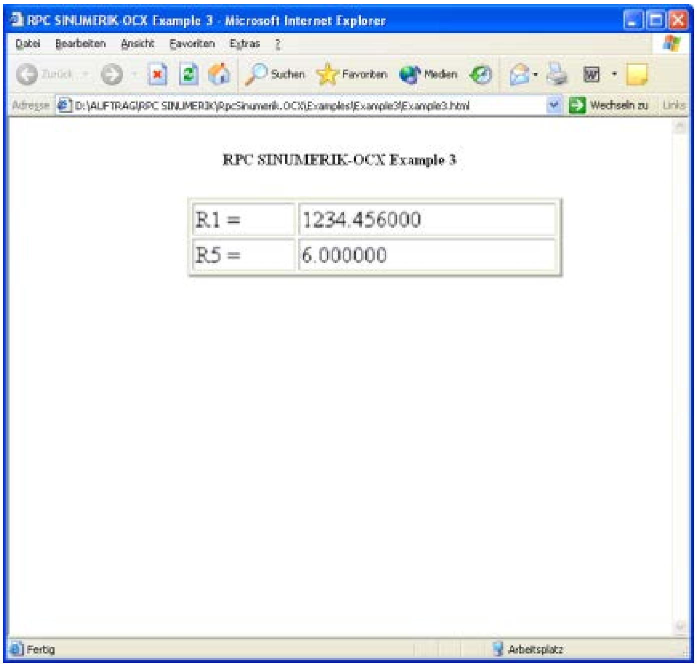
In this example, a variable set named "Set01" consisting of R parameters R1 and R5 is used. The variable set is defined in the file c:\add_on\scvarset.ini on the controller.

Variable set [Set01]

```
[Set01]
Mode=0
Host=FLR1
Var01=/Channel/Parameter/R[1]
Var02=/Channel/Parameter/R[5]
```

Use of MS Internet Explorer requires that SINUMERIK-OCX has already been installed.

File: \Examples\Example3\Example3.html displayed with MS Internet Explorer

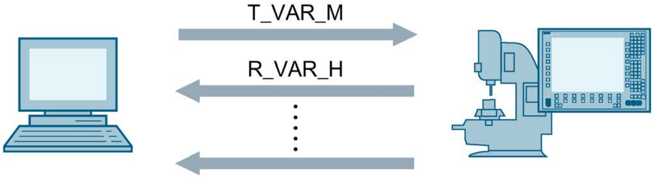


D:\AUFTRAG\RPC SINUMERIK\RpcSinumerik.OCX\Examples\Example3\Example3.html

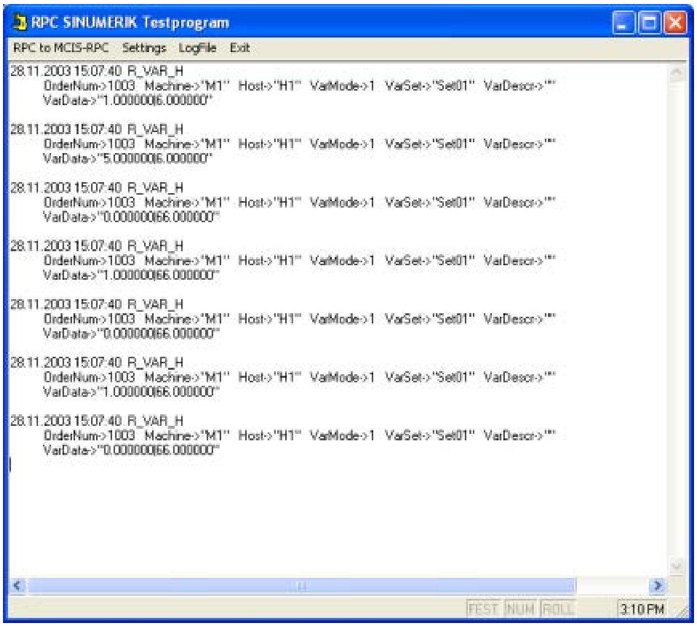
RPC SINUMERIK-OCX Example 3

R1 =	1234.456000
R5 =	6.000000

Flowchart for active reading of R parameters



The RPC `T_VAR_M()` is sent to RPC SINUMERIK at the time the HTML page is loaded using a VBScript function. The actual values of the variables in "Set01" are requested with this RPC. This data is immediately supplied by RPC SINUMERIK with the RPC `R_VAR_H()`



```
28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"1.000000|6.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"5.000000|6.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"0.000000|66.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"1.000000|66.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"0.000000|66.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"1.000000|66.000000"

28.11.2003 15:07:40 R_VAR_H
        OrderNum->1003  Machine->"M1"  Host->"H1"  VarMode->1  VarSet->"Set01"  VarDescr->""
        VarData->"0.000000|66.000000"
```

Source code of the HTML page

The SINUMERIK-OCX is integrated using the <OBJECT> tag in the HTML code.

The attributes of the SINUMERIK-OCX are stored within the <OBJECT> tag.

File: Examples\Example3\Example3.html

```
<HTML>
 <HEAD>
     <TITLE>MCIS_RPC.OCX Example 3</TITLE>
 </HEAD>
 <BODY>
  <OBJECT classid=CLSID:EDF199C1-4F2E-11D3-9DC3-00A0249B4877 id=Machine1>
     <PARAM NAME="MachineID"      VALUE="M1">
     <PARAM NAME="MachineIP"      VALUE="195.208.2.233">
     <PARAM NAME="MachinePort"    VALUE="3011">
     <PARAM NAME="MachineTimeout" VALUE="5">
     <PARAM NAME="HostID"         VALUE="H1">
     <PARAM NAME="HostPort"       VALUE="3010">
  </OBJECT>
     <P align=center><STRONG>MCIS_RPC.OCX Example 3</STRONG> </P>
     <TABLE border=2 align=center width=60% id=TABLE1>
        <TR>
           <TD> R1 = </TD> <TD><LABEL id=R1Param></LABEL> </TD>
        </TR>
        <TR>
           <TD> R5 = </TD> <TD><LABEL id=R5Param></LABEL> </TD>
        </TR>
   </TABLE>
 </BODY>
```

The HTML page contains the following three VBScript functions.

Window_OnLoad	This is called when the HTML page is loaded.
Machine1_RxVARxH	This is called when RPC `R_VAR_H` is received.
Machine1_RxREPORTxH	This is called when RPC `R_REPORT_H` is received.

Continuation of the file: Examples\Example3\Example3.html

```
<SCRIPT LANGUAGE="VBScript">
  Option Explicit
  Sub Window_OnLoad
     dim ret
     ret = Machine1.T_VAR_M(0, 0, "Set01", "")
     if ret <> 0 then MsgBox "T_VAR_M()->" & ret
  End Sub
  Sub Machine1_RxVARxH( OrderNum, VarMode, VarSet, VarDescr, VarData )
     dim pos
     pos = InStr( VarData, "|" )
     if pos = 0 then
          R1Param.innerText = VarData
       else
          R1Param.innerText = Left(VarData, pos-1)
          R5Param.innerText = Mid (VarData, pos+1)
       end If
  End Sub
  Sub Machine1_RxREPORTxH( OrderNum, Typ, Number, Time, Flag, ResInt1,
ResInt2, ResByte )
   If Number(0) <> 0 Then
     MsgBox "On RPC R_REPEOR_H ( ... Number(0)->" & Number(0) & " )"
   End If
  End Sub
 </SCRIPT>
</HTML>
```

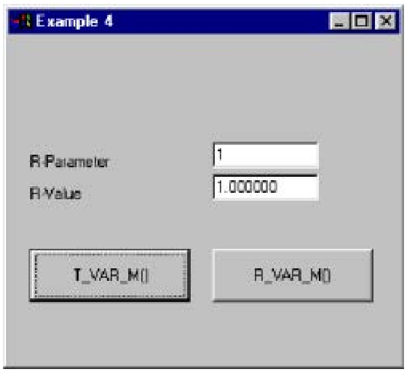
10.6.5 Example 4 - Reading and writing of R parameters (Visual J++)

Introduction

In this example, the same functionality is implemented as in Example 2, except that MS Visual J++ 6.0 SP3 is used.

The flowchart corresponds to that of Example 2.

See section: Example 2: Reading and writing of R parameters (Visual Basic) (Page 181)



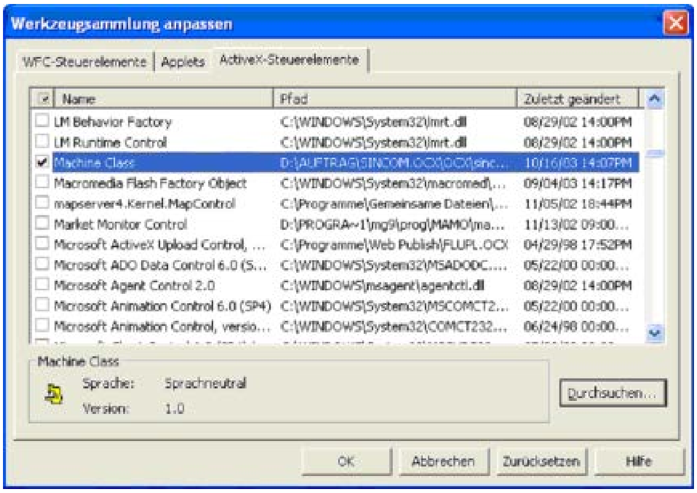
R-Parameter 1
R-Value 1.000000

Integration of RPC SINUMERIK-OCX in MS Visual J++

The MS Visual J++ development environment allows ActiveX components to be used.

The integration is done using the menu Tools > Customize ToolBox > ActiveXControls.

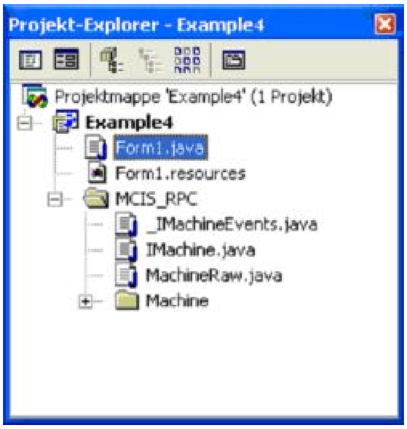
Select the "Machine Class" entry in this window.



☑	Name	Pfad	Zuletzt geändert
☐	LM Behavior Factory	C:\WINDOWS\System32\lmrt.dll	08/29/02 14:00PM
☐	LM Runtime Control	C:\WINDOWS\System32\lmrt.dll	08/29/02 14:00PM
☑	Machine Class	D:\AUFTRAG\SINCOM.OCX\OCX\sinc...	10/16/03 14:07PM
☐	Macromedia Flash Factory Object	C:\WINDOWS\System32\macromed\...	09/04/03 14:17PM
☐	mapserver4.Kernel.MapControl	C:\Programme\Gemeinsame Dateien\...	11/05/02 18:44PM
☐	Market Monitor Control	D:\PROGRA~1\mg9\prog\MAMO\ma...	11/13/02 09:00...
☐	Microsoft ActiveX Upload Control, ...	C:\Programme\Web Publish\FLUPL.OCX	04/29/98 17:52PM
☐	Microsoft ADO Data Control 6.0 (S...	C:\WINDOWS\System32\MSADODC....	05/22/00 00:00...
☐	Microsoft Agent Control 2.0	C:\WINDOWS\msagent\agentctl.dll	08/29/02 14:00PM
☐	Microsoft Animation Control 6.0 (SP4)	C:\WINDOWS\System32\MSCOMCT2...	05/22/00 00:00...
☐	Microsoft Animation Control, versio...	C:\WINDOWS\System32\COMCT232...	06/24/98 00:00...

Machine Class
Sprache: Sprachneutral
Version: 1.0

With the integration of SINUMERIK-OCX, VJ++ generates additional classes in the RPC SINUMERIK directory.



Source code of the Visual-J++ application

The number of the R parameter for the reading and writing is specified using the "R-Parameter" input field.

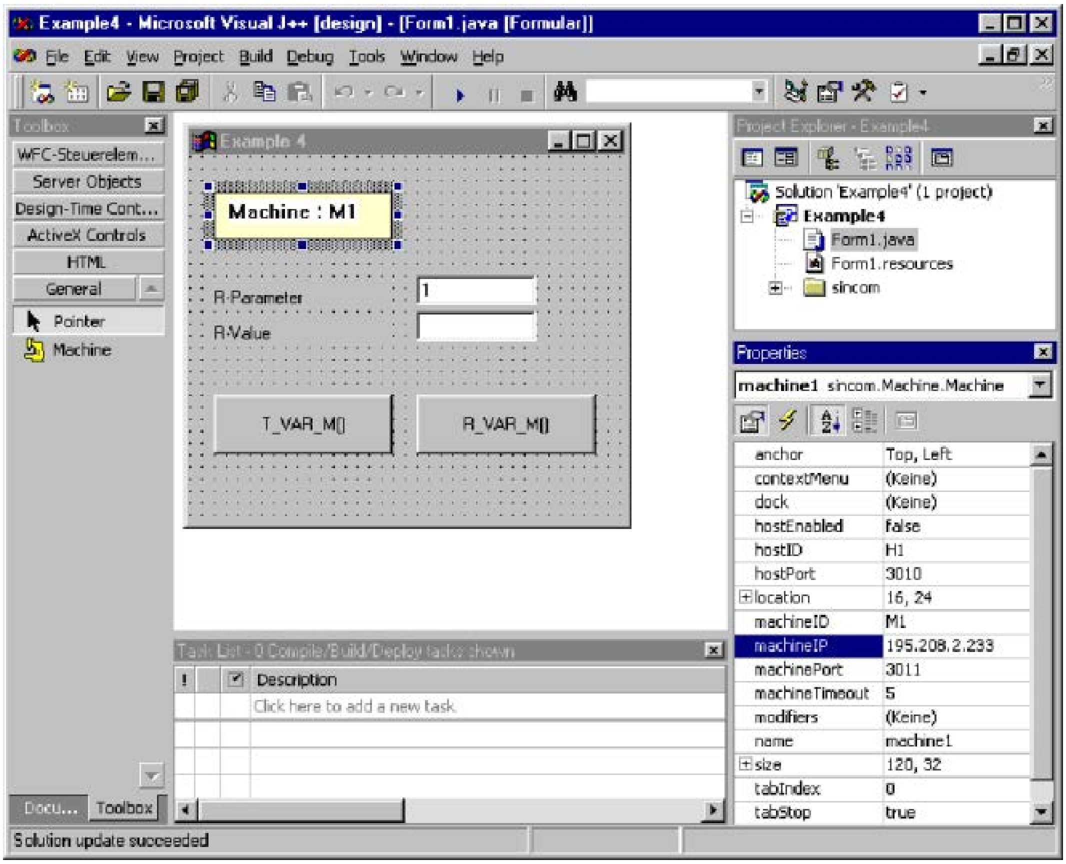
File: Examples\Example4\Form1.java

```
private void cmdT_VAR_M_click(Object source, Event e)
{
   // read R parameter
   long      ret;
   String[]  VarDescr    = new String[1];    // item for access
   String[]  VarSet      = new String[1];
   VarSet[0]   = "";
   VarDescr[0] = "/Channel/Parameter/R[" + txtRParam.getText() + "]";
   ret = machine1.T_VAR_M(0,0,VarSet,VarDescr );
   if ( ret != 0 ) MessageBox.show("T_VAR_M() -> " + ret);
}
private void cmdR_VAR_M_click(Object source, Event e)
{
   // write R parameter
   long      ret;
   String[]  VarDescr    = new String[1];    // item for access
   String[]  VarSet      = new String[1];
   String[] VarData = new String[1];
   VarSet[0]    = "";
   VarDescr[0]  = "/Channel/Parameter/R[" + txtRParam.getText() + "]";
   VarData [0]  = txtRValue.getText();
   ret = machine1.R_VAR_M(0, 0, VarSet, VarDescr, VarData);
   if ( ret != 0 ) MessageBox.show("R_VAR_M() -> " + ret);
}
private void machine1_RxVARxH(Object source, MCIS_RPC.Machine.RxVARxHEvent e)
{
   // show R parameter in the form txtRValue.setText( e.VarData ); } private void ma-
chine1_RxREPORTxH(Object source, MCIS_RPC.Machine.RxREPORTxHEvent e)
{
   int ErrorNr = e.Number.getVariantArray()[0].getInt();
   if ( ErrorNr != 0 )
   }
     MessageBox.show( "On RPC R_REPORT_H ( ... Number(0)->" + ErrorNr + " )" );
   }
}
```

The following occurs in the "R-Value" input field:

- When reading – The current value of the R parameter is displayed.
- When writing – the new value will be entered.

The RPCs `T_VAR_M()` and `R_VAR_M()` are sent using the corresponding buttons.



R-Parameter: 1

machine1 sincom.Machine.Machine

anchor	Top, Left
contextMenu	(Keine)
dock	(Keine)
hostEnabled	false
hostID	H1
hostPort	3010
⊞location	16, 24
machineID	M1
machineIP	195.208.2.233
machinePort	3011
machineTimeout	5
modifiers	(Keine)
name	machine1
⊞size	120, 32
tabIndex	0
tabStop	true

Solution update succeeded

10.7 Support for host computer software development

10.7.1 Usage matrix

WinNT **Host 32 bit**	**WinXP** **Host 32 bit**	**Win7/32** **Host 32 bit**	**Win7/64** **Host 32 bit**	**Win7/64** **Host 64 bit**
Solution for host computer application without .NET				
RPCIDL	RPCIDL	RPCIDL	RPCIDL	RPCIDL
OCX	OCX	---	---	---
---	OCXNEW	OCXNEW	OCXNEW	---
Optional solution for host computer application with .NET				
---	---	---	---	64ASS
OCX	OCX	---	---	---
---	OCXNEW	OCXNEW	OCXNEW	---
Use of test program TP				
TP	TP1	---	---	---
---	---	TP1	TP2	TP2
Explanation:				
RPCIDL	The existing C++ interface			
OCX	The RPC OCX supplied previously with RPC 2.0 powerline (file version 2.0)			
OCXNEW	The new SINUMERIK-OCX (file version 4.0)			
64ASS	The new .NET assembly			
TP1	Test program (VB6 with OCXNEW)			
TP2	Test program (VB .NET 64 bit application with 64ASS)			

The table below provides information about the naming in the file properties.

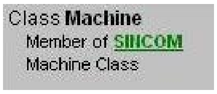
Properties	**OCX**	**OCXNEW**	**64ASS (.NET Assembly)**
File name	RpcSinumerik.ocx	RpcSinumerik.ocx	Siemens.Sinumerik.Rpc.dll
File version	2.0.0.0	4.0.0.0	4.0.0.0
File description	RPC SINUMERIK Control	RPC SINUMERIK Control	Rpc Sinumerik Assembly
Copyright	Copyright 2003	Copyright 2012	Copyright 2012
Product name	RPC SINUMERIK.OCX 1.00 RPC SINUMERIK control	RPC SINUMERIK OCX 3.00	RPC Sinumerik

Note

Name of the COM project class

The existing name of the class of the COM object is not changed.

For compatibility reasons this also remains unchanged in OCXNEW and 64ASS.


Class **Machine**
Member of SINCOM
Machine Class
	`Private Sub Form_Load( )`
	` Dim m As SINCOM.Machine`
	` Call m.R_MESSAGE_M(1, "", 1, 2, "")`
	`EndSub`

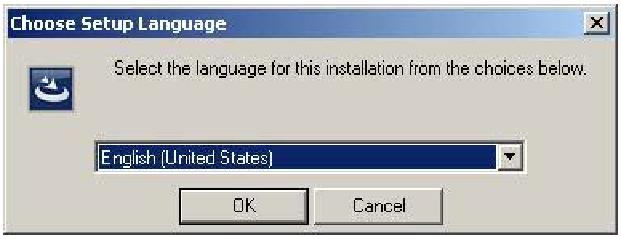
10.7.2 Installing variants

OCX, assembly, examples and the test program are installed using their own setup program.

The "RPC_SINUMERIK_OCX_SETUP.zip" file contains all of the mentioned versions. The file is supplied with Create MyInterface 2.6.2.0, but it is not automatically installed.

Procedure

1. Extract the files to your local hard disk.
2. Start the setup "RPC_SINUMERIK_OCX_SETUP.exe".
   The setup installs the .NET Framework 4.0 and the VC 2010 Runtime, among other things, on the target system.
3. From the drop-down list box, select the language for the installation.
   You can select between German and English.
   Click "OK".

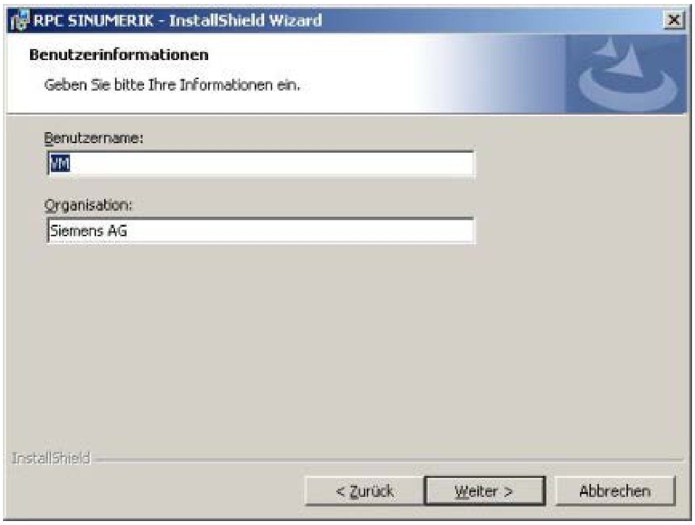
   Choose Setup Language

   Select the language for this installation from the choices below.

   English (United States)

4. The files needed by the setup are unpacked in a temporary directory and the welcome window opens.
   Click "Next"
5. The "License agreement" window opens.
   Select the "I accept the terms of the license agreement..." option button and click "Next >".

6. The "User information" window opens.
   - Enter the user name and organization.
   - Click "Next >".



RPC SINUMERIK - InstallShield Wizard

Benutzerinformationen

Geben Sie bitte Ihre Informationen ein.

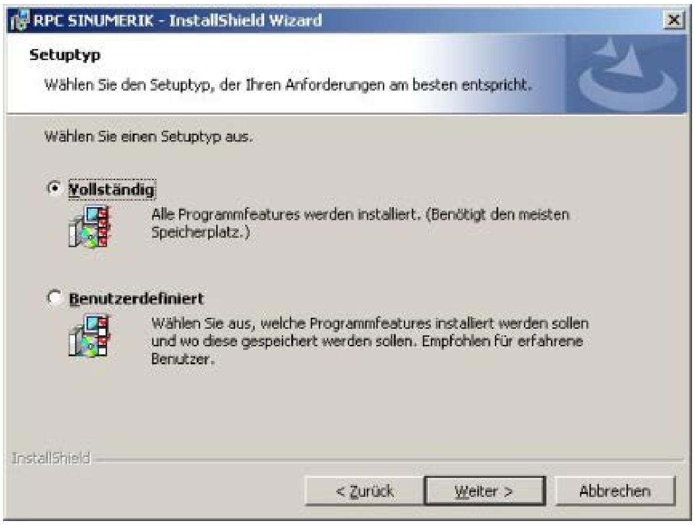
Benutzername: VM

Organisation: Siemens AG

7. The "Setup type" window opens.

   Select the "Complete" option button.
   The following components are installed:
   - Installation directory: "C:\Siemens\MCIS\RPC\OCX"
   - 3 example projects with source code
   - Test program (32-bit version)
   - Test program (64-bit version- only for installation on 64-bit operating systems)

   Click "Next >".



RPC SINUMERIK - InstallShield Wizard

Setuptyp

Wählen Sie den Setuptyp, der Ihren Anforderungen am besten entspricht.

Wählen Sie einen Setuptyp aus.

● **Vollständig**

Alle Programmfeatures werden installiert. (Benötigt den meisten Speicherplatz.)

○ **Benutzerdefiniert**

Wählen Sie aus, welche Programmfeatures installiert werden sollen und wo diese gespeichert werden sollen. Empfohlen für erfahrene Benutzer.

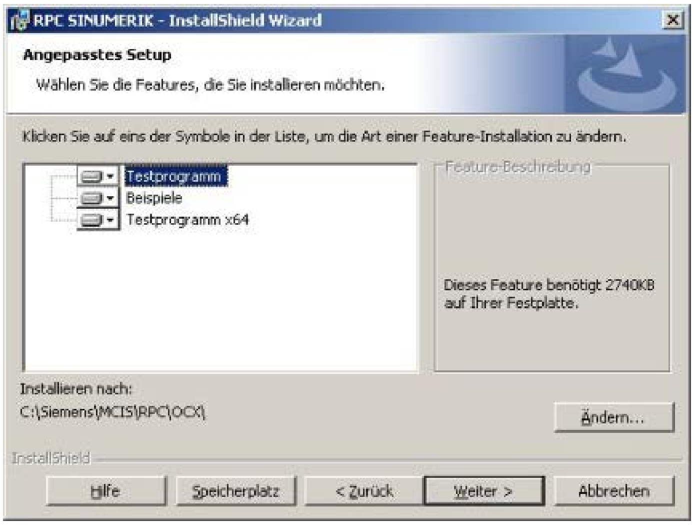
- OR -
Select the "User-defined" option button
You can install the following components:

– Test program

– Examples

– Test program x64 - The test program for 64-bit systems is only installed on 64-bit operating systems.

Click "Next >" to confirm the selection.



RPC SINUMERIK - InstallShield Wizard

Angepasstes Setup

Wählen Sie die Features, die Sie installieren möchten.

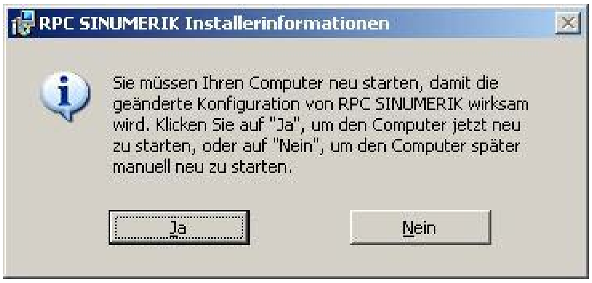
Klicken Sie auf eins der Symbole in der Liste, um die Art einer Feature-Installation zu ändern.

Dieses Feature benötigt 2740KB auf Ihrer Festplatte.

Installieren nach:
C:\Siemens\MCIS\RPC\OCX\

8. The files are copied.
   Click "Finish" to complete the installation.

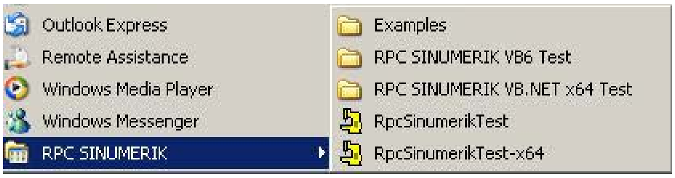
9. You can receive an information window prompting you to restart your computer.
   Click "Yes".



RPC SINUMERIK Installerinformationen

Sie müssen Ihren Computer neu starten, damit die geänderte Konfiguration von RPC SINUMERIK wirksam wird. Klicken Sie auf "Ja", um den Computer jetzt neu zu starten, oder auf "Nein", um den Computer später manuell neu zu starten.

10. You can now open the test programs and the directories for the examples using "Start" > "All Programs"> "RPC SINUMERIK".



Directory structure

You can find the following directory structure on your computer:

Directory	Path
Installation directory	C:\Siemens\MCIS\RPC\OCX
Program files	<Installation directory>\Bin
Directory for examples	Directory for examples: <Installation directory>\ Examples
32-bit test program	Directory for 32-bit test program: <Installation directory>\RpcSinumerik-VB6-Test
64-bit test program	<Installation directory>\RpcSinumerik-VB.NET-x64-Test

Interface and machine PLC 11

11.1 Overview

The interface between the computer link software (RPC SINUMERIK) and the machine PLC is described in the following.

Introduction

An interface DB is required for communication between the RPC SINUMERIK and the PLC.

The standard Siemens DB12 is reserved for this. The DB is set up by the user. The data elements of the DB interface are grouped in blocks that each cover one aspect of the interface, e.g. DB12 global data, docking position data, NC program assignment.

The individual blocks are represented in tabular form. All blocks are placed one after the other in the interface DB.

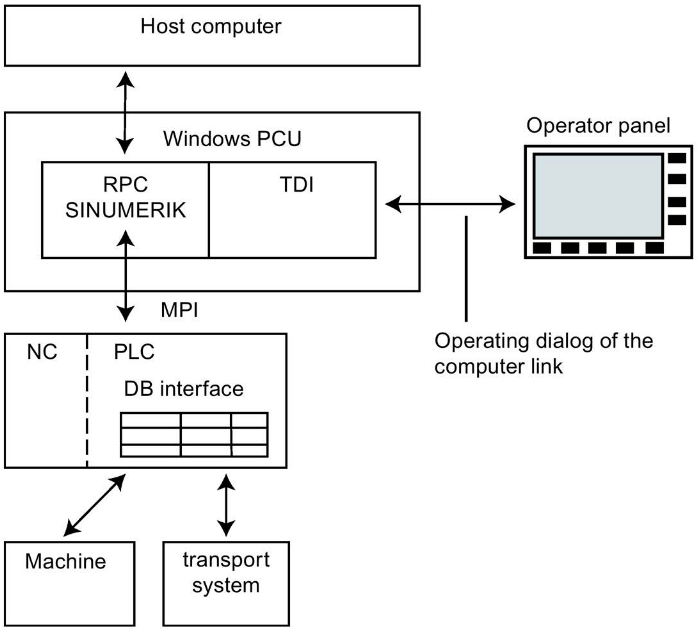
Binary data elements of type "int(WORD)" and "long(DWORD)" are stored in the DB in S7 format (little-endian). When the PLC accesses these elements, they are converted to Intel format (big-endian). Data elements that represent an identifier are implemented as byte fields with ASCII characters.

The interface is written in tabular form. The "Access from" column indicates who writes this field. The following abbreviations are used for this:

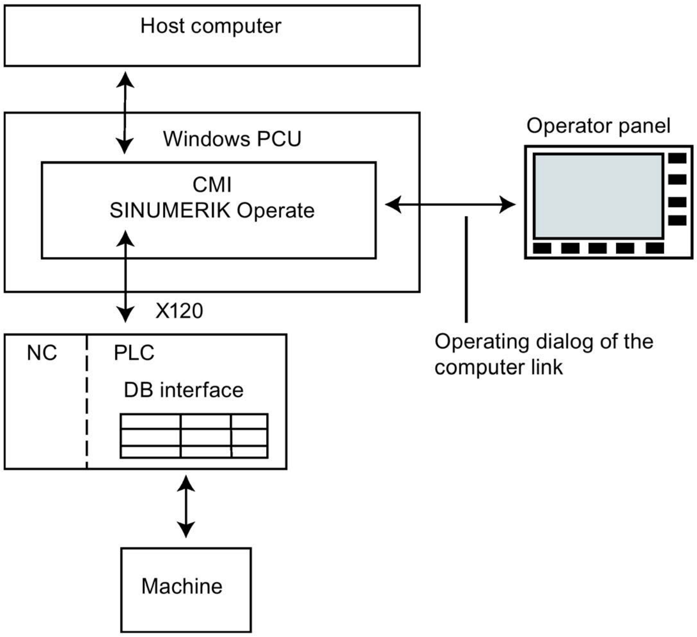
- RPC SINUMERIK: Computer link software on PCU (indirectly from host computer)
- Operator: Dialog program of the computer link
- PLC: User PLC program

To minimize the internal overhead for communication, each change in the interface of RPC SINUMERIK is communicated using "Request from PLC" (part of the interface). RPC SINUMERIK notices the arrival of such requests immediately. (Hotlink).

SINUMERIK HMI-Advanced with RPC



SINUMERIK Operate with CMI



Machine (possible for RPC and CMI)

Global data	
1st docking position	The number of docking positions is defined in the global data
2nd docking position	
...	
nth docking position	

Transport system (only possible for RPC)

Global data	
1st docking position	The number of docking positions is defined in the global data
2nd docking position	
...	
nth Docking positions	
1st transport order	The number of transport orders is defined in the DB12 global data
2nd transport order	
...	
nth transport order	

Languages

RPC SINUMERIK generally supports all languages which are covered by the ASCII character set or for which a language package can be installed on the SINUMERIK.

Languages that use an extended character set (e.g. Chinese, Russian, etc.) are processed by SINUMERIK in the DBCS format. In these languages, the following designations may not be specified in the relevant foreign language.

- NC programs
- Tool data

11.2 DB12 global data

Overview

Table 11- 1 DB12 global data - interface between RPC and PLC

Data element	Short name	Data type	Access from	Offset
Request from PLC	PLCReq	Byte	PLC/ RPC SINUMERIK	0
Change trigger	Trigger	Byte	PLC	1
Request from RPC SINUMERIK	SCReq	Byte	RPC SINUMERIK/ PLC	2
Machine mode	MODE_PLC	Byte	PLC	3
RPC mode	MODE_RPC	Byte	RPC SINUMERIK/ Operator	4
Identifier for tool data	DataTyp	Byte	PLC	5
Magazine number	MagNum	Word	PLC	6-7
Location number	PlaceNum	Word	PLC	8-9
T number	TNum	Word	PLC	10-11
Number of docking positions	DockPosCount	Byte	PLC	12
Number of program assignments	NC4WpcCount	Byte	PLC	13
Machine status	MachineStatus	Byte	PLC	14
NC operating mode	MachineMode	Byte	PLC	15
Reserve 1	Reserve1	Word	PLC	16-17
Reserve 2	Reserve2	Word	PLC	18-19

The individual entries of the data block are explained below:

Request from PLC (byte 0)

Table 11- 2 Request from PLC

Bit No.	Function	Access from
0	Workpiece holder status changed	PLC - 1/ RPC SINUMERIK - 0
1	Report tool	PLC - 1/ RPC SINUMERIK - 0
2	Status change	PLC - 1/ RPC SINUMERIK - 0

The PLC indicates changes in the interface with this byte. Once the PLC has set the request byte, it must set the next bit in the trigger byte (see below).

The PLC may only write in the request byte again when it has been set to 0 by RPC SINUMERIK after processing.

Workpiece holder status changed (byte 0, bit 0)

Workpiece holder status changed is set by the PLC whenever the status of one of the workpiece holders has been changed by the PLC, see also section: DB12 docking position information of the machine (Page 207) or in the case of pallet movements within the machine.

Report tool (byte 0, bit 1)

Report tool is set by the PLC whenever a tool is to be reported to the host computer (e.g. tool breakage). Which tool is to be reported is indicated in the data elements: "Magazine number" and "Location number or T number".

Status change (byte 0, bit 2)

Status change is set by the PLC for each status change (machine mode, RPC mode, machine status, NC mode) about which the host computer is to be informed. RPC SINUMERIK must then send `R_MACHINE_H()` to the host computer, e.g. workpiece holder arrival, NC start, NC end, mode switchovers, etc.

Change trigger (byte 1)

The computer link sets a DDE hotlink to this byte. The PLC sets one bit each in this byte when changes by the PLC are present. For each new trigger, the PLC must set the next bit and reset the previous bit; after bit 7, the process starts over with bit 0.

Request from RPC SINUMERIK (byte 2)

The individual bits are set and cleared by RPC SINUMERIK.

Table 11- 3 Request from RPC SINUMERIK

Bit No.	Function	Access from
0	Synchronization identifier	RPC SINUMERIK
1	Switch off components	RPC SINUMERIK
2	Switch on components	RPC SINUMERIK
3	Request write access to docking position data	RPC SINUMERIK
4	Project-specific special function 1	RPC SINUMERIK
5	Project-specific special function 2	RPC SINUMERIK

Synchronization identifier (byte 2, bit 0)

The synchronization identifier is set and cleared by the host computer, see also section: Overview (Page 141), `C_SYNCH_M()`.

The machine status must remain unchanged for the duration of the synchronization process. The PLC may not start a new machining operation or perform pallet movements.

Switch-off or switch-on components (byte 2, bit 1 and bit 2)

Switch off or switch on components is set by the host computer, see also section: Overview (Page 139) `C_MODE_M()`.

It serves as a request to the PLC components (drives) to switch on or off, see also machine mode, switching components off or on.

Write access to docking position data identifier (byte 2, bit 3)

Request write access to docking position data is set by RPC SINUMERIK as a request when RPC SINUMERIK wants to change the docking position data (workpiece holder status, follow-up machining, machining side). When the PLC has set "Write access to docking position data allowed" in the "Machine mode" data element, only RPC SINUMERIK may make changes.

RPC SINUMERIK must reset the request after performing the change, and the PLC must then also reset the "Write access to docking position data allowed" flag. These coordinated actions prevent RPC SINUMERIK from writing incorrect docking positions due to a pallet movement.

Project-specific special function 1 (byte 2, bit 4)

Project-specific special function 1 is set by the host computer, see also section: Overview (Page 139), `C_MODE_M 5`.

Project-specific special function 1 is reset by the host computer, see also section: Overview (Page 139), `C_MODE_M 6`.

Project-specific special function 2 (byte 2, bit 5)

Project-specific special function 2 is set by the host computer, see also section: Overview (Page 139) `C_MODE_M 7`.

Project-specific special function 2 is reset by the host computer, see also section: Overview (Page 139) `C_MODE_M 8`.

Machine mode (byte 3)

Table 11- 4 Machine mode

Bit No.	Function	Access from
1	Components switched off (after-work hours)	PLC
2	Write access to docking position data allowed	PLC

Components switched off (after work hours) (byte 3, bit 1)

Components switched off is set by the PLC when this status is reached. The request for this is made by bit 1 in the request identifiers.

Write access to docking position data permitted (byte 3, bit 2)

Write access to docking position data allowed is set by the PLC in reaction to the "Request write access to docking position data" request. These coordinated actions prevent RPC SINUMERIK from writing incorrect docking positions due to a pallet movement.

RPC SINUMERIK mode (byte 4)

Table 11- 5 RPC SINUMERIK

Bit No.	Operating mode	Access from
0	Unmanned host computer mode	Operator
1	Manned host computer mode	Operator
2	Manual mode	Operator
3	Special mode	RPC SINUMERIK, Operator
4	Offline host computer 1	RPC SINUMERIK
5	Offline host computer 2	RPC SINUMERIK
6	For `R_MACHINE_H`, is signaled as machine mode 500	PLC
7	For `R_MACHINE_H`, is signaled as machine mode 600	PLC

The RPC SINUMERIK mode can be set by the operator via an RPC SINUMERIK dialog. Special mode can also be selected and deselected from the host computer with `C_MODE_M`. When RPC SINUMERIK detects an interruption in the connection, it sets the bit for "offline". When the offline bit is set, no more data is sent from RPC SINUMERIK to the host computer.

Host computer mode unmanned/manned (byte 4, bit 0 and bit 1)

In the manned and unmanned host computer modes, the NC is started by the PLC, initiated by the host computer. The unmanned and manned host computer modes differ in that, in unmanned production there may be different strategies for handling malfunctions than in manned production.

Manual mode (byte 4, bit 2)

In manual mode, there is no automatic NC start, but material flow on the machine is still automatic.

Special mode (byte 4, bit 3)

In special mode there is no automatic NC start and no automatic material flow on the machine.

Offline host computer 1 / host computer 2 (byte 4, bit 4 and bit 5)

Offline means that there is no connection to the host computer. No data is sent to the host computer. Offline is canceled when RPC SINUMERIK recognizes from the arrival of an RPC from the host computer that the connection is available again.

For `R_MACHINE_H` is signaled as `machine mode 500` (byte 4, bit 6)

For bit 6 = 1, for `R_MACHINE_H` is signaled as `machine mode 500` to the host computer.

For `R_MACHINE_H`, is signaled as `machine mode 600` (byte 4, bit 7)

For bit 7 = 1, for `R_MACHINE_H` is signaled as `machine mode 600` to the host computer.

Identifier for tool data (byte 5)

The tool data identifier can be used to select one of three tool data records for transfer to the host computer. The data areas that are contained in these sets is specified in the configuration program. Identifiers 21, 22, 23, 31, 32, 33 are permitted.

If identifiers 31, 32, 33 are used, then the channel No must be specified in byte 13.

The identifier is transferred with the tool data to the host computer, see also section: Tool data (Page 75).

Magazine number, location number, T number (word 6 - 7, 8 - 9, 10 - 11)

The tool to be reported is specified by magazine number and location number or alternatively by T number. When the T number is specified, magazine number and location are set to 0 and vice versa.

With the request from the PLC "Report tool", RPC SINUMERIK is prompted to read the following data elements and to transfer the tool data to the host computer.

- Magazine number (word 6 - 7)
- Location number (word 8 - 9)
- T number (word 10 - 11)
- Identifier for tool data (byte 5)
- Channel No. (byte 13) for identifier 31, 32, 33

RPC SINUMERIK must then clear these data elements (by filling them with zeros).

Note

Tool reports

Tool reports that occur within the scope of tool loading and unloading are not initiated by the PLC user program.

Number of docking positions (byte 12)

The number of docking positions of the machine (maximum 12) is statically stored when the machine is commissioned. It corresponds to the number of docking position data blocks in the interface, see also section: DB12 docking position information of the machine (Page 207)

Number of program assignments (byte 13)

The number of program assignments is statically stored when the machine is commissioned. It corresponds to the number of NC program assignment blocks in the interface.

Number of program assignments byte 13 is valid **up to** version:

RPC 2.0 SP2 HF3

CMI 2.6 SP2 HF1

Channel No. (byte 13)

The channel assignment for the identifier for tool data 31, 32, 33 is entered here.

Channel No. byte 13 is valid **from** version:

RPC 2.0 SP2 HF4

CMI 2.6 SP2 HF2

Machine status (byte 14)

The machine status is reported to the host computer with `R_MACHINE_H`. Byte 14 must be supplied by the OEM on the PLC side.

Table 11- 6 Machine status

Bit No.	Function	Access from
0	Machine is active	PLC
1	Machine fault	PLC
2	Restart of the machine	PLC

Machine status from DB12.DBB14

Byte 14 must be supplied by the OEM on the PLC side.

When the machine status is read out from DB12.DBB14, MCIS_RPC and Create MyInterface respond as follows:

Table 11- 7 MCIS_RPC on HMI-Advanced and CMI on SINUMERIK Operate (machine version):

DB12.DBX3.1	DB12.DBX14.0	DB12.DBX14.1	DB12.DBX14.2	Status	Meaning
1	X	X	X	4	Components switched off
0	1	0	0	2	Active
0	0	1	0	3	Fault
0	0	0	1	0	Restart
0	0	0	0	1	Inactive

Table 11- 8 MCIS_RPC on HMI-Advanced and CMI on SINUMERIK Operate (TPS variant):

DB12.DBX7.1	DB12.DBX5.0	DB12.DBX5.1	DB12.DBX5.2	Status	Meaning
1	X	X	X	4	Components switched off
0	1	0	0	2	Active
0	0	1	0	3	Fault
0	0	0	1	0	Restart
0	0	0	0	1	Inactive

NC operating mode (byte 15)

Byte 15 must be supplied by the OEM on the PLC side.

The NC operating mode is reported to the host computer with `R_MACHINE_H`. In the host computer you can evaluate.

Table 11- 9 NC operating mode

Bit No.	Function	Access from
0	AUTO	PLC
1	MDA	PLC
2	JOG	PLC
3	TEACH IN	PLC

Reserve 1 (word 16 - 17) and reserve 2 (word 18 -19)

The machine manufacturer can use these variables as required by way of the PLC. The values are reported to the host computer with `R_MACHINE_H()`.

These values can then be evaluated by the host computer and processed further.

11.3 DB12 docking position information of the machine

Introduction

The docking position data describes the respective machine location, machining location, In/Out location.

The number of machine locations is stored in DB12 global data "Number of docking positions" of the global data.

The following docking position data is available:

Table 11- 10 Docking position data

Data element	Short name	Data type	Access from	Offset
Docking position status	DockPos Status	Byte	PLC	20 + (n * 12)
Workpiece holder status	WPCStatus	Byte	PLC/RPC SINUMERIK	21 + (n * 12)
Workpiece holder	WPC	Byte[6]	PLC	22 - 27 + (n * 12)
Machining side	ClampCube Side	Word	PLC/RPC SINUMERIK	28 - 29 + (n * 12)
Follow-up machining	FB	Byte	RPC SINUMERIK	30 + (n * 12)
Reserve	Reserve1	Byte	PLC	31 + (n * 12)

Docking position status (byte 20 + [n * 12])

For every docking position, a block of 12 bytes must be attached in DB12, from byte 20.

n = number of the docking position

1st docking position = 0

2nd docking position = 1 .... etc.

A maximum of 12 docking positions can be used.

The following states are available:

- Bit 0 = Fault
- Bit 1 = Disabled

The bit field describes the current status of the docking position. It is set by the PLC. If no bit is set, the docking position is enabled.

The "Fault" bit is set or cleared on the basis of I/O signals. The cause of the fault is communicated to the host computer using the "Reports" function, see also: Message from SINUMERIK to the host computer (Page 94), `R_REPORT_H`.

The PLC does not execute any pallet transports between docking positions with "Fault" status.

If the docking position has the "Disabled" status, it may not be approached by the transport system.

Workpiece holder status (byte 21 + [n * 12])

Table 11- 11 Workpiece holder states

Bit No.	Function	Status	Access from
0	Newly delivered (without program assignment)	1	PLC
1	Machining is planned (with program assignment)	2	RPC SINUMERIK
2	Prepare program selection	4	PLC
3	Program selection complete	8	RPC SINUMERIK
4	In progress	16	PLC
5	Machining finished	32	PLC
6	Machining aborted	64	PLC
7	Machining is not planned (only for buffering)	128	PLC

Newly delivered (byte 21, bit 0)

The "Newly delivered" status (status 1) is assigned by the PLC for a newly delivered workpiece holder.

Exception: Machining is not planned.

This status causes the host computer to perform the program assignment. When the program has been assigned, the "Machining is planned" status (status 2) is set by the host computer.

Machining is planned (byte 21, bit 1)

For a workpiece holder with the "Machining is planned" status (status 2), the PLC sets the "Prepare program selection" status as soon as the active machining has finished (status 4).

Prepare program selection (byte 21, bit 2)

The precondition for "Prepare program selection" (status 4) is that the program assigned for this pallet by the host computer has been loaded into the NC and selected for machining.

Program selection complete (byte 21, bit 3)

When the program has been selected, the host computer sets the "Program selection complete" status (status 8) for the associated workpiece holder.

The PLC can now initiate the NC start. However, the machine manufacturer is responsible for ensuring the PLC will only start the NC when all safety-related requirements are met, e.g. protective door closed, etc.

Machining started (in progress) (byte 21, bit 4)

Once machining has been started, the PLC sets the "In progress" status (status 16). When machining has finished, the PLC then sets the "Machining finished" status (status 32) for the appropriate workpiece holder. Workpiece holders with the "Machining finished" status (status 32) are transported independently to an unloading location by the PLC.

Machining finished (byte 21, bit 5)

When the "Follow-up machining" identifier is set, the workpiece holder remains at the machining location. As response to the "Machining finished" status (status 32), the host computer sets the "Machining is planned" status (status 2) again. As subsequent response, the PLC uses the "Prepare program selection" status (status 4) to prompt the host computer to select the program for the next machining steps. The remaining sequence of operations is the same as for initial machining.

Machining aborted (byte 21, bit 6)

The "Machining aborted" status (status 64) is set when a workpiece holder will no longer be machined after a malfunction. This identifier might occur, in particular, in unmanned production. A workpiece holder with this identifier may not be brought to another machine for a further operation. It may only be brought to a storage location.

Machining is not planned (byte 21, bit 7)

A workpiece holder that was delivered for buffering by the TPS is assigned the "Machining is not planned" status (status 128) and not the "Newly delivered" status (status 1). The TPS transfers this information to the PLC. For workpiece holders with this status, there is no program assignment by the host computer.

Workpiece holder (byte 22 - 27)

Identifier of the workpiece holder that is currently located at the docking position, e.g. "WST01". This information is entered by the PLC. This requires that this information can be transferred from the transport system or directly from the workpiece holder. If no workpiece holder is in the docking position, the field must be filled with binary 0. The identifier must be terminated with '\0' after the last character because RPC SINUMERIK expects a string. This means only identifiers up to 5 bytes long are permissible.

Machining side (byte 28 - 29)

The host computer sets this information simultaneously with the "Program selection complete" machining status (status 8). The PLC uses this value to set a corresponding side of a clamp cube for machining, or passes this value on to the NC.

The computer link specifies the order in which the sides of clamp cubes will be machined based on the program assignment. If the PLC is to be able to influence the machining order, an entry can be optionally made in the configuration file of the host computer that causes the program assignment data to be mirrored in a separate data block of the PLC. The PLC then has read access to this data.

The host computer is notified of the machining side selected by the PLC in the "Machining side" field. This takes place simultaneously with the setting of the "Prepare program selection" machining status (status 4). The host computer selects the program for the side specified by the PLC. The rest of the sequence remains unchanged.

Follow-up machining (byte 30)

The host computer sets this identifier simultaneously with the "Program selection complete" machining status (status 8). It informs the PLC as to whether - in addition to the current machining - follow-on machining is planned. The PLC uses this information to control the workpiece holder transport within the machine.

Reserve (byte 31)

Free for the OEM

Workpiece holder and production dialog 12

12.1 Arrival of workpiece holder

Communication takes place using the above-described interface DBs.

Additional details are described below.

Overview

When a workpiece holder arrives at a docking position of the machine, either the PLC must read the workpiece holder designation or the transport system must transfer it to the machine. In addition, the transport system must notify the machine whether the workpiece holder is being delivered for machining or only for buffering, if this information is not reported by the host computer to the machine `[NC4WPC_M]`. The way in which this information is transferred between the TPS and machine is not a part of this description.

The PLC must write the docking position status and the workpiece holder number and status to the corresponding interface DB and then set the status change request identifier in the interface for states and increment the change trigger. RPC SINUMERIK then sends `R_MACHINE_H()` to the host computer.

Note

Transport system only possible with RPC

The functionality with the transport system (TPS) is only available for RPC, but not for CMI.

12.2 Production dialog between PLC/NC and RPC SINUMERIK

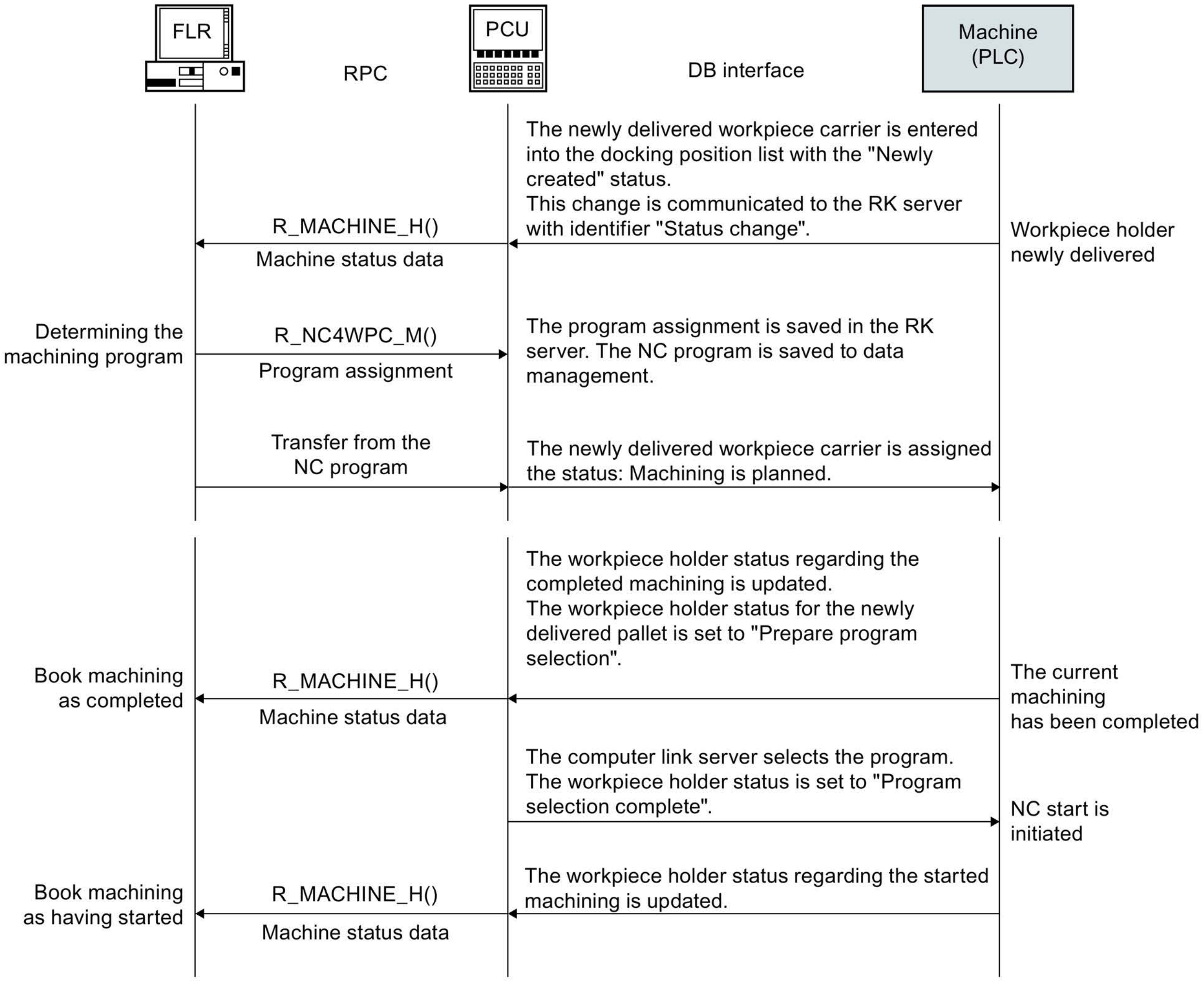
Introduction

When the host computer has detected based on `R_MACHINE_H()` that a workpiece holder to be machined has arrived at the machine, it sends one or more NC program assignments `NC4WPC_M()`.

RPC SINUMERIK enters this information in an internal list and optionally in the interface for NC program assignment.

Determining data for R_MACHINE_H

The interfaces for states and docking position data must be read as well as the SINUMERIK operating mode (AUTO, MDA, JOG) and the program status. The operating mode reported to the host computer consists of the computer link operating mode and the SINUMERIK operating mode.


FLR
RPC
PCU
DB interface
Machine (PLC)
The newly delivered workpiece carrier is entered into the docking position list with the "Newly created" status.
This change is communicated to the RK server with identifier "Status change".
R_MACHINE_H()
Machine status data
Workpiece holder newly delivered
Determining the machining program
R_NC4WPC_M()
Program assignment
The program assignment is saved in the RK server. The NC program is saved to data management.
Transfer from the NC program
The newly delivered workpiece carrier is assigned the status: Machining is planned.
The workpiece holder status regarding the completed machining is updated.
The workpiece holder status for the newly delivered pallet is set to "Prepare program selection".
Book machining as completed
R_MACHINE_H()
Machine status data
The current machining has been completed
The computer link server selects the program.
The workpiece holder status is set to "Program selection complete".
NC start is initiated
Book machining as having started
R_MACHINE_H()
Machine status data
The workpiece holder status regarding the started machining is updated.

Loading/unloading tools via the host computer 13

13.1 Overview

Functions

- "Load" function

  A tool can be loaded into a machine via the host computer initiated by the Tool Ident Connection with command 4. To do this, the tool ID and the Duplo No. or the int. tool No. must be known. The tool ID is transferred to the host computer in DB19 from the Tool Ident Connection with command code 4 SFkt 26 using RPC or CMI.
  The host computer prepares the tool data and transfers this to the SINUMERIK.

  For a description of the Tool Ident Connection, see the Function Manual, Tool Management: Chapter 6.

- "Unload tool" function
  Using command 5, the "Tool Ident Connection" can initiate that a tool is unloaded from the machine to the host computer. Tool data is transferred from the Tool Ident Connection using command code 5 SFkt 27 using RPC or CMI to the host computer.
  The host computer accepts the tool data.

  For a description of the Tool Ident Connection, see the Function Manual, Tool Management: Chapter 6.

Functions are available for:

- From Create MyInterface 02.06.01.00 and higher, these functions are available for PCU 50 with SW 4.4 SP3 (Create MyInterface and Tool Ident Connection installed on PCU 50).
- From Create MyInterface 02.06.01.01 and higher, these functions are available for PCU 50 with SW 2.7 SP3 HF3 (Create MyInterface and Tool Ident Connection installed on PCU 50).
- From Create MyInterface 02.06.01.02 and higher, these functions are available for PCU 50 - also for SW 4.5 SP2 (Create MyInterface and Tool Ident Connection installed on PCU 50).
- From Create MyInterface 02.06.02.00 and higher, these functions are available for PCU 50 or industrial PC and also for SW from 4.5 SP2 or higher (Create MyInterface and Tool Ident Connection installed on PCU 50 or industrial PC).
- For installation on an IPC, SW 2.6 / 2.7 / 4.4 / 4.5 / 4.7 / 4.8 of the NCU are supported. Note that in these cases, Tool Ident Connection must be configured on the IPC and no longer on the NCU. If necessary, you must transfer/adapt an existing configuration and deactivate Tool Ident Connection on the NCU. If necessary, contact your Siemens sales partner or the Siemens hotline.

13.2 System requirements for Tool Ident Connection

This Chapter provides information on the requirements for using the "Load/unload via host computer" functions on a PCU 50, industrial PC or host computer.

Requirement

Installation of a licensed and executable Tool Ident Connection.

13.3 Installing CMI on a PCU 50.x, IPC, or host computer

Restriction

- Use of the two functions "Load/unload via host computer" is only permitted in a 1:1 relationship.
- Simultaneous use of Tool Ident Connection on an NCU and PCU 50.x, IPC, or host computer is not permissible.
- Deactivate Tool Ident Connection on the NCU after commissioning, see Function Manual "Tool Management, Chapter 6".
  Then activate Tool Ident Connection on the PCU 50.x, IPC, or host computer.

13.4 Activate Tool Ident Connection

Configuration files

The following files are available to configure Tool Ident Connection:

Standard configuration files	Directory: CompactFlash card	Directory: PCU 50 WinXP	Directory: PCU 50 Win7
tdiidentcfg.xml	/card/siemens/sinumerik/hmi/template\toolidentconnection	F:\hmisl\siemens\sinumerik\hmi\template\toolidentconnection	C:\ProgramFiles(x86)\Siemens\MotionControl\siemens\sinumerik\hmi\template\toolidentconnection
toolSpec.xml			
wkonvert.mcx			
DefToolDat.txt			
wkonvert.mcc			
systemconfiguration.ini			
oem_sltmlistconfig.xml			

Setup of the Tool Ident Connection option

The following entry must be made in the "systemconfiguration.ini", in the section [services] to set up the software:

```
SVCxxx=name:=SlMcisTdiIdentConnectionService,

implementation:=slmcistdiidentconnectionservice.SlMcisTdiIdentConnectionService,

process:=SlHmiHost1, timeout:=5000, shutdowntime:=5000
```

Note

The `xxx` in `SVCxxx` must be replaced by the free number of a service. Only numbers as of 200 can be used.

To adapt the Tool Ident Connection configuration function, the systemconfiguration.ini and, if adapted, the configuration files (from the directory mentioned above) must be copied to the following directory:

- CompactFlash card: /card/addon/sinumerik/hmi/cfg
- PCU 50 (WinXP): F:\hmisl\addon\sinumerik\hmi\cfg
- PCU 50 (Win7): C:/Programs(x86)\Siemens\MotionControl\/addon\sinumerik\hmi\cfg

The oem_sltmlistconfig.xml file must be renamed, depending on the technology. This is explained in the Commissioning Manual IM9, Section "Tool management", Section "Configuring the user interface".

A detailed description of the configuration options for the oem_sltmlistconfig.xml can be found in the Commissioning Manual IM9, Section "Tool management", Section "Configuring the code carrier connection".

Files "wkonvert.txt, DefToolDat.txt" and "wkonvert.mcc" are configuration files. They must be compiled after every change with the "wkonvert wizard". The result of the compilation is a "wkoonvert.mcx". This must be copied to the appropriate target directory for the settings to take effect.

13.5 Configuring Tool Ident Connection

In order to use functions "Load/unload on/to host computer", you must commission Tool Ident Connection.

The Tool Ident Connection product must be separately purchased and commissioned. Please contact your local Siemens sales organization.

Note

Tool Ident Connection interface - CMI

When the interface between Tool Ident Connection and Create MyInterface is used, special restrictions regarding the number of machines and number of hosts must be observed.

K code 4 = command code 4; load tool from the host computer.

K code 5 = command code 5; load tool to the host computer.

Example 1

Tool Ident Connection and Create MyInterface are installed on one PCU.

- Configure just one host for the function, with K code 4 and K code 5 for this machine.
- For this machine, it is not permissible that you configure the remaining 7 hosts with Tool Ident Connection.
- However, all 8 hosts can use the complete functional scope of Create MyInterface.

Example 2

Tool Ident Connection and Create MyInterface are installed on one IPC.

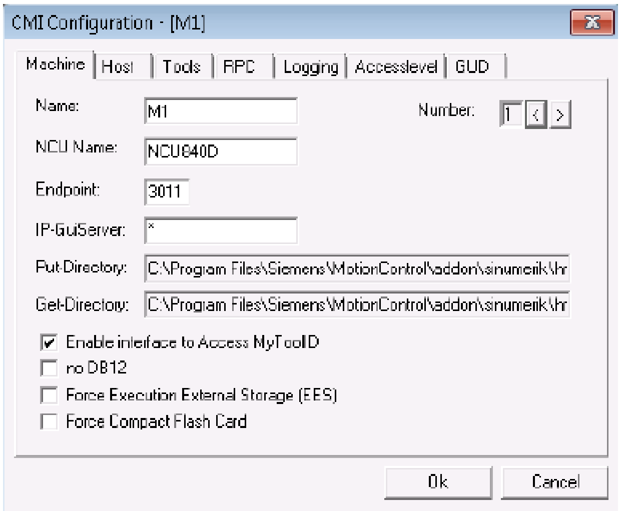
- Only one of the 4 possible machines may use the interface between Tool Ident Connection and Create MyInterface.
- It is not permissible that the remaining 3 of the 4 machines use the interface between Tool Ident Connection and Create MyInterface.
- If, for one machine, you activate Tool Ident Connection, then the option box for the other machines is deactivated.
- Configure just one host for the function, with K code 4 and K code 5, for the corresponding machine.
- It is not permissible that you use Tool Ident Connection to configure the remaining 7 hosts for this machine.
- However, all 8 hosts can use the complete functional scope of Create MyInterface.

13.6 Configuring CMI with Create MyInterface Configuration

To use the "Load/unload via host computer" functions, you must select the machine that establishes the connection to Tool Ident Connection in the configuration tool of Create MyInterface.

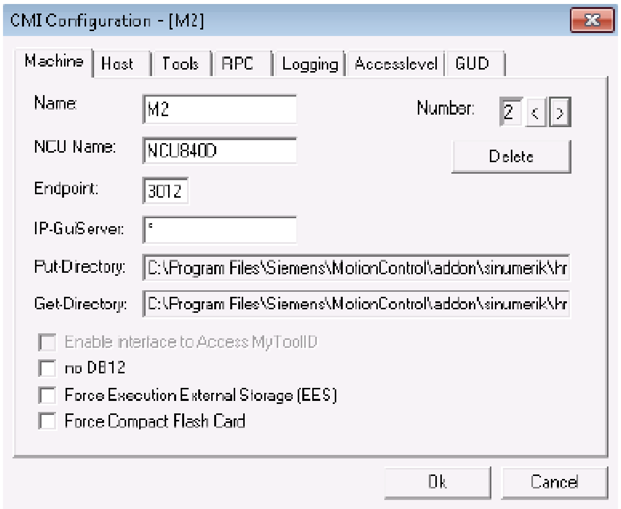
Procedure

1. Start the configuration program "Create MyInterface Configuration".
2. Select the machine that establishes the connection to Tool Ident Connection.
3. Open the "Machine" tab.
   - Select the "Enable interface to Access MyToolID" check box.
   - Click "OK" to save the selection.


CMI Configuration - [M1]

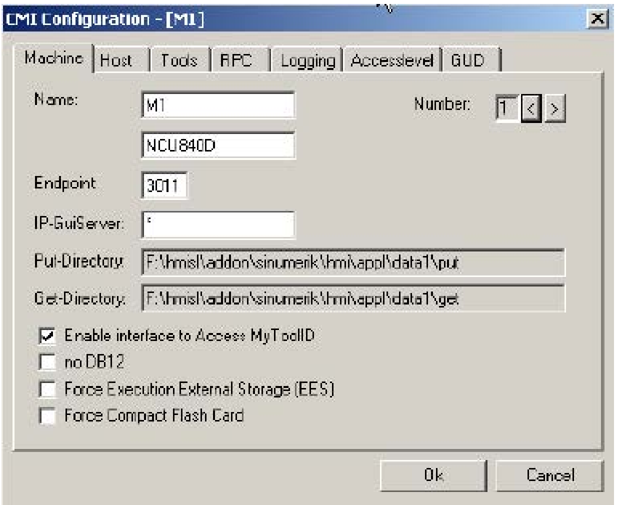
Name: M1
Number: 1
NCU Name: NCU840D
Endpoint: 3011
IP-GuiServer: *
Put-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr
Get-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr
☑ Enable interface to Access MyToolID
☐ no DB12
☐ Force Execution External Storage (EES)
☐ Force Compact Flash Card

After activation, the selection cannot be made for any other machines, e.g. [M2].


CMI Configuration - [M2]

Name: M2
Number: 2
NCU Name: NCU840D
Endpoint: 3012
IP-GuiServer: *
Put-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr
Get-Directory: C:\Program Files\Siemens\MotionControl\addon\sinumerik\hr
☐ Enable interface to Access MyToolID
☐ no DB12
☐ Force Execution External Storage (EES)
☐ Force Compact Flash Card

On the PCU, the check box can only be selected for the current machine. No more machines can be configured.



Name: M1

Number: 1

NCU840D

Endpoint: 3011

IP-GuiServer: *

Put-Directory: F:\hmisl\addon\sinumerik\hmi\appl\data1\put

Get-Directory: F:\hmisl\addon\sinumerik\hmi\appl\data1\get

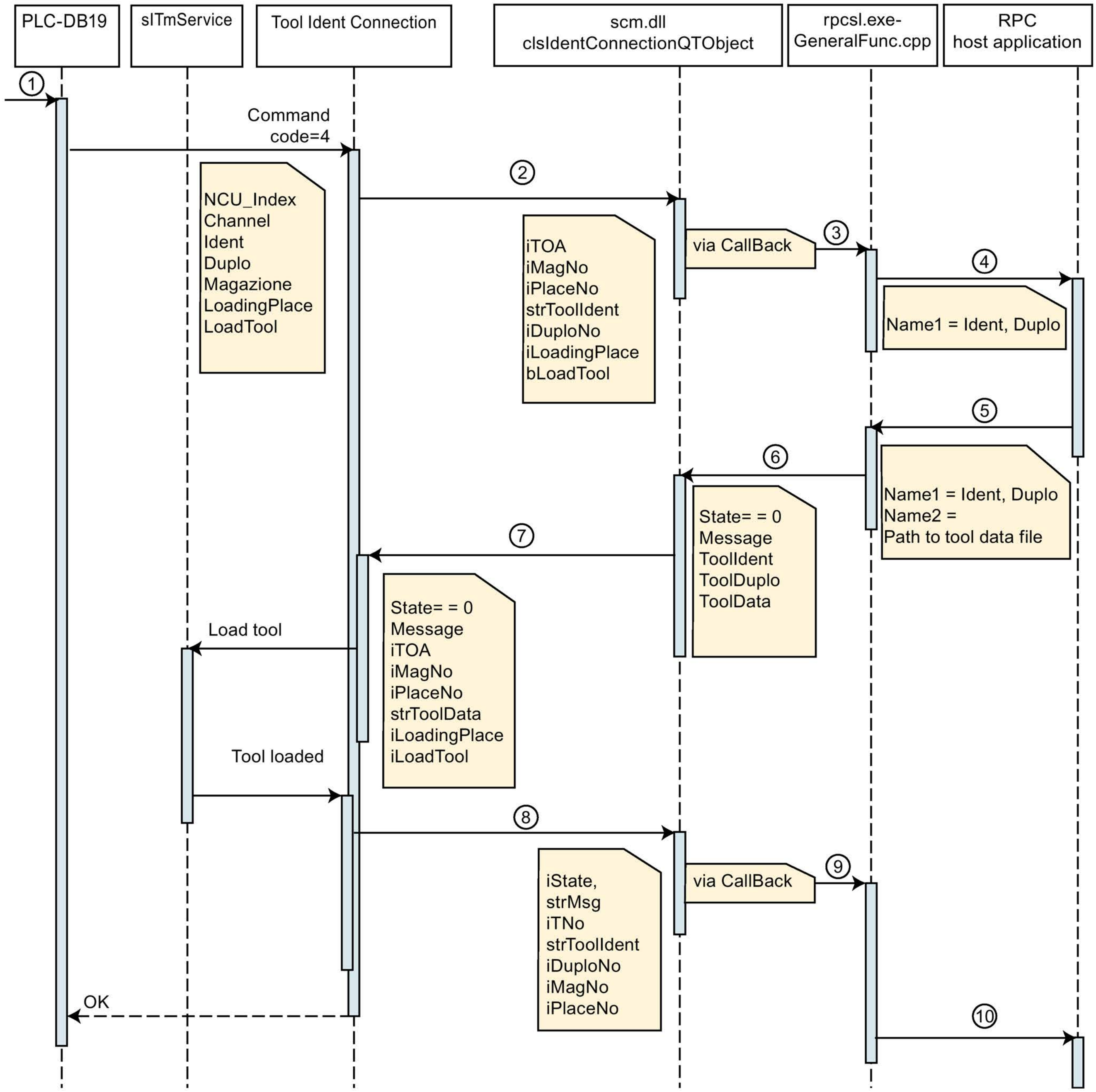
☑ Enable interface to Access MyToolID

□ no DB12

□ Force Execution External Storage (EES)

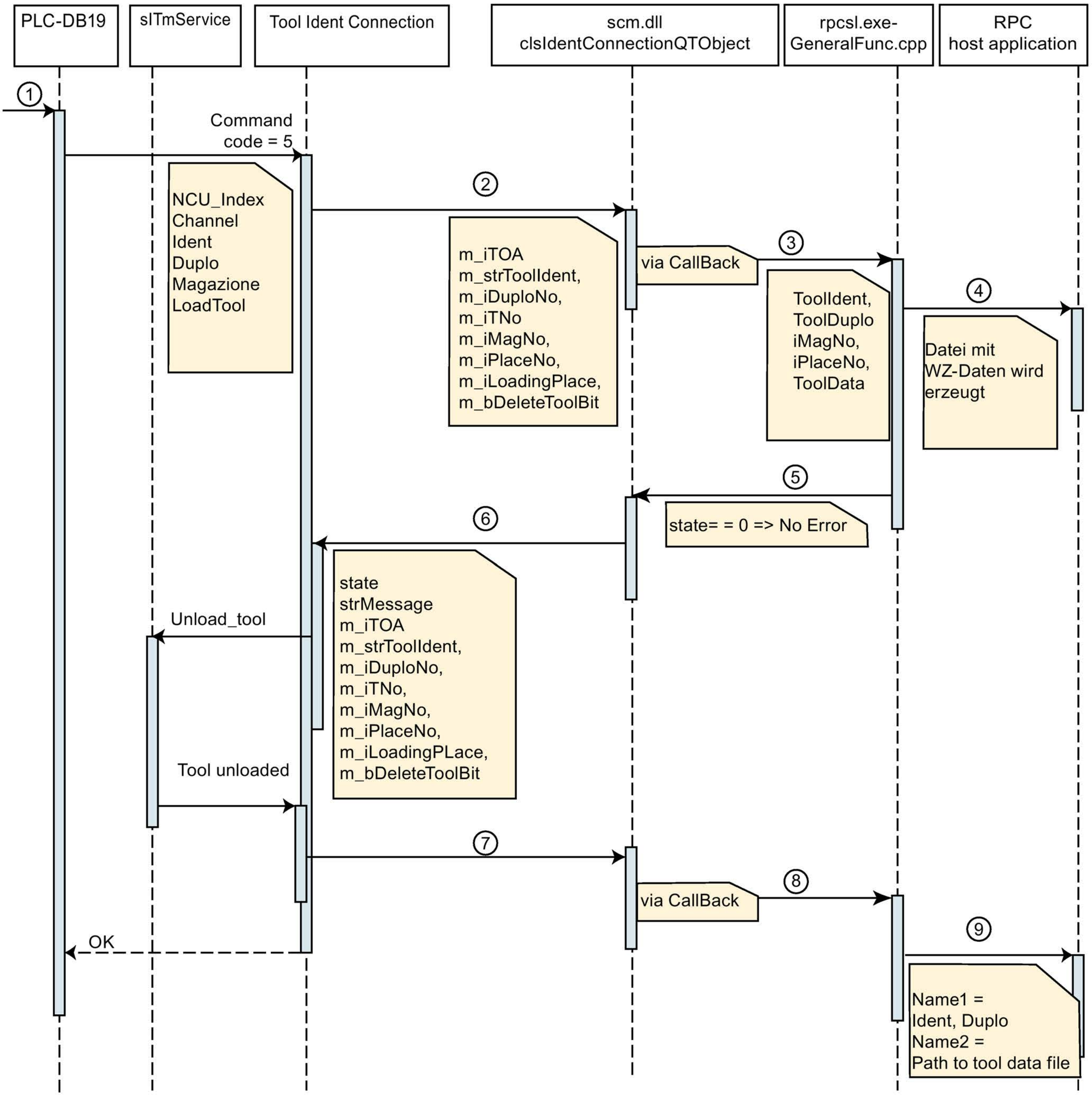
□ Force Compact Flash Card

13.7 Load function – command code 4



① Read Balluf chip
② signalNotifyRequestLoadToolFromServer(...)
③ DoRequestLoadToolFromServer(ToolIdent, ToolDuplo)
④ T_DATA_H(SFkt=26)
⑤ R_DATA_M(SFkt=26)
⑥ ResponseLoadToolFromServer(...)
⑦ slotServerRequestLoadToolFromServer(...)
⑧ signalNotifyResultLoadToolFromServer(...)
⑨ DoResponseLoadToolFromServer(state, message)
⑩ R_REPORT_H(SFkt=26 Type=5)

13.8 Unload function – command code 5



① Balluf chip ...
② signalNotifyRequestUnloadToolToServer(...)
③ DoRequestUnloadToolToServer()
④ CreateFile()
⑤ ResponseUnloadToolToServer(state, message)
⑥ slotServerResponseUnloadToolToServer(lState, strMessa...)
⑦ signalNotifyResultUnloadToolToServer(State, Message)
⑧ DoResultUnloadToolToServer(State, Message)
⑨ R_DATA_H(SFkt=27)

Online/offline detection of host computer 14

14.1 System requirements

CMI internally remembers the connection status of the host computer, i.e. whether it is "online" or "offline".

Versions

The online/offline detection of the host computer in CMI has been revised and is available for CMI 2.6.2.1 and higher as follows:

- Start RPC/CMI (Page 222)
- RPC/CMI operation (Page 223)
- Detect status via network interface (Page 223)
- Detect status via PLC interface (Page 224)

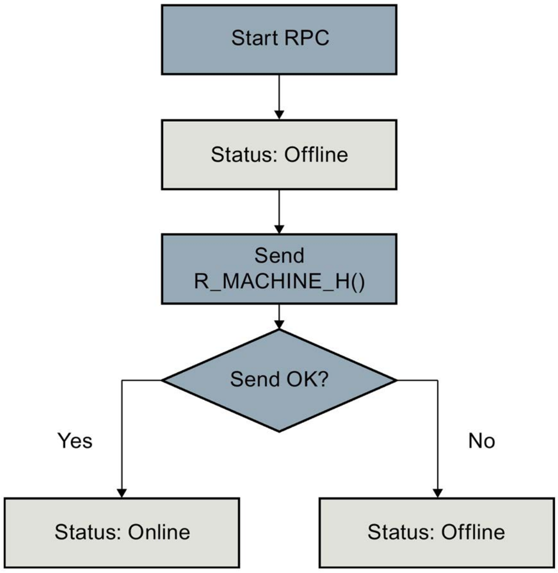
14.2 Start RPC/CMI

Function

Initial state: Host computer is offline.

After the start of RPC/CMI, the current machine status is automatically sent to the host computer with the RPC call `R_MACHINE_H`.

- If RPC/CMI was able to send the RPC call to the host computer, RPC notes the status of the host computer internally as "online".
- If RPC/CMI was not able to send the RPC call to the host computer, the internal status remains "offline".



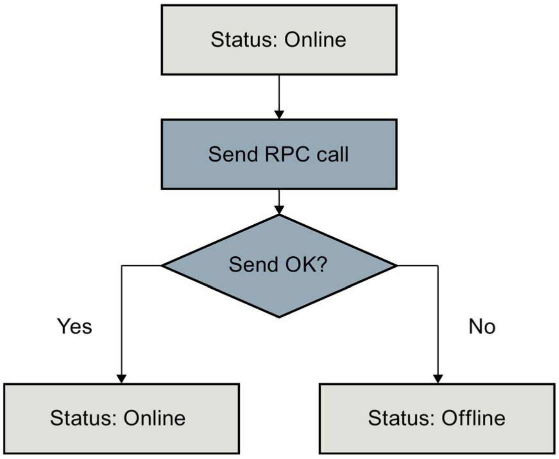
14.3 RPC/CMI operation

Function

Initial state: Host computer is online.

During operation of RPC/CMI, the current machine status is reestablished after each RPC call sent to the host computer.

- If RPC/CMI was able to send the RPC call to the host computer, the internal status remains "online".
- If RPC/CMI was not able to send the RPC call to the host computer, RPC/CMI notes the status of the host computer internally as "offline".

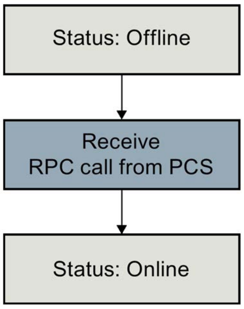


14.4 Detect status via network interface

Function

Initial state: Host computer is offline.

With the receipt of an RPC call sent from the host computer, the status changes from "offline" to "online".

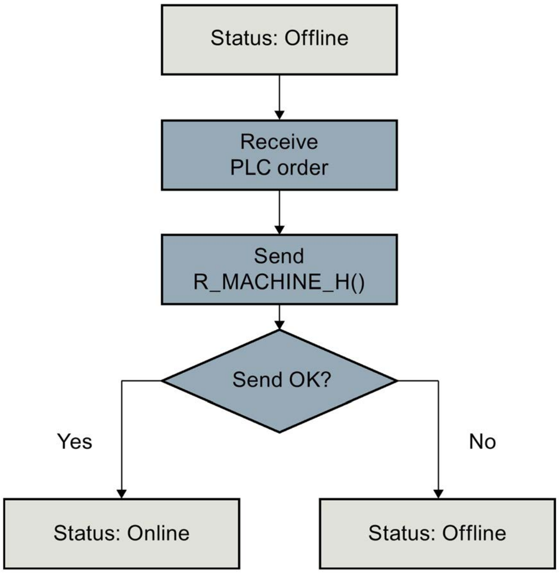


14.5 Detect status via PLC interface

Function

Initial state: Host computer is offline and RPC/CMI receives a PLC order.

- If RPC/CMI was able to send the RPC call to the host computer, the host computer status is set internally to "online".
- If RPC/CMI was not able to send the RPC call to the host computer, the internal status remains "offline".



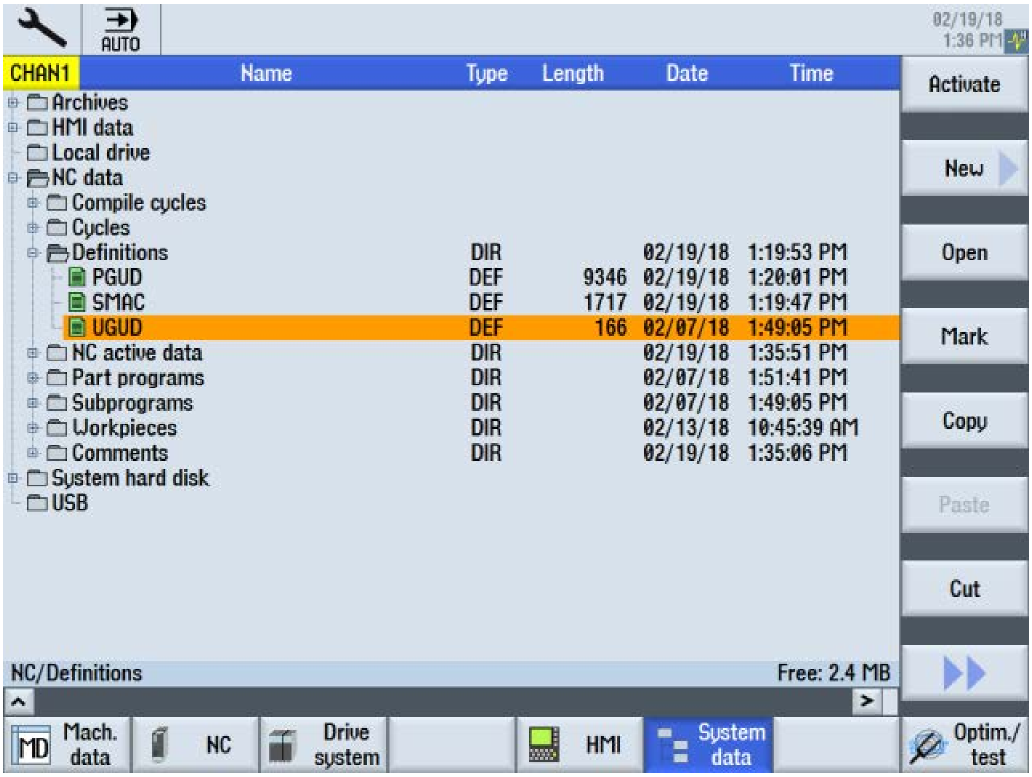
Explanation of GUD variables 15

15.1 Definitions in the NC

The use of GUD variables is described below using an example.

Procedure

1. Press the "System data" softkey on the controller.
2. Open the "Definitions" directory and select, e.g. the GUD variable "MGUD".



CHAN1 Name	Type	Length	Date	Time
Archives				
HMI data				
Local drive				
NC data				
Compile cycles				
Cycles				
Definitions	DIR		02/19/18	1:19:53 PM
PGUD	DEF	9346	02/19/18	1:20:01 PM
SMAC	DEF	1717	02/19/18	1:19:47 PM
UGUD	DEF	166	02/07/18	1:49:05 PM
NC active data	DIR		02/19/18	1:35:51 PM
Part programs	DIR		02/07/18	1:51:41 PM
Subprograms	DIR		02/07/18	1:49:05 PM
Workpieces	DIR		02/13/18	10:45:39 AM
Comments	DIR		02/19/18	1:35:06 PM
System hard disk				
USB				

NC/Definitions

Free: 2.4 MB

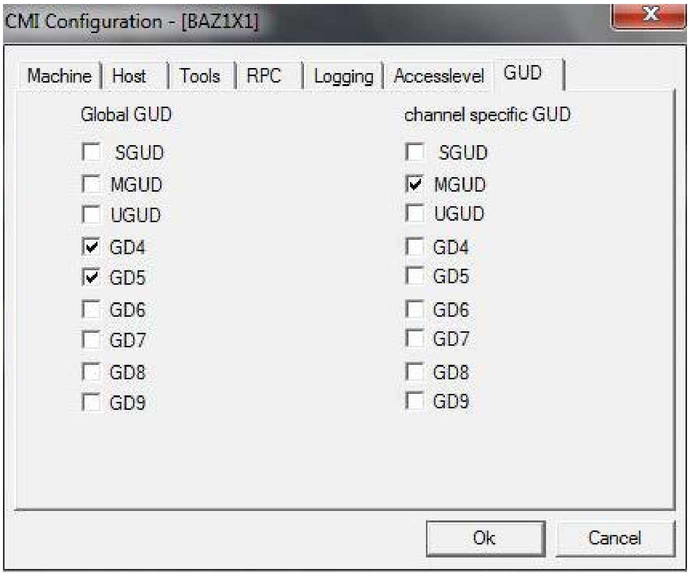
15.2 Enabling GUD variables in CMI

15.2.1 Introduction

The use of GUD variables is described below using an example.

Procedure

1. Start the configuration program and open the "GUD" tab.
2. Select the "MGUD" checkbox and click "OK".



Global GUD	channel specific GUD
□ SGUD	□ SGUD
□ MGUD	☑ MGUD
□ UGUD	□ UGUD
☑ GD4	□ GD4
☑ GD5	□ GD5
□ GD6	□ GD6
□ GD7	□ GD7
□ GD8	□ GD8
□ GD9	□ GD9

15.2.2 Channel-specific GUD variables

Definition

The use of channel-specific GUD variables is described below using an example.

The definition is:

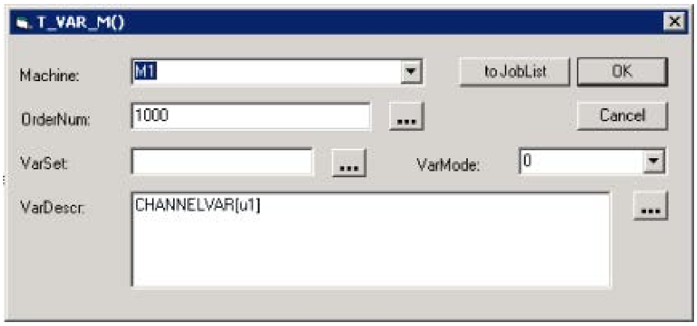
```
DEF CHAN INT CHANNEL VARIABLE
```

```
M30
```

15.2.3 Reading a channel-specific GUD variable

Sequence

VarDescr: <Variable name[u<1,2 …>] -> 1,2 … Channel number



T_VAR_M()

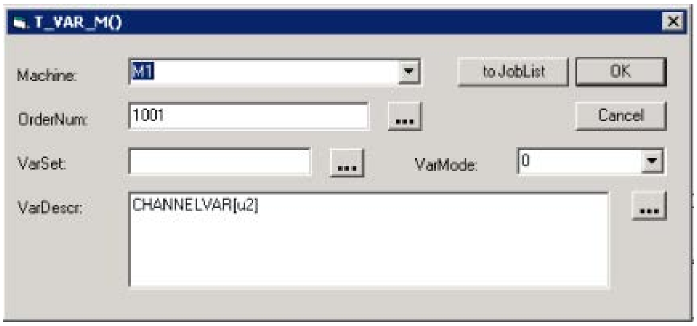
Machine: M1

OrderNum: 1000

VarSet:

VarMode: 0

VarDescr: CHANNELVAR[u1]



T_VAR_M()

Machine: M1

OrderNum: 1001

VarSet:

VarMode: 0

VarDescr: CHANNELVAR[u2]

The channel-specific GUD variable is read and the following "VarDescr:" is output:

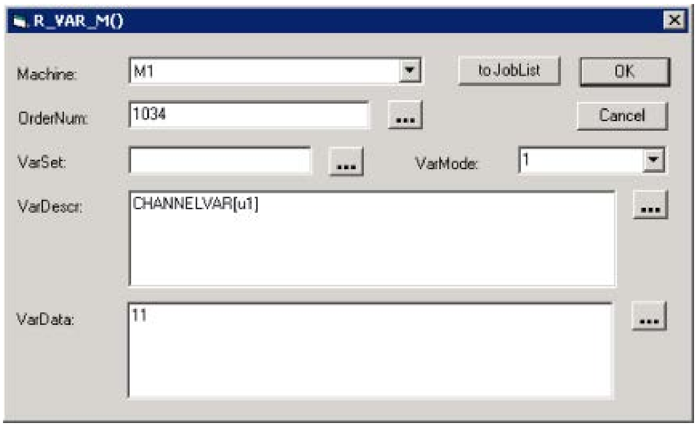
Read channel-specific GUD variable

```
07.08.2017 11:50:07    T_VAR_M    Ret=0
OrderNum->1000 Machine->"M1" Host->"H1" VarMode->"0" VarSet->"" VarDescr->"CHANNELVAR[u1]"
Runtime->0.0049
07.08.2017 11:50:07    R_VAR_H
OrderNum->1000 Machine->"M1" Host->"H1" VarMode->0 VarSet->"Tmp" VarDescr->""
VarData->"2"
Runtime->0.0118
07.08.2017 11:53:52    T_VAR_M     Ret=0
OrderNum->1001 Machine->"M1" Host->"H1" VarMode->"0" VarSet->"" VarDescr->"CHANNELVAR[u2]"
Runtime->0.0045
07.08.2017 11:53:52    R_VAR_H
OrderNum->1037 Machine->"M1" Host->"H1" VarMode->0 VarSet->"Tmp" VarDescr->""
VarData->"0"
Runtime->0.0117
```

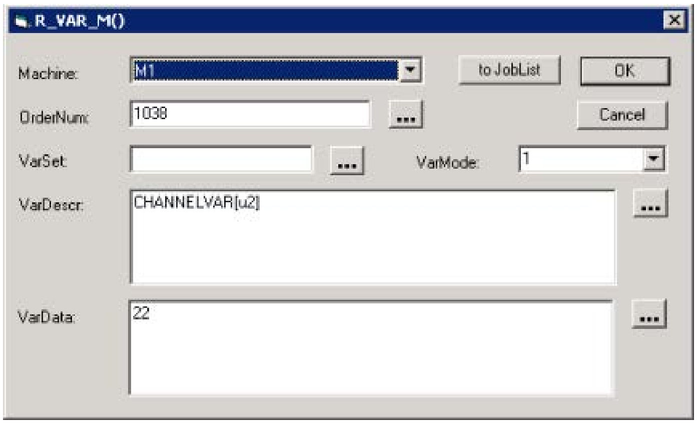
15.2.4 Writing channel-specific GUD variables

Sequence

VarDescr: <Variable name[u<1,2 …>] -> 1,2 … Channel number



Machine: M1
OrderNum: 1034
VarSet:
VarMode: 1
VarDescr: CHANNELVAR[u1]
VarData: 11



Machine: M1
OrderNum: 1038
VarSet:
VarMode: 1
VarDescr: CHANNELVAR[u2]
VarData: 22

The channel-specific GUD variable is written and the following "VarDescr:" is output:

Write channel-specific GUD variable

```
07.08.2017 11:58:59    R_VAR_M     Ret=0
OrderNum->1038 Machine->"M1" Host->"H1" VarMode->1 VarSet->"" VarDescr->"CHANNELVAR[u1]"
VarData->"11"
Runtime->0.0048
07.08.2017 11:59:00    R_REPORT_H
OrderNum->1038 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 ResInt1->0
ResInt2->0 ResByte->""
07.08.2017 12:02:40    R_VAR_M    Ret=0
OrderNum->1039 Machine->"M1" Host->"H1" VarMode->1 VarSet->"" VarDescr->"CHANNELVAR[u2]"
VarData->"22" Runtime->0.0066
07.08.2017 12:02:40 R_REPORT_H
OrderNum->1039 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 0
ResInt1->0 ResInt2->0 ResByte->""
```

15.2.5 Global GUD variables

Definition

The use of global GUD variables is described below using an example.

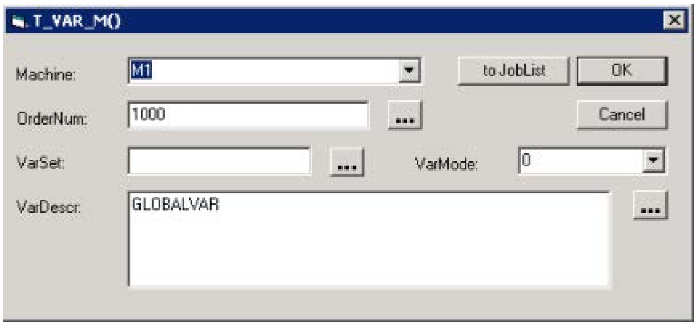
The definition in "MGUD.DEF" is:

```
DEF NCK REAL GLOBALVAR
```

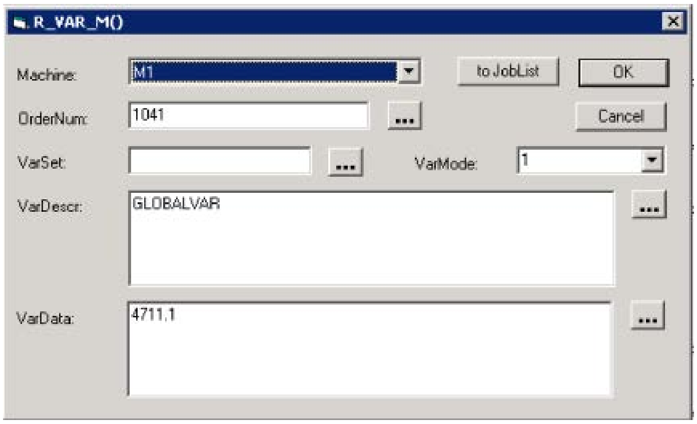
```
M30
```

15.2.6 Reading global GUD variables

Sequence



Machine: M1
OrderNum: 1000
VarSet:
VarMode: 0
VarDescr: GLOBALVAR



Machine: M1
OrderNum: 1041
VarSet:
VarMode: 1
VarDescr: GLOBALVAR
VarData: 4711.1

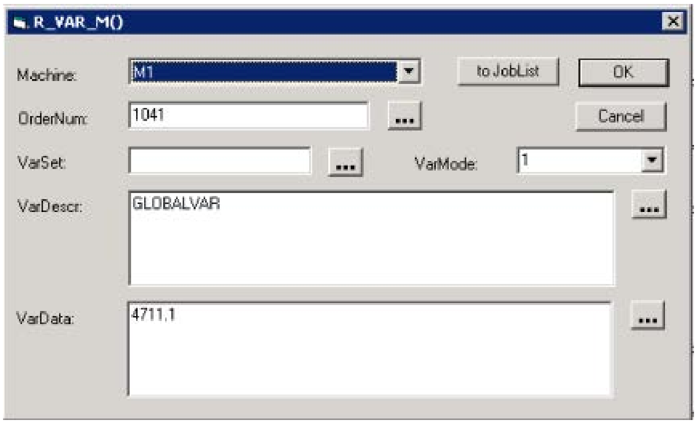
The global GUD variable is read and the following "VarDescr:" is output:

Read a global GUD variable

```
07.08.2017 12:07:27    T_VAR_M     Ret=0
OrderNum->1042 Machine->"M1" Host->"H1" VarMode->"0" VarSet->"" VarDescr->"GLOBALVAR"
Runtime->0.0027
07.08.2017 12:07:27    R_VAR_H
OrderNum->1042 Machine->"M1" Host->"H1" VarMode->0 VarSet->"Tmp" VarDescr->"" VarData-
>"4711.1"
Runtime->0.015
```

15.2.7 Writing global GUD variables

Sequence



Machine: M1
OrderNum: 1041
VarSet:
VarMode: 1
VarDescr: GLOBALVAR
VarData: 4711.1

The global GUD variable is written to and the following "VarDescr:" is output:

Write a global GUD variable

```
07.08.2017 12:07:16    R_VAR_M    Ret=0
OrderNum->1041 Machine->"M1" Host->"H1" VarMode->1 VarSet->""
VarDescr->"GLOBALVAR" VarData->"4711.1" Runtime->0.0027
07.08.2017 12:07:16    R_REPORT_H
OrderNum->1041 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 0
ResInt1->0 ResInt2->0 ResByte->""
```

15.2.8 Global ARRAY GUD variables

Definition

The use of global GUD variables is described below using an example.

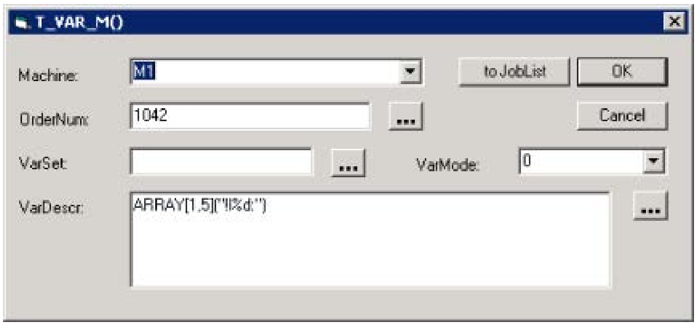
The definition in "UGUD.DEF" is:

```
DEF NCK REAL ARRAY[5] = SET(1,2,3,4,5)

M30
```

15.2.9 Reading global ARRAY GUD variables

Sequence

Format ("!l%d:") for reading of multiple variables: -> values as integer with separator <:>



T_VAR_M()

Machine: M1

OrderNum: 1042

VarSet:

VarMode: 0

VarDescr: ARRAY[1,5]("!l%d:")

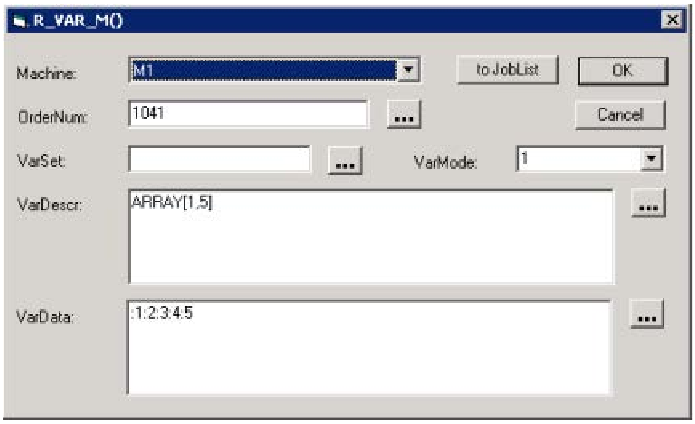
Read a global ARRAY GUD variable

```
07.08.2017 12:18:37    T_VAR_M    Ret=0
OrderNum->1043 Machine->"M1" Host->"H1" VarMode->"0" VarSet->"" VarDescr-
>"ARRAY[1,5]("!l%d:")"
Runtime->0.0046
07.08.2017 12:18:37    R_VAR_H
OrderNum->1043 Machine->"M1" Host->"H1" VarMode->0 VarSet->"Tmp" VarDescr->""
VarData->"1:2:3:4:5:" Runtime->0.0179
```

15.2.10 Writing global ARRAY GUD variables

Sequence

Format (":1:2:3:4:5") for writing to several variables: The first character is separator <:>



Machine: M1

OrderNum: 1041

VarSet:

VarMode: 1

VarDescr: ARRAY[1,5]

VarData: :1:2:3:4:5

Write a global ARRAY GUD variable

```
07.08.2017 13:16:37    R_VAR_M     Ret=0
OrderNum->1041 Machine->"M1" Host->"H1" VarMode->1 VarSet->"" VarDescr->"ARRAY[1,5]"
VarData->":1:2:3:4:5" Runtime->0.0056
07.08.2017 13:16:37    R_REPORT_H
OrderNum->1041 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 0
ResInt1->0 ResInt2->0 ResByte->""
```

15.2.11 Global STRING GUD variables

Definition

The use of global GUD variables is described below using an example.

The definition in "UGUD.DEF" is:

```
DEF NCK STRING[20] PALLET_NR

M30
```

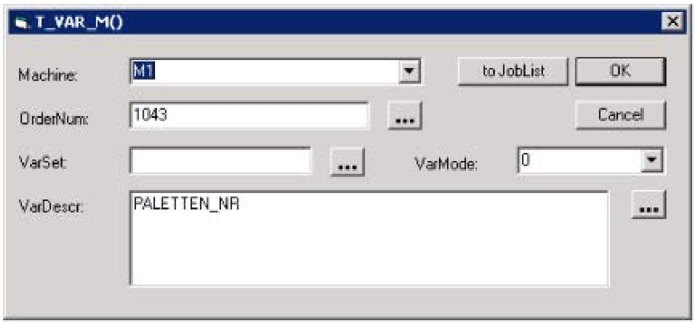
Note

Writing of strings

When writing strings, ensure that the first character of the data is a number or letter. Any other character is evaluated as a separator and leads to data falsification.

15.2.12 Reading global STRING GUD variables

Sequence



T_VAR_M()

Machine: M1

OrderNum: 1043

VarSet:

VarMode: 0

VarDescr: PALETTEN_NR

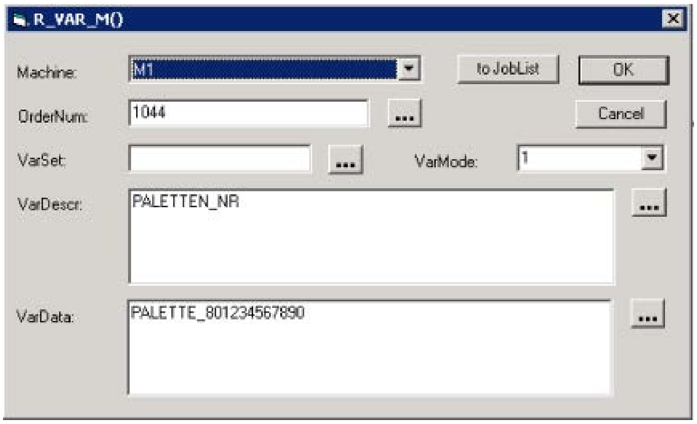
The global STRING GUD variable is read and the following "VarDescr:" is output:

Read global STRING GUD variable

```
07.08.2017 13:28:20    T_VAR_M    Ret=0
OrderNum->1043 Machine->"M1" Host->"H1" VarMode->"0" VarSet->"" VarDescr->"PALLET_NR"
Runtime->0.0066
07.08.2017 13:28:20    R_VAR_H
OrderNum->1043 Machine->"M1" Host->"H1" VarMode->0 VarSet->"Tmp" VarDescr->""
VarData->"PALETTE_801234567890" Runtime->0.0209
```

15.2.13 Writing global STRING GUD variables

Sequence



R_VAR_M()

Machine: M1

OrderNum: 1044

VarSet:

VarMode: 1

VarDescr: PALETTEN_NR

VarData: PALETTE_801234567890

The global STRING GUD variable is written to and the following "VarDescr:" is output:

Write global STRING GUD variable

```
07.08.2017 13:26:33    R_VAR_M    Ret=0
OrderNum->1044 Machine->"M1" Host->"H1" VarMode->1 VarSet->"" VarDescr->"PALLET_NR"
VarData->"PALETTE_801234567890" Runtime->0.0036
07.08.2017 13:26:33    R_REPORT_H
OrderNum->1046 Machine->"M1" Host->"H1" Typ->5 Number( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0
Time( 0 ... 9 ) -> 0 0 0 0 0 0 0 0 0 0 Flag( 0 ... 9 ) -> C 0 0 0 0 0 0 0 0 0
ResInt1->0 ResInt2->0 ResByte->""
```

CMI operator interface 16

16.1 Configuring CMI

16.1.1 Configuring CMI with "MCIS_RPCSLCFG"

Communication between Create MyInterface and the operator interface of Create MyInterface is implemented over a TCP/IP network connection and must be configured accordingly during commissioning.

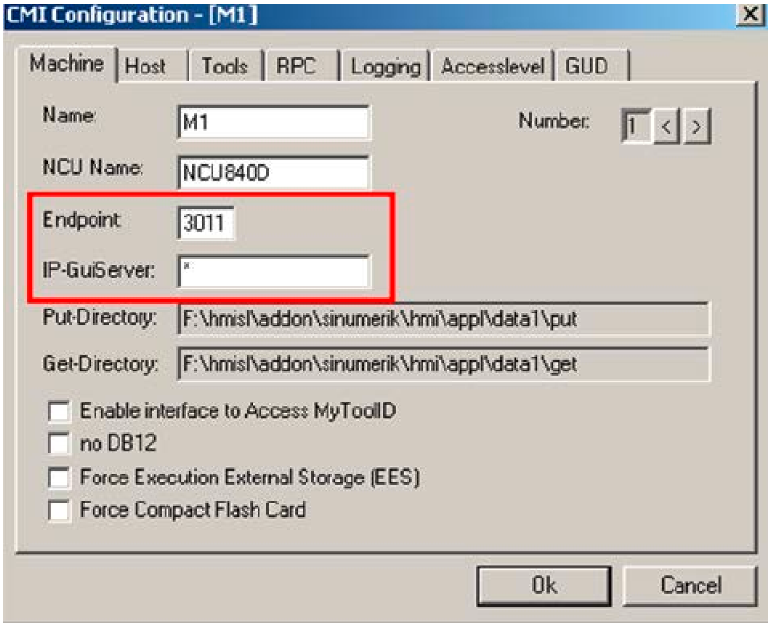
The configuration program "MCIS_RPCSLCFG.exe" is available for configuring the operator interface of Create MyInterface.

Parameter

Parameter	Description
IP-GuiServer	With the "*" identifier, the GUI server waits for arriving connections on all configured network adapters of the PCU 50 or Industrial PC.
Port number	It is not necessary to configure the port number since the port number of the GUI server is oriented to the machine endpoint. The port number of the GUI server is always the port number of the machine plus 100. Example: Endpoint 3011, GUI port number 3111

Procedure

1. Double-click the configuration program "MCIS_RPCSLCFG.exe" to start it.
2. Follow the configuration instructions for the individual tabs, see also section: Configuring CMI (Page 32).



Name: M1

Number: 1

NCU Name: NCU840D

Endpoint: 3011

IP-GuiServer: *

Put-Directory: F:\hmisl\addon\sinumerik\hmi\appl\data1\put

Get-Directory: F:\hmisl\addon\sinumerik\hmi\appl\data1\get

□ Enable interface to Access MyToolID

□ no DB12

□ Force Execution External Storage (EES)

□ Force Compact Flash Card

16.1.2 Configuring CMI on the controller

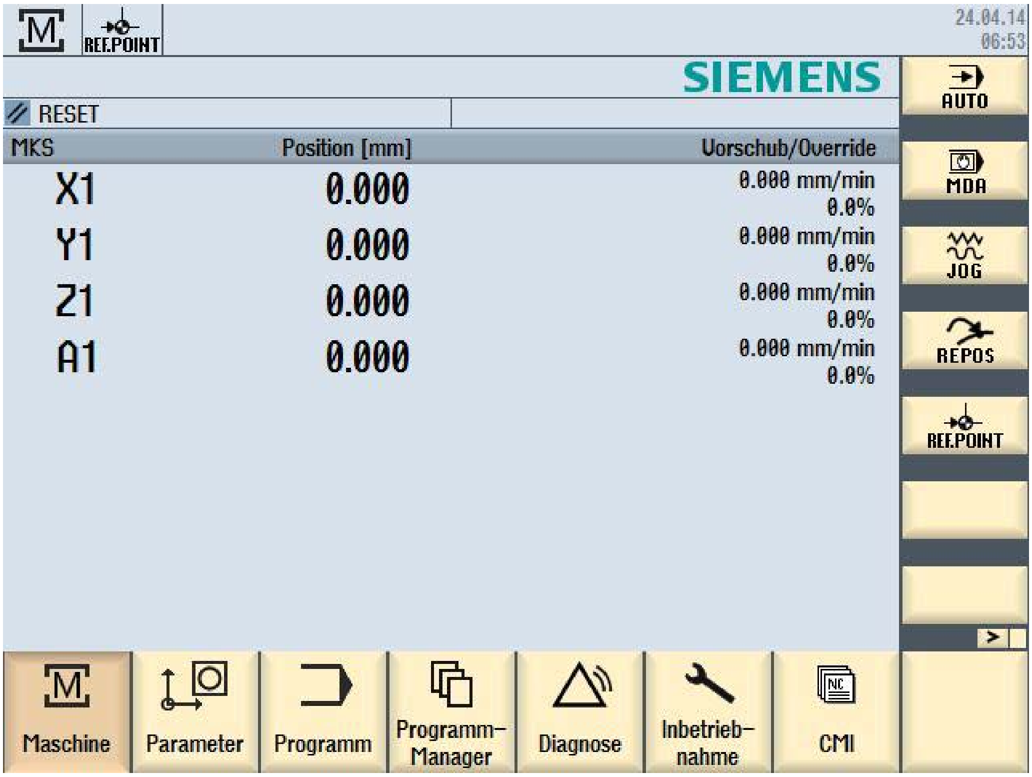
The dialogs of the "CMI" operating area are integrated as an add-on application in SINUMERIK Operate of the PCU 50/NCU.

Requirement

- The installation of the operator interface on an NCU always requires the installation of the actual CMI software on a Windows IPC. The actual CMI software cannot be installed on an NCU.
- The default setting is the X120 interface of the PCU 50.

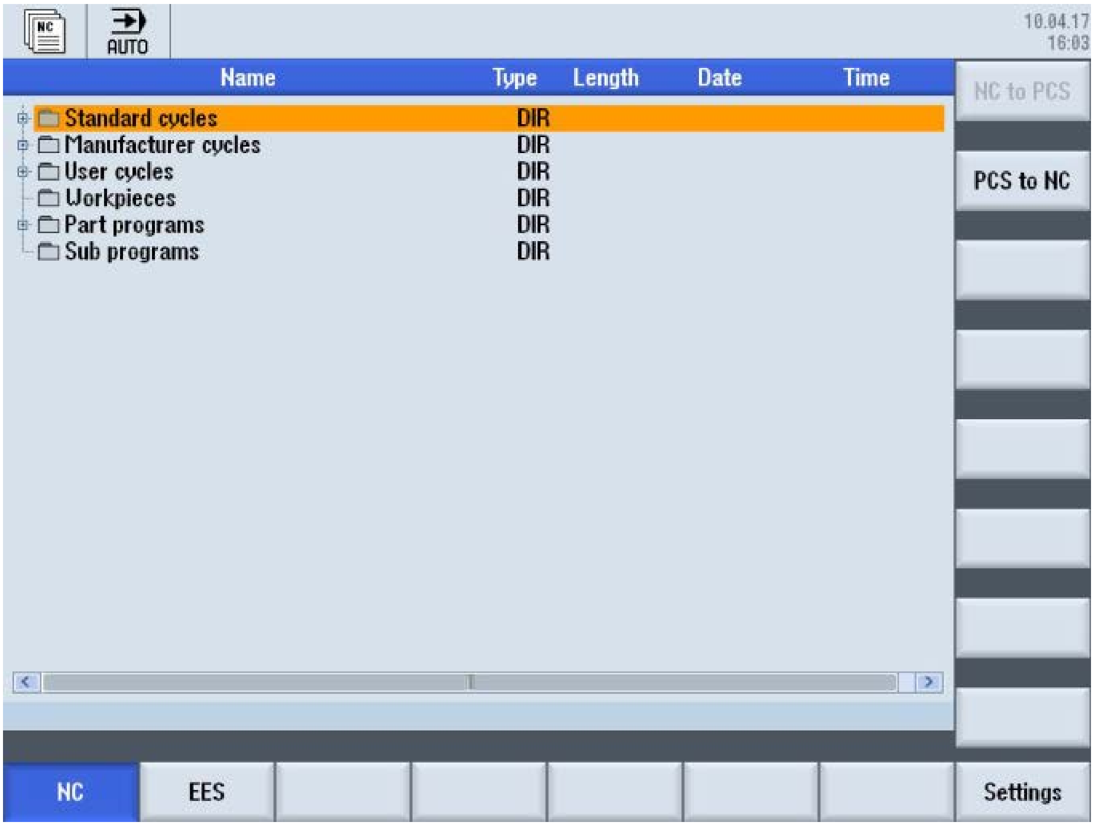
Procedure

1. Start SINUMERIK Operate.
2. Press the "CMI" softkey.



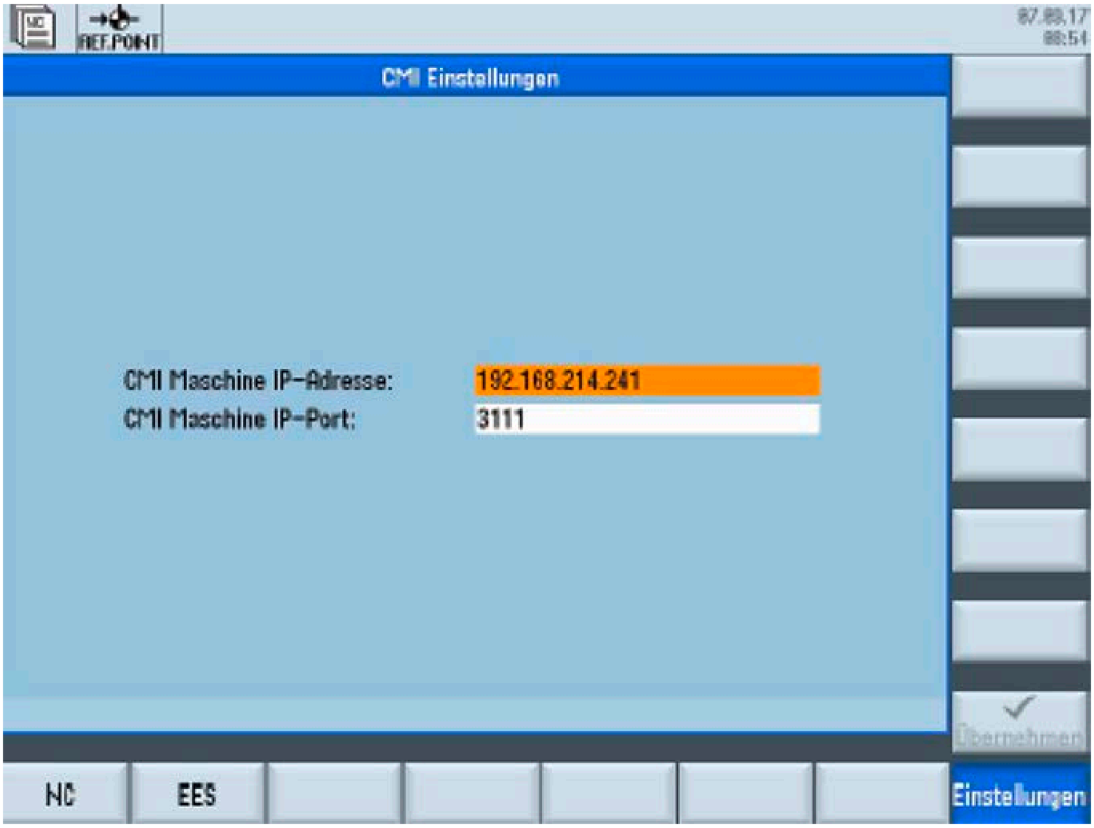
MKS	Position [mm]	Vorschub/Override
X1	0.000	0.000 mm/min 0.0%
Y1	0.000	0.000 mm/min 0.0%
Z1	0.000	0.000 mm/min 0.0%
A1	0.000	0.000 mm/min 0.0%

3. The window of the "NC" operating area opens.
   Press the "Settings" softkey.



Name	Type	Length	Date	Time
Standard cycles	DIR			
Manufacturer cycles	DIR			
User cycles	DIR			
Workpieces	DIR			
Part programs	DIR			
Sub programs	DIR			

4. The "CMI settings" window opens.
   Enter the required addresses:
   - IP address
   - Port number



CMI Einstellungen

CMI Maschine IP-Adresse: 192.168.214.241

CMI Maschine IP-Port: 3111

5. Press the "Accept" softkey.

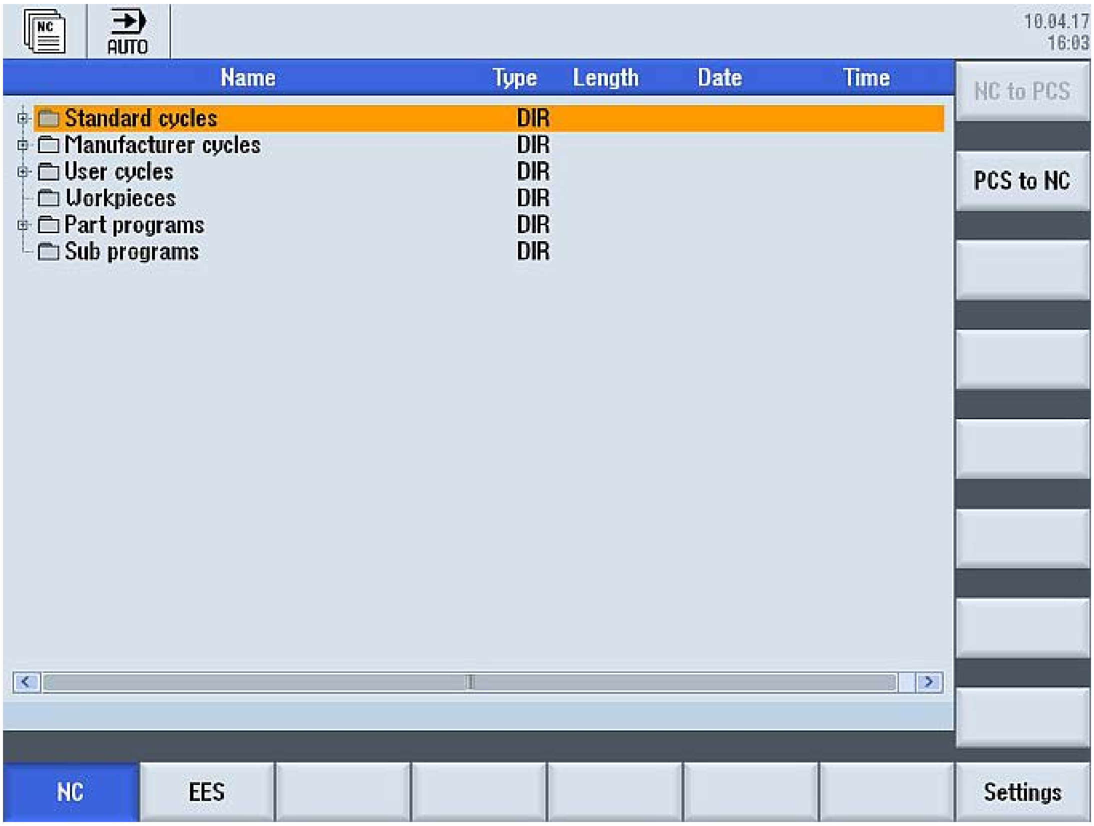
Additional information on the 'operating areas" can be found in the following section CMI operating area on the controller (Page 240).

16.2 CMI operating area on the controller

16.2.1 NC operating area

16.2.1.1 Overview

In this dialog you can transfer NC programs, workpiece directories, part programs, or cycles from the NC memory to and from the host computer.



Name	Type	Length	Date	Time
Standard cycles	DIR			
Manufacturer cycles	DIR			
User cycles	DIR			
Workpieces	DIR			
Part programs	DIR			
Sub programs	DIR			

16.2.1.2 Requesting an NC program from the host computer and loading it in the NC memory

This section uses an example to show how to request an NC program from the host computer and to load it in the NC memory of the machine immediately after the transfer.

Requirement

The CMI operating area is selected; see section: Configuring CMI on the controller (Page 236)

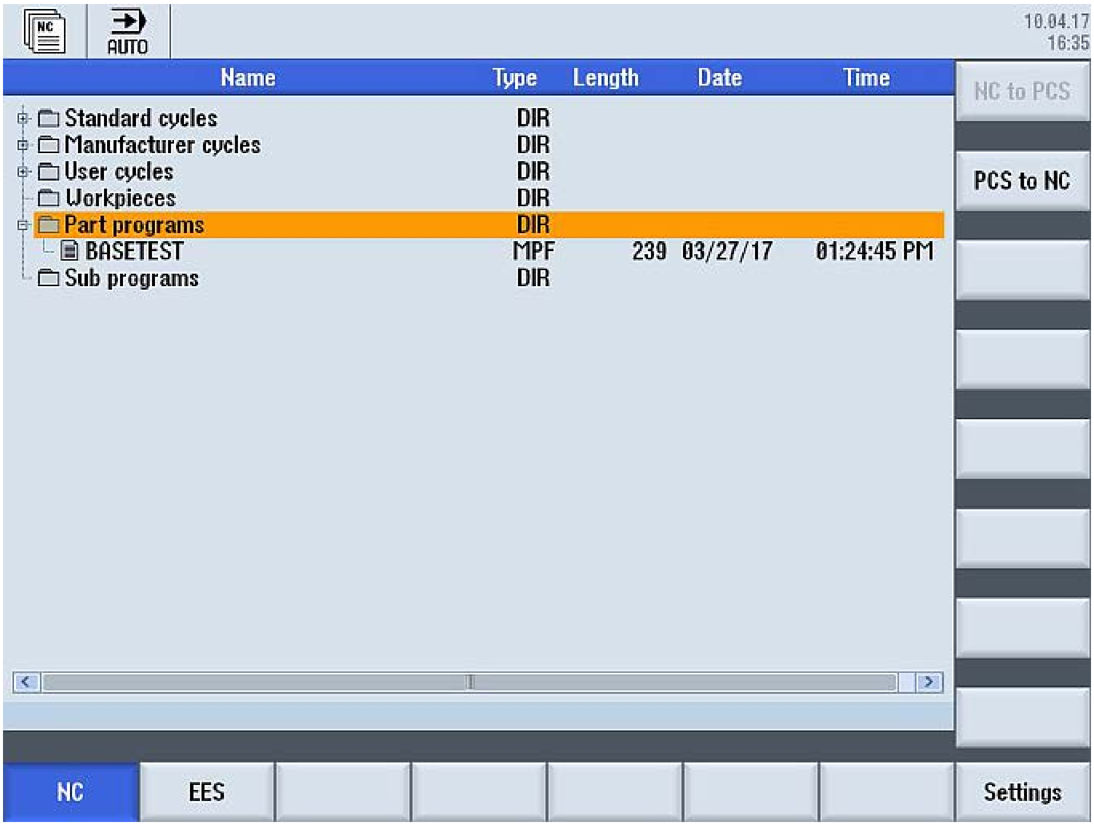
Note

NC programs

- You must select a destination in the data management schema.
- You can only request and transfer individual NC programs.

Procedure

1. Press the "NC" softkey.
2. Select the "Part programs" directory and press the "Host computer to NC" softkey.

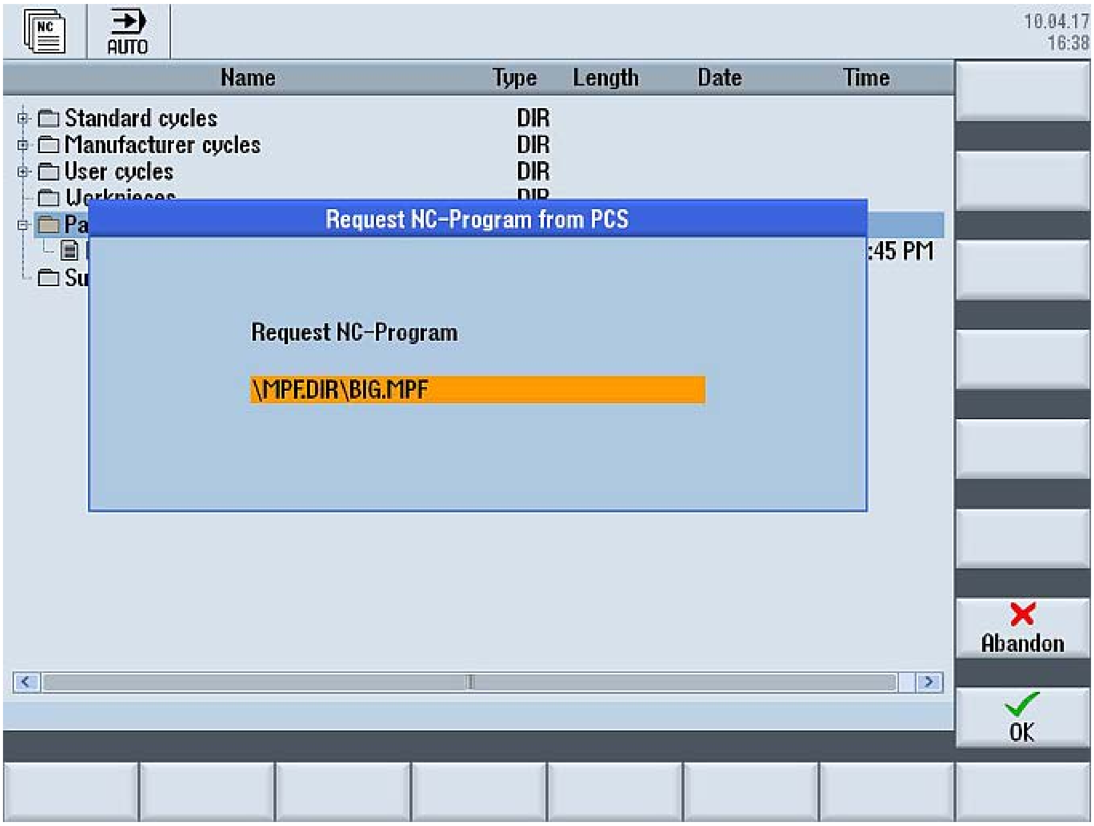


Name	Type	Length	Date	Time
Standard cycles	DIR			
Manufacturer cycles	DIR			
User cycles	DIR			
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
Sub programs	DIR			

3. The "Request NC program from host computer" window opens.
   - Enter an NC program name.
   - Press the vertical softkey "OK" to complete the operation.
     The order is sent to Create MyInterface to transfer the desired NC program to the machine and to then immediately load it in the machine.

     - OR -

     Press the vertical softkey "Cancel" to cancel the operation.



Request NC-Program from PCS

Request NC-Program

\MPF.DIR\BIG.MPF

16.2.1.3 Transferring an NC program from the NC memory to the host computer

This section uses an example to show how an NC program is transferred from the machine to the host computer.

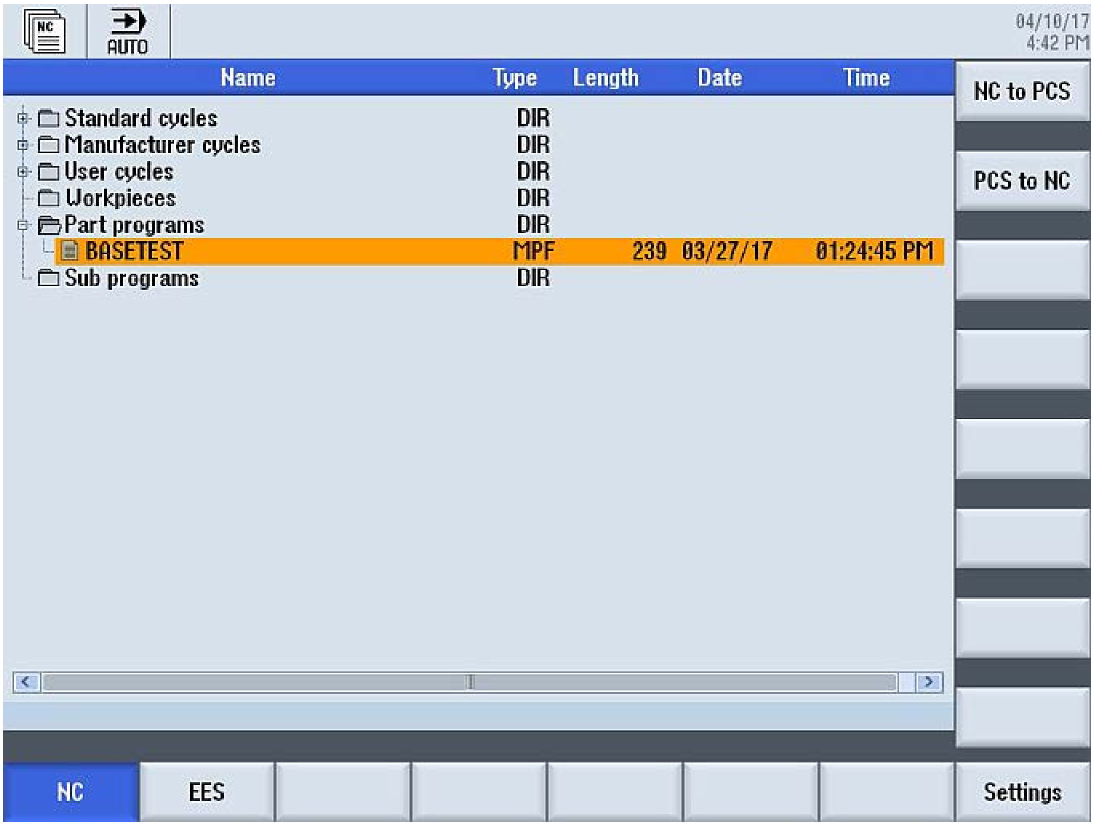
Requirement

The CMI operating area is selected; see section Configuring CMI on the controller (Page 236)

Procedure

1. Press the "NC" softkey.
2. Select an NC program or a workpiece directory in the "Part programs" directory.

   Press the vertical softkey "NC to host computer".



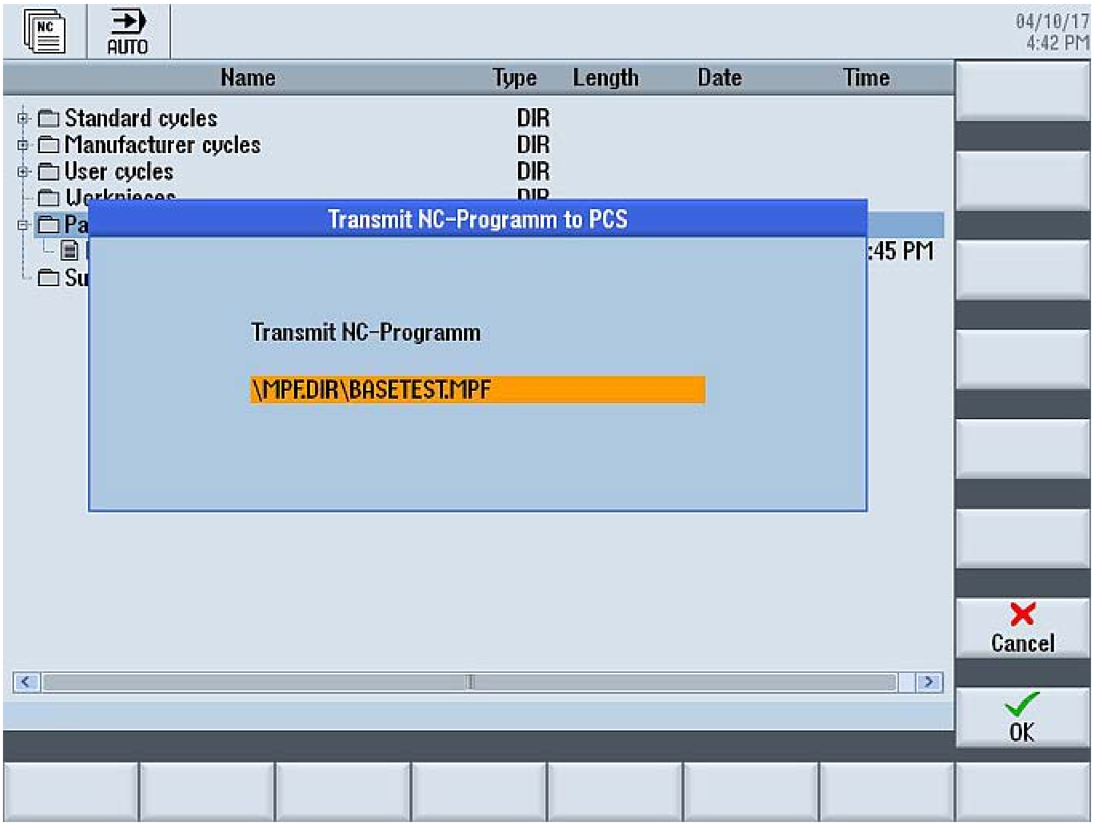
Name	Type	Length	Date	Time
Standard cycles	DIR			
Manufacturer cycles	DIR			
User cycles	DIR			
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
Sub programs	DIR			

3. The "Transfer NC program to host computer" window opens and shows the NC program selected for the transfer.

   – Press the "OK" softkey to start the operation.
     The order is sent to Create MyInterface to transfer the desired NC program from the machine to the host computer.

     - OR -

     Press the vertical softkey "Cancel" to cancel the operation.



Transmit NC-Programm to PCS

Transmit NC-Programm

\MPF.DIR\BASETEST.MPF

Name	Type	Length	Date	Time
Standard cycles	DIR			
Manufacturer cycles	DIR			
User cycles	DIR			

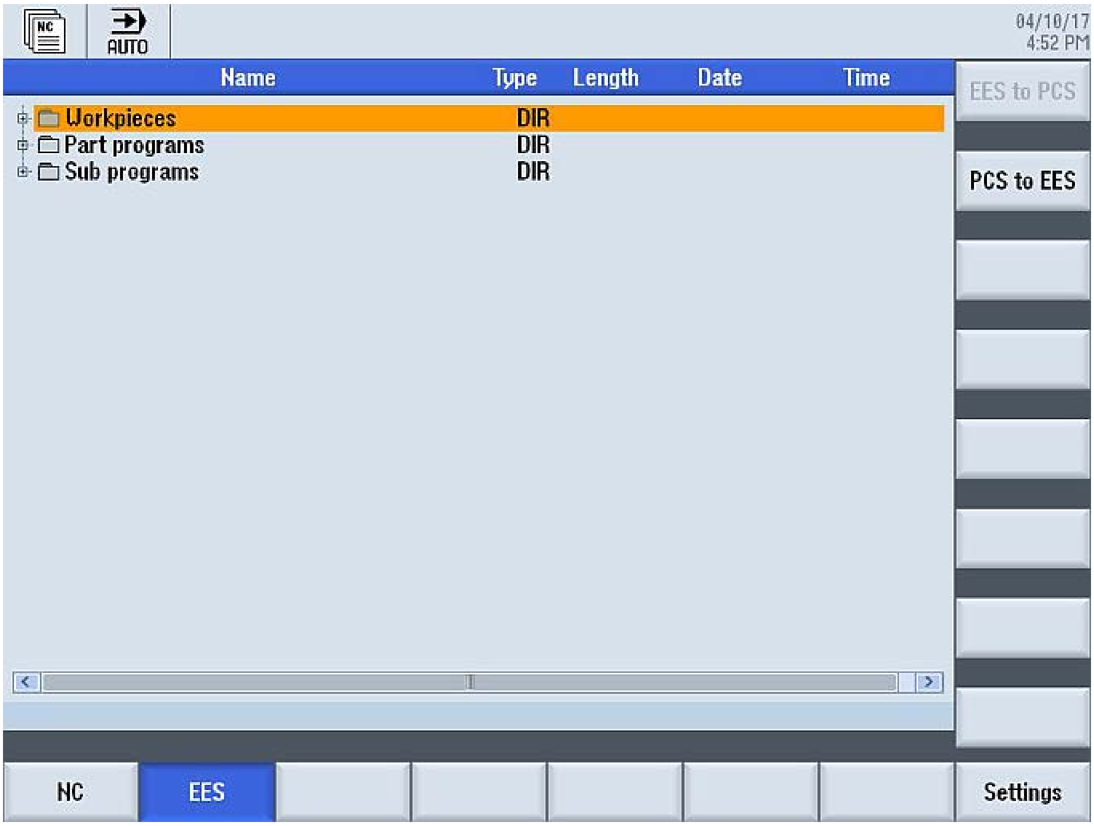
16.2.2 EES operating area

16.2.2.1 Overview

The "EES" (Execution External Storage) CMI operating area is only visible and selectable if the EES functionality was configured in the SINUMERIK Operate operating software.

In this dialog box, you can transfer NC programs or workpiece directories from the EES memory to and from the host computer.

The dialog of the "EES" CMI operating area allows you to navigate through the programs, subprograms, cycles and workpieces that have been loaded in the EES memory.



Name	Type	Length	Date	Time
Workpieces	DIR			
Part programs	DIR			
Sub programs	DIR			

16.2.2.2 Requesting an NC program from the host computer and storing it in the EES memory

This section uses an example to show how to request an NC program from the host computer and to store it in the EES memory of the machine immediately after the transfer.

A destination must be selected in the data management schema with the selection dialog.

Requirement

The CMI operating area is selected; see section: Configuring CMI on the controller (Page 236)

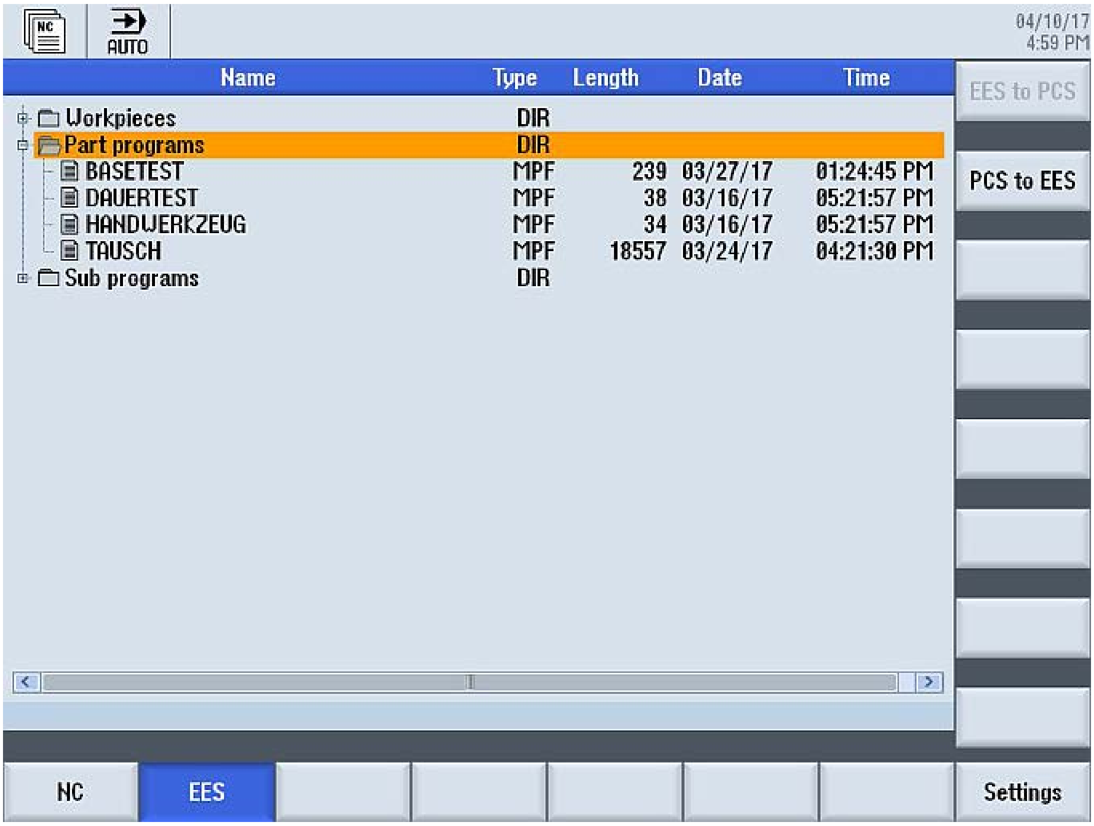
Procedure

1. Press the "EES" softkey.
2. Select the "Part programs" directory and press the "Host computer to EES" softkey.

Note

NC programs

- You must select a destination in the data management schema.
- You can only request and transfer individual NC programs.



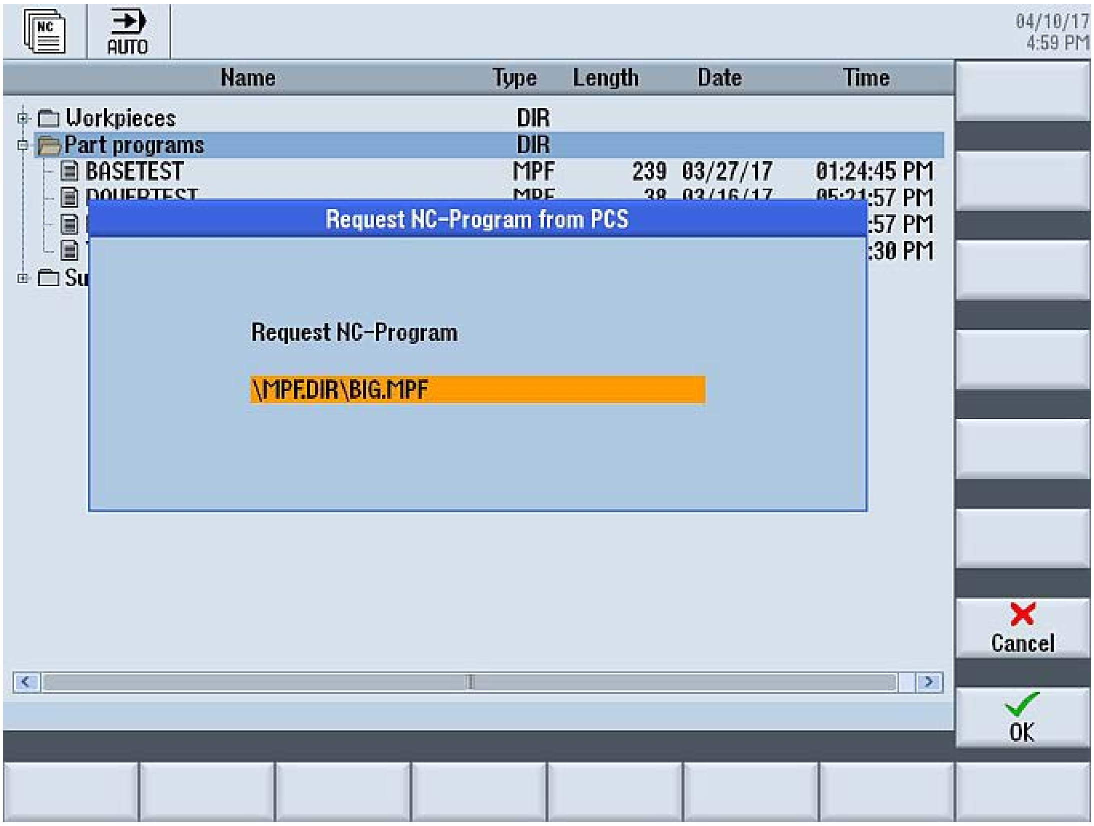
Name	Type	Length	Date	Time
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
DAUERTEST	MPF	38	03/16/17	05:21:57 PM
HANDWERKZEUG	MPF	34	03/16/17	05:21:57 PM
TAUSCH	MPF	18557	03/24/17	04:21:30 PM
Sub programs	DIR			

3. The "Request NC program from host computer" window opens.
   - Enter an NC program name.
   - Press the vertical softkey "OK" to complete the operation.

     The order is sent to Create MyInterface to transfer the desired NC program to the machine and to then immediately store it in the EES memory.

     - OR -

     Press the vertical softkey "Cancel" to cancel the operation.



Name	Type	Length	Date	Time
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
DOVERTEST	MPF	38	03/16/17	05:21:57 PM
				:57 PM
				:30 PM
Su				

Request NC-Program from PCS

Request NC-Program

\MPF.DIR\BIG.MPF

16.2.2.3 Transferring an NC program from the EES memory to the host computer

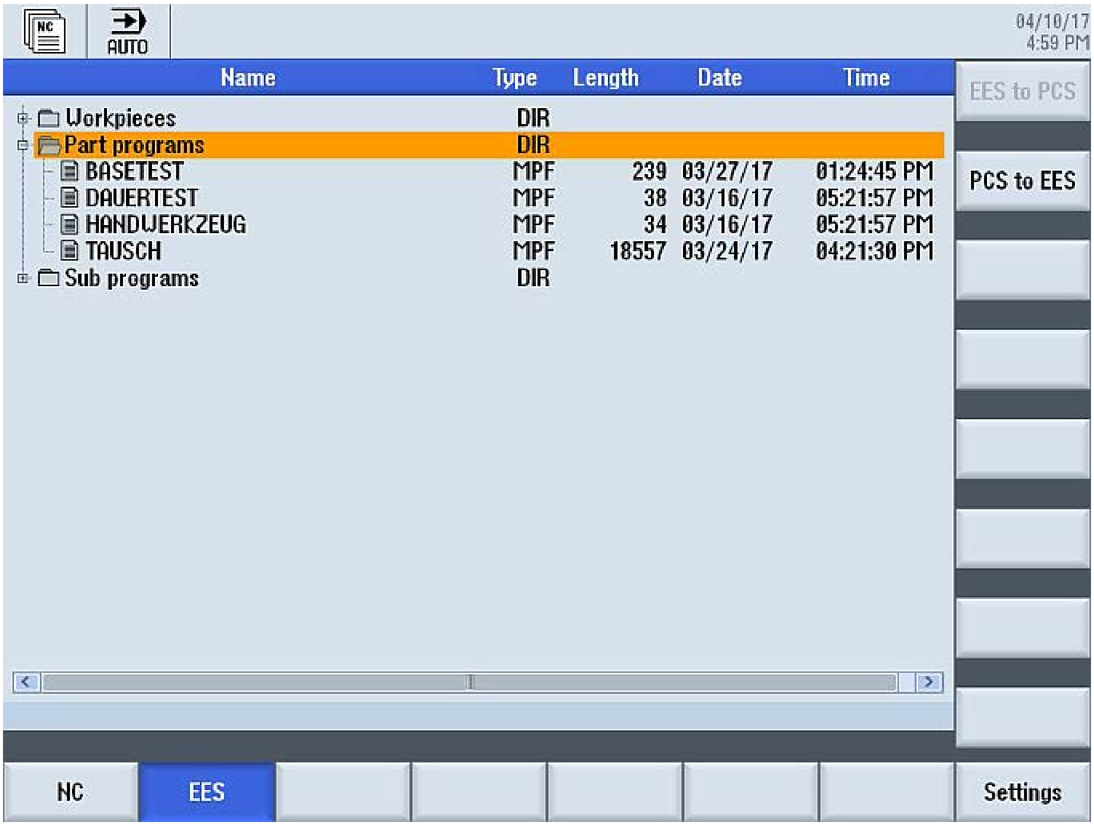
This section uses an example to show how an NC program is transferred from the machine to the host computer.

Requirement

The CMI operating area is selected; see section: Configuring CMI on the controller (Page 236)

Procedure

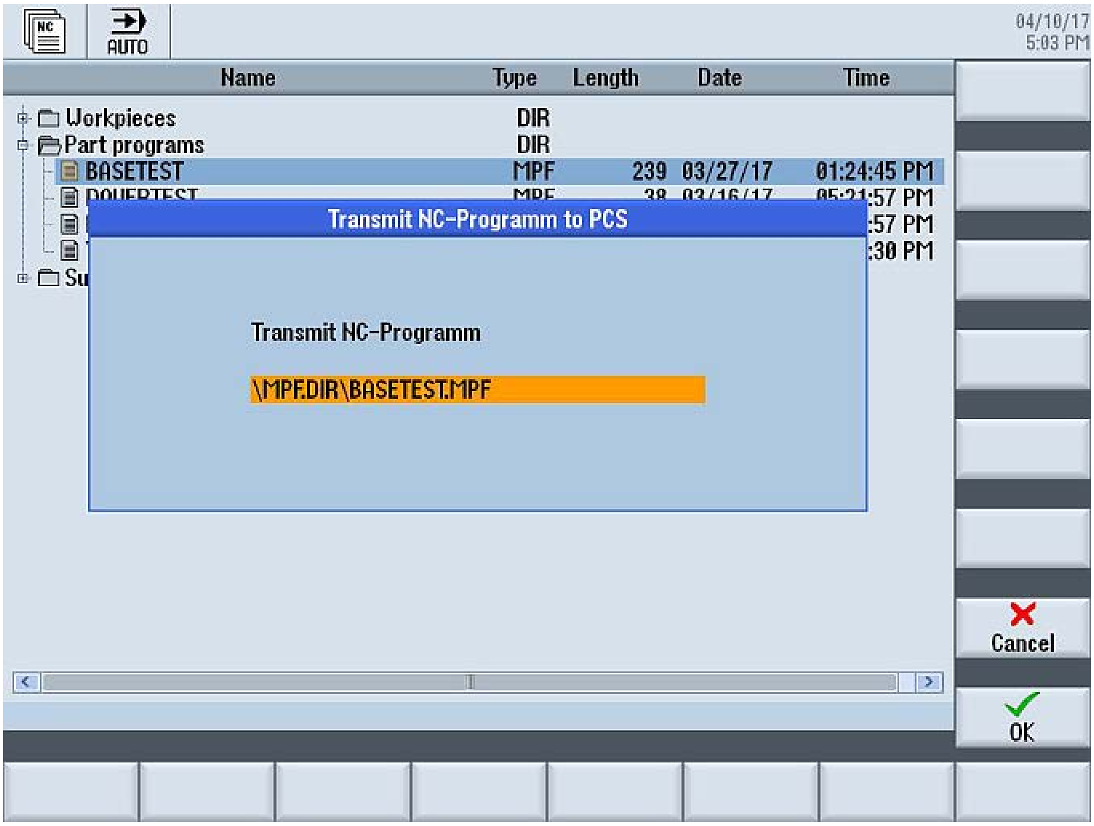
1. Press the "EES" softkey.
2. Select an NC program or a workpiece directory in the "Part programs" directory.



Name	Type	Length	Date	Time
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
DAUERTEST	MPF	38	03/16/17	05:21:57 PM
HANDWERKZEUG	MPF	34	03/16/17	05:21:57 PM
TAUSCH	MPF	18557	03/24/17	04:21:30 PM
Sub programs	DIR			

3. Press the vertical softkey "EES to host computer".

   The "Transfer NC program to host computer program" window opens and shows the selected NC program or workpiece directory.



Name	Type	Length	Date	Time
Workpieces	DIR			
Part programs	DIR			
BASETEST	MPF	239	03/27/17	01:24:45 PM
DOUERTEST	MPF	38	03/16/17	05:21:57 PM
				:57 PM
				:30 PM
Su				

Transmit NC-Programm to PCS

Transmit NC-Programm

\MPF.DIR\BASETEST.MPF

4. Press the vertical softkey "OK" to complete the operation.

   The order is sent to Create MyInterface to transfer the desired NC program from the EES memory to the host computer.

   - OR -

   Press the vertical softkey "Cancel" to cancel the operation.

16.2.3 Keyboard assignment

16.2.3.1 Navigation mode

This section provides information about the key assignment in navigation mode.

Keys	Description
Cursor left	Closes the selected data management schema or workpiece directory
Cursor right	Opens the selected data management schema or workpiece directory
Cursor up	Selects the previous NC program, data management schema or workpiece directory
Cursor down	Selects the next NC program, data management schema or workpiece directory
Return	Opens/closes the selected data management schema or workpiece directory
Page up	Selects the entry on the previous page
Page down	Selects the entry on the next page

16.2.3.2 Editing mode

This section provides information about the key assignment in editing mode.

Keys	Description
Cursor left	Moves the cursor to the left by one character
Shift + cursor left	Moves the cursor to the left by one character and selects the text
Cursor right	Moves the cursor to the right by one character
Shift + cursor right	Moves the cursor to the right by one character and selects the text
Backspace	Deletes the character left of the cursor
Ctrl + Backspace	Deletes the word left of the cursor
Delete	Deletes the character right of the cursor
Ctrl + Delete	Deletes the word right of the cursor
Ctrl + C	Copies the selected text to the clipboard
Ctrl + V	Pastes the content of the clipboard
Ctrl + X	Deletes the selected text and copies it to the clipboard
Insert	Toggles between editing mode and navigation mode
Return	Exits editing mode

Function calls of EES and CF card 17

17.1 Function EES (Execution External Storage) for CMI

Requirement

The EES functionality with Create MyInterface can only be used when CMI has been installed on a PCU 50 / Industrial PC with SINUMERIK Operate Version 4.7 SP2 HF1 or higher.

EES must be ready to operate, i.e. configured and licensed on the controller.

Software option

To have unlimited use of this function, you require the software option "Execution from External Storage (EES)."

Function

Note

EES memory

If "Force Execution External Storage (EES)" was configured, NC programs are executed exclusively in the EES memory!

17.1.1 Transferring the NC program

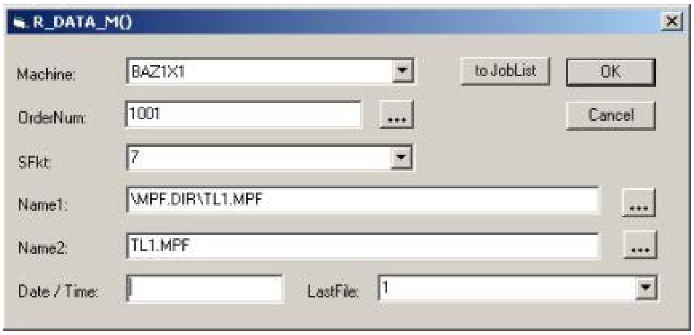
With the general transfer function `R_DATA_M SFkt = 7`, the NC program indicated under "Name1:" is transferred to the PCU 50.

If the file already exists in the data management, transfer is rejected.

Function

Table 17- 1 Subfunction number: SFkt for transfer NC program

SFkt	Function	Remark
7	Transferring the NC program	Parameter required, see SFkt = 1 Summary of the RPC calls (Page 153)



Machine: BAZ1X1
OrderNum: 1001
SFkt: 7
Name1: \MPF.DIR\TL1.MPF
Name2: TL1.MPF
Date / Time:
LastFile: 1

Positive acknowledgment: If the file does not exist in the NC or in the MMC, the function is acknowledged with RPC call `R_REPORT_H type = 5 number[0]=7`.

Negative acknowledgment: If the file does not exist in the NC, the function is acknowledged with RPC call `R_REPORT_H Typ = 4 Number[0]=-310`.

Negative acknowledgment: If the file does not exist in the MMC, the function is acknowledged with RPC call `R_REPORT_H Typ = 4 Number[0]=-330`.

`R_REPORT_H`

```
OrderNum->1002 Machine->"BAZ1X1" Host->"H1" Typ->5

Number(0..9) -> 7 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->0.0142
```

Result

`Type :=` 5 `- Identifier of positive acknowledgment`

`Number(0) :=` 7 `- Initiated SFkt`

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, you must use the latest version of the program!

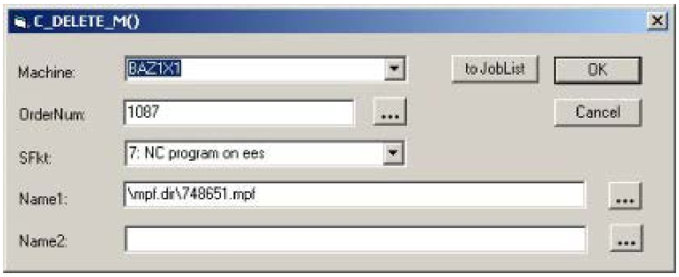
17.1.2 Transferring the NC program back

With the general transfer function `R_DATA_M SFkt = 7`, the NC program indicated under "Name1:" is transferred to the PCU 50.

Function

Table 17- 2 Subfunction number: SFkt for transfer NC program back

SFkt	Function	Remark
7	Transferring the NC program back	Parameter required, see SFkt = 1 Summary of the RPC calls (Page 153)


C_DELETE_M()

Machine: BAZ1X1

OrderNum: 1087

SFkt: 7: NC program on ees

Name1: \mpf.dir\748651.mpf

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 5` to the host computer as positive acknowledgment.

`R_Data_H`

```
OrderNum->1002 Machine->"BAZ1X1" Host->"H1" SFkt->1

Name1->"\mpf.dir\748651.mpf"

Name2->"\\192.168.5.228\RPCHostDirectory\Putdir\748651.mpf"

datVal->1465379838 LastFile->0

Runtime->0.0142
```

Result

The NC program is stored in the configured host directory.

RPC SINUMERIK sends RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-320 - NC program not in data management.

Note

Use the new version of the "RPC SINUMERIK Test" program

If you use the "RPC SinumerikTest.exe" program to test the function, you must use the latest version of the program!

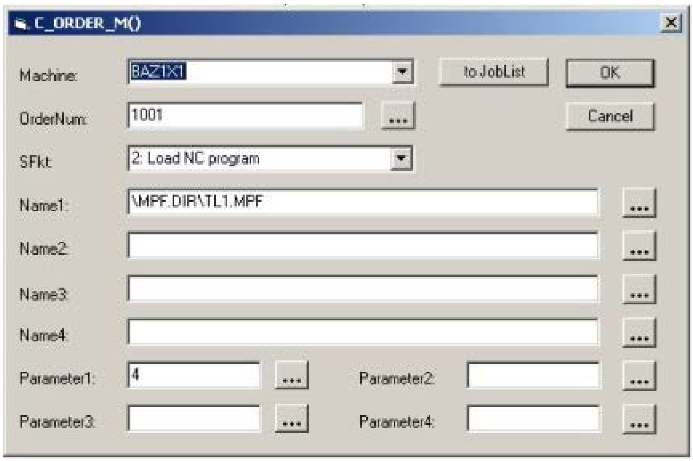
17.1.3 Loading the NC program in the EES memory

With the general order function `C_ORDER_M SFkt = 2`, Parameter1 = 4 the NC program indicated under "Name1:" is loaded in the EES memory.

Function

Table 17- 3 Subfunction number: SFkt for load NC program in the EES memory

SFkt	Function	Remark
2	Loading the NC program in the EES memory	Parameter1 = 4 (destination EES)



C_ORDER_M()

Machine: BAZ1X1

OrderNum: 1001

SFkt: 2: Load NC program

Name1: \MPF.DIR\TL1.MPF

Name2:

Name3:

Name4:

Parameter1: 4

Parameter2:

Parameter3:

Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

`R_REPORT_H`

```
OrderNum->1003 Machine->"BAZ1X1" Host->"H1" Typ->6

Number(0..9) -> 2 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->0.016
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 2 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-262 - NC program cannot be loaded in EES memory

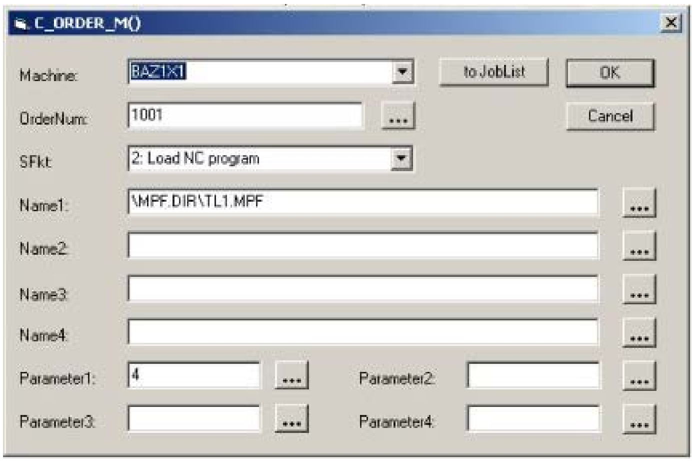
17.1.4 Load workpiece in the EES memory

With the general order function `C_ORDER_M SFkt = 2`, Parameter1 = 6 the NC program indicated under "Name1:" is loaded in the EES memory.

Function

Table 17- 4 Subfunction number: SFkt for load NC program in the EES memory

SFkt	Function	Remark
2	Load workpiece in the EES memory	Parameter1 = 6 (destination EES)



Machine: BAZ1X1
OrderNum: 1001
SFkt: 2: Load NC program
Name1: \MPF.DIR\TL1.MPF
Name2:
Name3:
Name4:
Parameter1: 4
Parameter2:
Parameter3:
Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1003 Machine->"BAZ1X1" Host->"H1" Typ->6

Number(0..9) -> 2 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->0.016
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 2 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-262 - NC program cannot be loaded in EES memory

17.1.5 Selecting the NC program in the EES memory

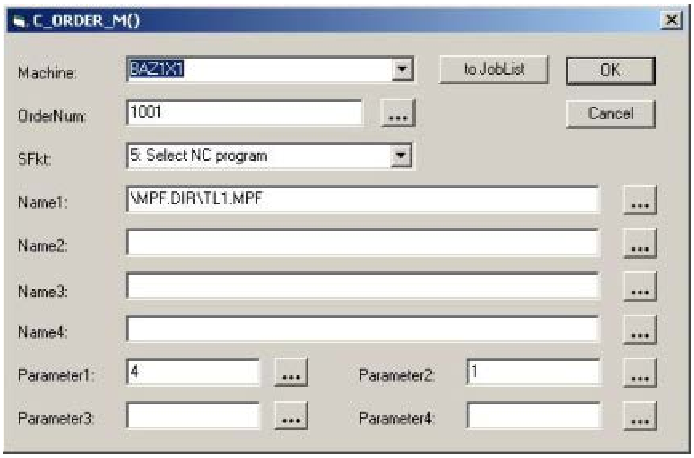
With the general order function `C_ORDER_M SFkt = 5`, Parameter1 = 4, the NC program indicated under "Name1:" is selected for processing in the EES memory.

Function

Table 17- 5 Subfunction number: SFkt for select NC program in the EES memory

SFkt	Function	Remark

5	Selecting the NC program in the EES memory	Parameter1 = 4 (destination EES)
		Parameter2 = 1 (channel)



Machine: BAZ1X1
OrderNum: 1001
SFkt: 5: Select NC program
Name1: \MPF.DIR\TL1.MPF
Name2:
Name3:
Name4:
Parameter1: 4
Parameter2: 1
Parameter3:
Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1001 Machine->"BAZ1X1" Host->"H1" Typ->6

Number(0..9) -> 5 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->1.0355
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 5 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-265 - NC program cannot be selected in the EES memory

17.1.6 Deselecting the NC program in the EES memory

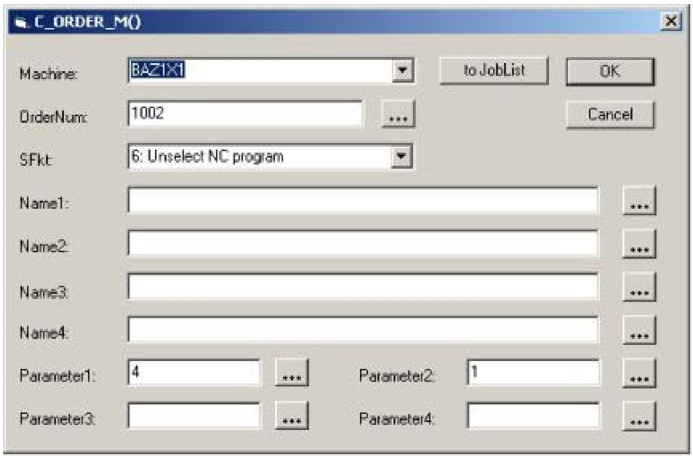
With the general order function `C_ORDER_M SFkt = 6`, Parameter1 = 4 the selected NC program for the channel indicated in "Parameter2:" is deselected.

Function

Table 17- 6 Subfunction number: SFkt for deselect NC program in the EES memory

SFkt	Function	Remark

6	Deselecting the NC program in the EES memory	Parameter1 = 4 (destination EES)
		Parameter2 = 1 (channel)



Machine: BAZ1X1
OrderNum: 1002
SFkt: 6: Unselect NC program
Name1:
Name2:
Name3:
Name4:
Parameter1: 4
Parameter2: 1
Parameter3:
Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1002 Machine->"BAZ1X1" Host->"H1" Typ->6

Number(0..9) -> 6 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->1.0206
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 6 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-266 - NC program cannot be deselected.

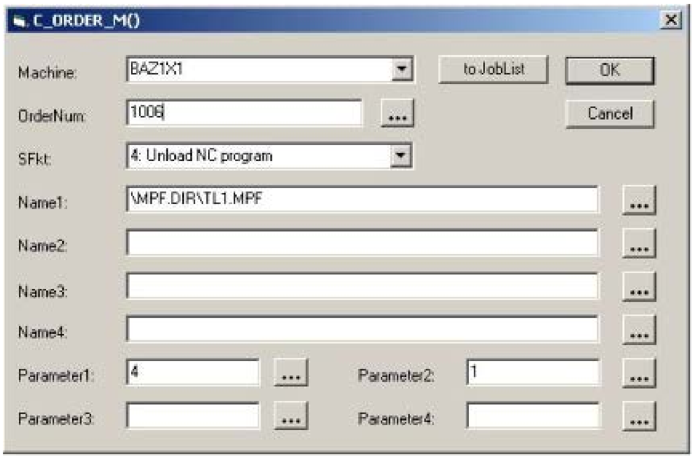
17.1.7 Unload NC program from the EES memory

With the general order function `C_ORDER_M SFkt = 4`, Parameter1 = 4 the NC program indicated under "Name1:" is unloaded from the EES memory.

Function

Table 17- 7 Subfunction number: SFkt for unload NC program from the EES memory

SFkt	Function	Remark
4	Unload NC program from the EES memory	Parameter1 = 4 (destination EES)



C_ORDER_M()

Machine: BAZ1X1

OrderNum: 1006

SFkt: 4: Unload NC program

Name1: \MPF.DIR\TL1.MPF

Name2:

Name3:

Name4:

Parameter1: 4

Parameter2: 1

Parameter3:

Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1002 Machine->"BAZ1X1" Host->"H1" Typ->6

Number(0..9) -> 4 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->1.1721
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 4 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-264 - NC program cannot be deselected.

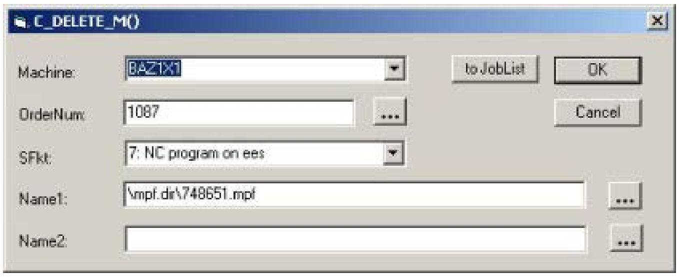
17.1.8 Deleting the NC program from the EES memory

With the general order function `C_ORDER_M SFkt = 7`, the NC program indicated under "Name1:" is deleted in the EES memory.

Function

Table 17- 8 Subfunction number: SFkt for delete NC program from the EES memory

SFkt	Function	Remark
7	Deleting the NC program from the EES memory	Parameter required, see SFkt = 1 Table 9-3 SFkt for data dialogs (Page 154)


C_DELETE_M()

Machine: BAZ1X1

OrderNum: 1087

SFkt: 7: NC program on ees

Name1: \mpf.dir\748651.mpf

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type=5` to the host computer as positive acknowledgment.

```
R_REPORT_H

OrderNum->1003 Machine->"BAZ1X1" Host->"H1" Typ->5

Number(0..9) -> 7 0 0 0 0 0 0 0 0 0

Time(0..9) -> 0 0 0 0 0 0 0 0 0 0

Flag(0..9) -> C 0 0 0 0 0 0 0 0 0

ResInt1->0 ResInt2->0 ResByte->"" Runtime->0.1721
```

Result

`Type :=` 5 `- Identifier of positive acknowledgment`

`Number(0) :=` 7 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type=4` to the host computer as negative acknowledgment.

Error number

-250 - NC program cannot be deleted.

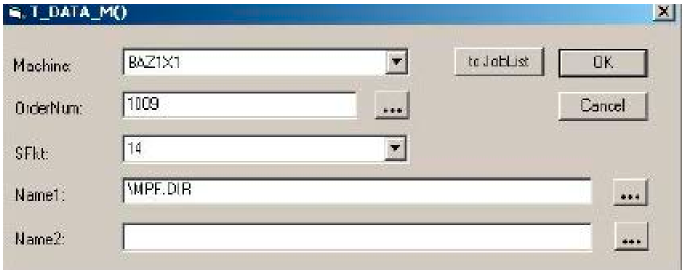
17.1.9 Listing the NC program(s) in the EES memory

With the general order function `R_DATA_M SFkt = 14`, the NC program directory indicated under "Name1:" is read in the EES memory of the PCU 50 and transferred to the host computer as a file.

Function

Table 17- 9 Subfunction number: SFkt for list NC program in the EES memory

SFkt	Function	Remark
14	List NC programs in the EES memory	Parameters required, see SFkt = 10 Table 9-3 SFkt for data dialogs (Page 154)



Machine: BAZ1X1
OrderNum: 1009
SFkt: 14
Name1: \MPF.DIR
Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 14` to the host computer as positive acknowledgment.

```
R_Data_H

OrderNum->1008 Machine->"BAZ1X1" Host->"H1" SFkt->14

Name1->"\mpf.dir"

Name2->"\\192.168.5.228\RPCHostDirectory\Putdir\NCPrList.001"

DateVal->0 LastFile->0 Runtime->0.1785
```

Result

SFkt = 14 - Positive acknowledgment identifier

File structure

The file transferred to the host computer has the following structure:

The first row of the file contains the requested directory name.

The following rows list the NC programs contained in the directory. Each row represents one NC program.

Row 1: \MPF.DIR

Row X: TL1.MPF,FN,201,1449237494

Program row structure

Each program row has the following structure:

- Each column is separated by a comma.
- Type and location:
  - F - File
  - D- Directory
  - M - NC program on the PCU/IPC
  - N- NC program in the EES memory

Example:

```
NC program name ,type and location   ,length   ,date and time of last change (UNIX)
TL1.MPF         ,FN                  ,201      ,1449237494
```

17.1.10 Program assignment (Execution External Storage)

Function

Function call

```
R_NC4WPC_M(                Host,
                           Machine,
                           OrderNum,
                           WPC,
                           NCProg,
                           Date,
                           NCPLength,
                           ClampCubeSide,
                           TpFlag,
                           NCExtern,
                           Resint1,
                           Resint2,
                           Resbyte)
```

Direction of transfer Host computer → SINUMERIK

Instructs the machine which program is to be activated.

Table 17- 10 Parameters for program activation

<table>
<tr><th>Parameter</th><th colspan="4">Description</th><th>Format</th></tr>
<tr><td>Host</td><td colspan="4">Name of the host computer</td><td>16 bytes (string)</td></tr>
<tr><td>Machine</td><td colspan="4">Machine name</td><td>16 bytes (string)</td></tr>
<tr><td>OrderNum</td><td colspan="4">Order number</td><td>4 bytes (long int)</td></tr>
<tr><td>WPC</td><td colspan="4">Workpiece holder designation</td><td>6 bytes (string)</td></tr>
<tr><td>NCProg</td><td colspan="4">NC program
NC program in the form:
"\mpf.dir\Zylinderkopf.mpf"</td><td>128 bytes (string)</td></tr>
<tr><td>Date</td><td colspan="4">Date of the last change of the NC program</td><td>4 bytes (long int)</td></tr>
<tr><td>NCPLength</td><td colspan="4">NC program size in bytes</td><td>4 bytes (long int)</td></tr>
<tr><td>TpFlag</td><td colspan="4">Transport flag
= 0: No follow-up operation
= 1: Follow-up operation
= 9: Only for buffering</td><td>4 bytes (long int)</td></tr>
<tr><td rowspan="8">NCExtern</td><td colspan="4">Process from external source</td><td></td></tr>
<tr><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td><td>Function</td><td rowspan="7">4 bytes (long int)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Process NC program on NC</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Process NC program from external source</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Process workpiece on NC</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Process workpiece from external source</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Process NC program from EES</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Process NC program from EES</td></tr>
<tr><td>Resint1</td><td colspan="4">Reserve 1</td><td>4 bytes (long int)</td></tr>
<tr><td>Resint2</td><td colspan="4">Reserve 2</td><td>4 bytes (long int)</td></tr>
<tr><td>Resbyte</td><td colspan="4">Reserve 3</td><td>8 bytes (string)</td></tr>
</table>

- This RPC may occur multiple times for a workpiece holder, if each side of a clamp cube is machined with a separate NC program. The sides are machined in the order that the program assignments are reported from the host computer to RPC SINUMERIK.
- For the program assignments of a workpiece holder, the transport flag "1 = Follow-up operation" must be set for all except the last. If the follow-up operation flag is set for a side, the workpiece holder remains at the machining station after the end of the NC program. Because this flag is not set for the last side, it is possible to recognize from this that no more machining will follow and the workpiece holder can be moved out of the machining station.

- If a workpiece holder is located in the machine only for buffering purposes, this can be reported with transport flag "9 = Only for buffering". In this case, an NC program is not specified.
- If a workpiece is processed from the EES memory (Execution External Storage), this requires one of two configured and licensed options:

Software option

To have unlimited use of this function, you require the software option "Execution from External Storage (EES)."

6FC5800-0AP75-0YB0

Software option

To use this function together with the user memory (100 MB) on the Compact Flash Card, you require the software option "CNC extended user memory."

6FC5800-0AP77-0YB0

If the EES configuration is incorrect, the workpiece holder status change from "4 = Prepare program selection" to "8 = Program selection complete" is **omitted**.

Note

Create MyInterface on an Industrial PC (IPC)

When Create MyInterface is used on an Industrial PC (IPC), the processing of a workpiece from EES occurs only from the CMI instance that represents the "Default machine".

17.2 Function CF card (CompactFlash card) for CMI

The Compact Flash card is used in Create MyInterface to transfer, transfer back, load, deselect, list or delete NC programs on the CF card of the SINUMERIK 840D sl.

Note

NC program on the CF card

It is not possible to select an NC program on the CF card.

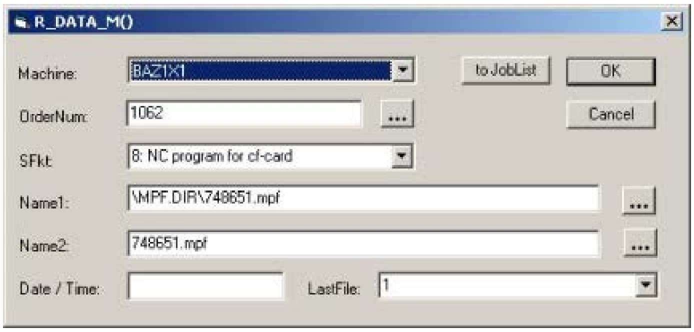
17.2.1 Transferring the NC program

With the general transfer function `R_DATA_M SFkt = 8`, the NC program indicated under "Name1:" is transferred to the PCU 50/Industrial PC.

Function

Table 17- 11 Subfunction number: SFkt for transfer NC program CF card

SFkt	Function	Remark
8	Transferring the NC program	Parameter required, see Fkt = 1 Function calls from SINUMERIK to the host computer (Page 154)



Machine: BAZ1X1
OrderNum: 1062
SFkt: 8: NC program for cf-card
Name1: \MPF.DIR\748651.mpf
Name2: 748651.mpf
Date / Time:
LastFile: 1

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 5` to the host computer as positive acknowledgment.

R_REPORT_H

```
R_REPORT_H
OrderNum->1002  Machine->"BAZ1X1"  Host->"H1" Typ->5
Number(0..9) -> 8 0 0 0 0 0 0 0 0 0
Time(0..9)   -> 0 0 0 0 0 0 0 0 0 0
Flag(0..9)   -> C 0 0 0 0 0 0 0 0 0
ResInt1->0  ResInt2->0  ResByte->""  Runtime->0.0142
```

Result

```
Type := 5 - Identifier of positive acknowledgment

Number(0) := 8 - Initiated SFkt
```

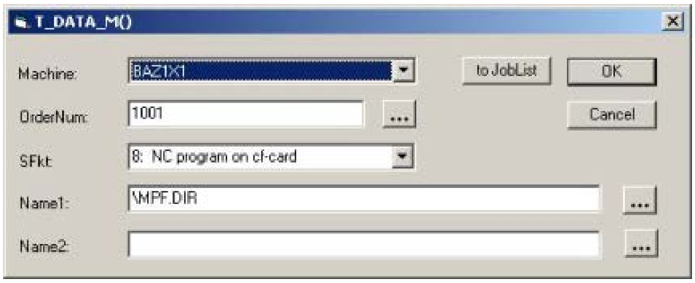
17.2.2 Transferring the NC program back

With the general transfer function `R_DATA_M SFkt = 8`, the NC program indicated under "Name1:" is transferred to the PCU 50/Industrial PC.

Function

Table 17- 12 Subfunction number: SFkt for transfer back NC program CF card

SFkt	Function	Remark
8	Transferring the NC program back	Parameter required, see SFkt = 1 Table 9-3 SFkt for data dialogs (Page 154)



T_DATA_M()

Machine: BAZ1X1

OrderNum: 1001

SFkt: 8: NC program on cf-card

Name1: \MPF.DIR

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 5` to the host computer as positive acknowledgment.

R_DATA_H

```
R_DATA_H
OrderNum->1002  Machine->"BAZ1X1"  Host->"H1" SFkt->1
Name1->"\mpf.dir\748651.mpf"
Name2->"\\192.168.5.228\RPCHostDirectory\Putdir\748651.mpf"
datVal->1465379838  LastFile->0  Runtime->0.0142
```

Result

The NC program is stored in the configured host directory.

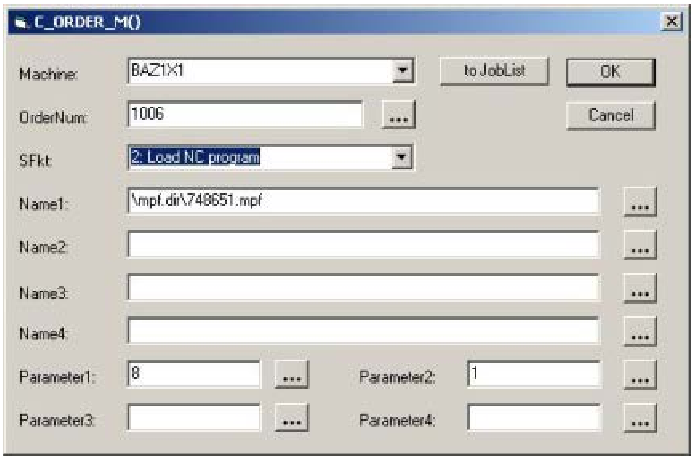
17.2.3 Loading the NC program on the CF card

With the general order function `C_ORDER_M SFkt = 2, Parameter1 = 8` the NC program indicated under "Name1:" is loaded in the NC memory of the PCU 50.

Function

Table 17- 13 Subfunction number: SFkt for load NC program on CF card

SFkt	Function	Remark
2	Loading the NC program on the CF card	Parameter1 = 8 (destination CF card)



Machine: BAZ1X1
OrderNum: 1006
SFkt: 2: Load NC program
Name1: \mpf.dir\748651.mpf
Name2:
Name3:
Name4:
Parameter1: 8
Parameter2: 1
Parameter3:
Parameter4:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

R_REPORT_H

```
R_REPORT_H
OrderNum->1003  Machine->"BAZ1X1"  Host->"H1" Typ->6
Number(0..9) -> 2 0 0 0 0 0 0 0 0 0
Time(0..9)   -> 0 0 0 0 0 0 0 0 0 0
Flag(0..9)   -> C 0 0 0 0 0 0 0 0 0 ResInt1->0
ResInt2->0  ResByte->""  Runtime->0.016
```

Result

`Type :=` 6 `- Identifier of positive acknowledgment`

`Number(0) :=` 2 `- Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-262 - NC program cannot be loaded.

17.2.4 Selecting an NC program on the CF card

Function

Note

Function not possible

This function cannot be implemented technically.

17.2.5 Deselect NC program on the CF card

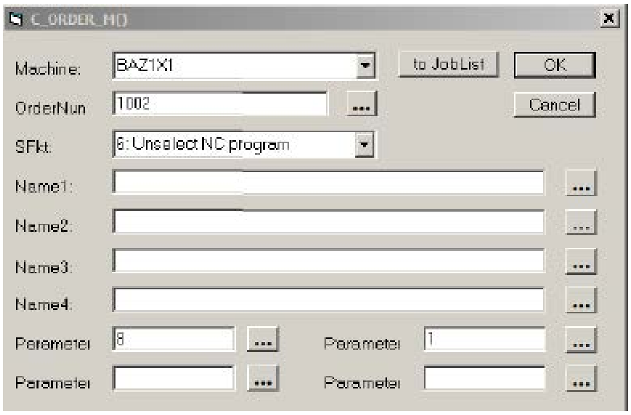
With the general order function `C_ORDER_M SFkt = 6`, `Parameter1 = 8` the selected NC program for the channel indicated in "Parameter2:" is deselected.

Function

Table 17- 14 Subfunction number: SFkt for deselect NC program on the CF card

SFkt	Function	Remark

6	Deselect NC program on the CF card	Parameter1 = 8 (destination CF card)
		Parameter2 = 1 (channel)



Machine: BAZ1X1
OrderNum: 1002
SFkt: 6: Unselect NC program
Name1:
Name2:
Name3:
Name4:
Parameter 8 Parameter 1
Parameter Parameter

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

R_REPORT_H

```
R_REPORT_H
OrderNum->1002  Machine->"BAZ1X1"  Host->"H1" Typ->6
Number(0..9) -> 8 0 0 0 0 0 0 0 0 0
Time(0..9)   -> 0 0 0 0 0 0 0 0 0 0
Flag(0..9)   -> C 0 0 0 0 0 0 0 0 0
ResInt1->0  ResInt2->0  ResByte->""  Runtime->1.0206
```

Result

`Type := 6 - Identifier of positive acknowledgment`

`Number(0) := 8 - Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type=4` to the host computer as negative acknowledgment.

Error number

-266 - NC program cannot be deselected.

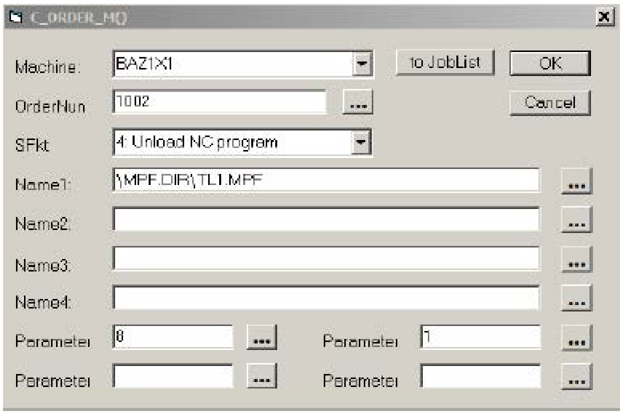
17.2.6 Unloading the NC program from the CF card

With the general order function `C_ORDER_M SFkt = 4`, Parameter1 = 8 the NC program selected under "Name1:" is unloaded from the CF card.

Function

Table 17- 15 Subfunction number: SFkt for unload NC program on the CF card

SFkt	Function	Remark
4	Unloading the NC program from the CF card	Parameter1 = 8 (destination CF card)



Machine: BAZ1X1

OrderNun 1002

SFkt 4: Unload NC program

Name1: \MPF.DIR\TL1.MPF

Name2:

Name3:

Name4:

Parameter 8 Parameter 1

Parameter Parameter

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 6` to the host computer as positive acknowledgment.

R_REPORT_H

```
R_REPORT_H
OrderNum->1002  Machine->"BAZ1X1"  Host->"H1" Typ->6
Number(0..9) -> 4 0 0 0 0 0 0 0 0 0
Time(0..9)   -> 0 0 0 0 0 0 0 0 0 0
Flag(0..9)   -> C 0 0 0 0 0 0 0 0 0
ResInt1->0  ResInt2->0  ResByte->""  Runtime->0.1721
```

Result

`Type := 6 - Identifier of positive acknowledgment`

`Number(0) := 4 - Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 4` to the host computer as negative acknowledgment.

Error number

-264 - NC program cannot be unloaded.

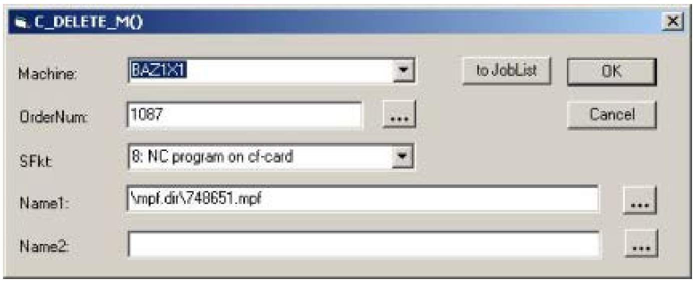
17.2.7 Deleting the NC program from the CF card

With the general order function `C_DELETE_M SFkt = 8`, the NC program indicated under "Name1:" is deleted on the CF card.

Function

Table 17- 16 Subfunction number: SFkt for delete NC program on the CF card

SFkt	Function	Remark
8	Delete NC program	Parameter required, see Sfkt = 1 Table 9-3 SFkt for data dialogs (Page 154)



C_DELETE_M()

Machine: BAZ1X1

OrderNum: 1087

SFkt: 8: NC program on cf-card

Name1: \mpf.dir\748651.mpf

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type = 5` to the host computer as positive acknowledgment.

R_REPORT_H

```
R_REPORT_H
OrderNum->1003  Machine->"BAZ1X1"  Host->"H1" Typ->5
Number(0..9) -> 8 0 0 0 0 0 0 0 0 0
Time(0..9)   -> 0 0 0 0 0 0 0 0 0 0
Flag(0..9)   -> C 0 0 0 0 0 0 0 0 0
ResInt1->0  ResInt2->0  ResByte->""  Runtime->0.1721
```

Result

`Type := `**5**` - Identifier of positive acknowledgment`

`Number(0) := `**8**` - Initiated SFkt`

RPC SINUMERIK sends an RPC call `R_REPORT_H Type=4` to the host computer as negative acknowledgment.

Error number

-250 - NC program cannot be deleted.

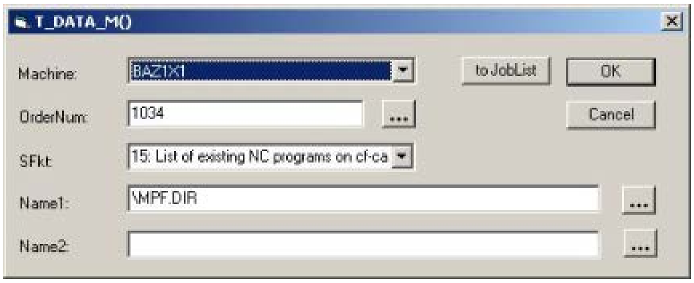
17.2.8 Listing NC program(s) on the CF card

With the general order function `R_DATA_M SFkt = 15`, the NC program directory indicated under "Name1:" is read from the CF card and transferred to the host computer as a file.

Function

Table 17- 17 Subfunction number: SFkt for list NC program on the CF card

SFkt	Function	Remark
15	List NC program(s) on the CF card	Parameters required, see SFkt = 10 Table 9-3 SFkt for data dialogs (Page 154)



Machine: BAZ1X1

OrderNum: 1034

SFkt: 15: List of existing NC programs on cf-ca

Name1: \MPF.DIR

Name2:

After processing is complete, RPC SINUMERIK sends an RPC call `R_REPORT_H Type=15` to the host computer as positive acknowledgment.

R_Data_H

```
R_Data_H
OrderNum->1008  Machine->"BAZ1X1"  Host->"H1" SFkt->15
Name1->"\mpf.dir"
Name2->"\\192.168.5.228\RPCHostDirectory\Putdir\NCPrList.001"
DateVal->0  LastFile->0  Runtime->0.1785
```

Result

SFkt = 15 - Positive acknowledgment identifier

File structure

The file transferred to the host computer has the following structure:

The first row of the file contains the requested directory name.

The following rows list the NC programs contained in the directory. Each row represents one NC program.

Row 1: \MPF.DIR

Row X: TL1.MPF,FN,201,1449237494

Program row structure

Each program row has the following structure:

- Each column is separated by a comma.
- Type and location:
  - F - File
  - D- Directory
  - M - NC program on the PCU/IPC
  - N - NC program on the CF card

Example:

```
NC program name ,type and   ,length ,date and time of last change (UNIX)
                 location
TL1.MPF         ,FN         ,201    ,1449237494
```

Error number

-262 - NC program cannot be loaded.

See also error numbers in the following section: Error numbers (Page 348)

Supplementary information 18

This section describes the use of IP addresses for the configuration of CMI, e.g. the "IP address" field in the "Host" tab of the configuration tool.

The alternative input of computer names is not deliberately prevented by this. However, the user bears the responsibility when using computer names. It must be ensured, for example, that the name resolution is correctly configured in the network.

18.1 Reading out TC_MOP parameters (single tool)

In the configuration program, you can specify that the cutting edge-related tool monitoring data is to be output when a single tool is read out. A single tool is then read out. The `TC_MOP11` parameter holds the target tool life and the `TC_MOP13` parameter holds the target workpiece count. If you open the file read out by RPC, these two target values are contained in the `TC_MOP5` and `TC_MOP6` parameters, however.

These two parameters are intended for the actual and target wear values.

Solution:

Table 18- 1 Monitoring data for each tool cutting edge

Description	Parameter	Software version
P1 = Prewarning limit for tool life in minutes	`$TC_MOP1`	
P2 = Remaining tool life in minutes	`$TC_MOP2`	
P3 = Prewarning limit for workpiece count	`$TC_MOP3`	
P4 = Remaining workpiece count	`$TC_MOP4`	
P5 = Target tool life	`$TC_MOP11`	
P6 = Target workpiece count	`$TC_MOP13`	
P7 = Prewarning limit for wear	`$TC_MOP5`	from SW 5.1
P8 = Remaining wear (actual value)	`$TC_MOP6`	from SW 5.1
P9 = Target wear value	`$TC_MOP15`	from SW 5.1

The following applies: In general, RPC returns a consecutive list of 9 parameters in `TC_MOP`.

There are 9 parameters, but they are called `MOP1-6`, `MOP11`, `MOP13` and `MOP15`.

18.2 Reading out the parameter designation for a single tool

The file that is returned when a single tool is read out contains the `TC_DP26-34` cutting-edge parameters.

These parameters are `$TC_DPV, $TC_DPV3, $TC_DPV4, $TC_DPV5, $TC_DPVN3, $TC_DPVN4, $TC_DPVN5` and `$TC_DPNT`. If you are able to have the complete list of all tools returned, then the parameters are correctly designated.

For reading all tool data (`T_DATA_M SFkt = 20`), the "TO_INI.INI" file is sent from the NC to the host computer. The extended cutting-edge data are correctly represented in this file, since it is created by the NC. `$TC_DPV`, `$TC_DPV3`, etc.

When a single tool is read or reported (`T_DATA_M SFkt = 21-23`), the extended cutting-edge data are sent as `$TC_DP26` to `$TC_DP<n> (`depending on the configuration). The parameter number of the extended cutting-edge data can vary depending on the cutting-edge data configuration. You must determine for your machine which data were set for the `TC_DP26-34` parameters.

18.3 T_DATA_M ignores specified paths or filenames

For the subfunctions of `T_DATA_M`, you have the option of specifying your own name for the data returned by the RPC. For example, when `SFkt = 10` (output of a list of all programs in the NC), you can specify a desired name and path under Name 2 under which the list is to be stored on the host computer. If you enter a name here, it is ignored however. The list is always stored in the form NcPrList.001. A specified path is also not taken into account. Instead, the list is always placed in the Host Put-Directory. The same happens for SFkt 20 (output of a list of all tools). Also affected is `SFkt = 1` (transfer NC program to the control system). You can specify a name under which you want to save the program on the host computer. However, this name is not evaluated. The name that the program has in the NC is always used. Conversely, for `R_DATA_M` (transfer a program to the NC), an alternative name defined by you is applied.

The following applies:

Table 18- 2 Call `T_DATA_M`

SFkt	Description	Parameter (example)
1	Transfer NC program to the host computer	Name1: \mpf.dir\nc1.mpf Name2: Empty

Table 18- 3 Response `R_DATA_H`

Machine	SFkt	Name1	Name2
SL_M1	1	\mpf.dir\nc1.mpf	C:\TMP\HOST\PUT\NC1.MPF

Table 18- 4 Call `T_DATA_M`

SFkt	Description	Parameter (example)
10	Transfer list of existing NC programs to the host computer	Name1: \mpf.dir Name2: Empty

Table 18- 5 Response `R_DATA_H`

Machine	SFkt	Name1	Name2
SL_M1	10	\mpf.dir	C:\TMP\HOST\PUT\NCPRLIST.001

Table 18- 6 Call `T_DATA_M`

SFkt	Description	Parameter (example)
20	Transfer list of existing tools to the host computer	Name1: Empty Name2: Empty

Table 18- 7 Response `R_DATA_H`

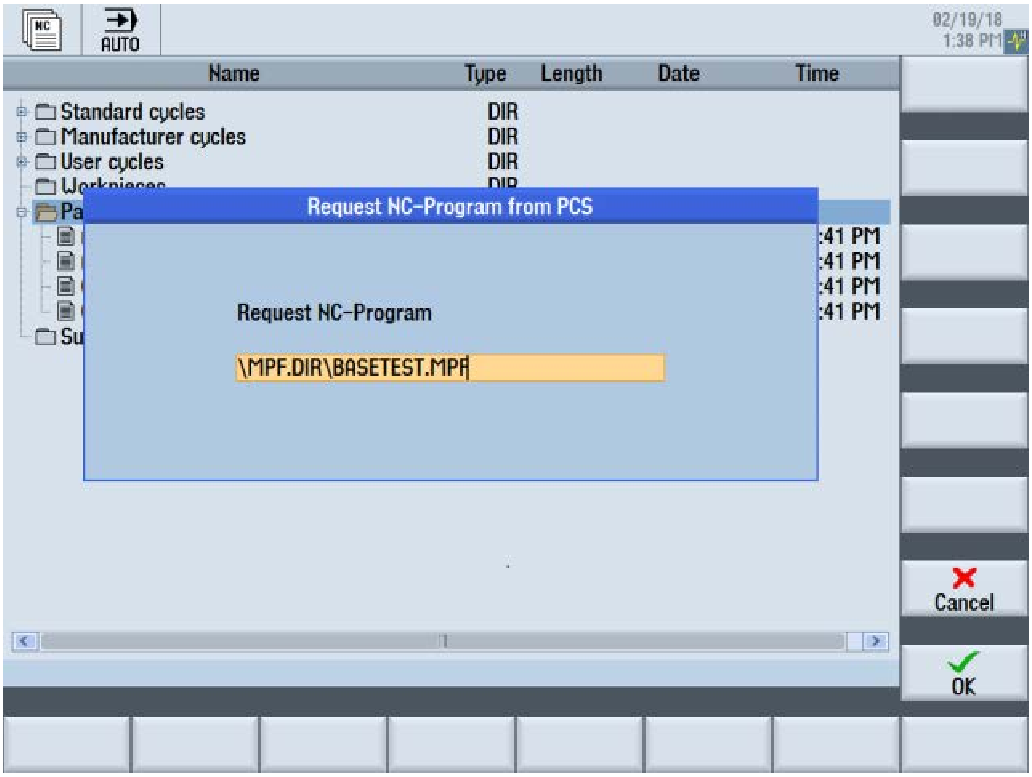
Machine	SFkt	Name1	Name2
SL_M1	20		C:\TMP\HOST\PUT\TOOLDATA.001

18.4 Entering a complete path for workpieces / NC program

Include a path specification for an NC program.

Examples for NC programs and workpiece directories:

```
\MPF.DIR\BIG7.MPF   // permissible
\MPF.DIR\14046.MPF  // permissible
\MPF.DIR\BIG7       // extension can be omitted
\MPF.DIR\14046      // extension can be omitted
BIG7                // impermissible
14046               // impermissible
```



Request NC-Program from PCS

Request NC-Program

\MPF.DIR\BASETEST.MPF

Note

Workpiece directories

The entry is also applicable for workpiece directories.

18.5 Transfer and loading of NC programs

To transfer an NC program from the host computer to the controller, the following steps are necessary in the host computer software:

1. `R_DATA_M SFkt = 1` transfers the NC program to the PCU 50
2. `C_ORDER_M SFkt` = 2 loads the NC program from the PCU 50 into the NC

18.6 Language switchover HMI Operate on industrial PC

It is not possible to switch the default language setting of HMI Operate, installed on an industrial PC to operate Create MyInterface.

Note

Default language

The default language is English.

Appendix A

A.1 Interface Definition Language (IDL)

A.1.1 Interface Definition Language (IDL)

Notes on handling

The installation CD of the "RPCTEST" installation contains the following files:

- SCHOST.IDL
- SCMACH.IDL
- SCHOST.ACF
- SCMACH.ACF

The IDL files describe the function calls along with their parameters.

The ACF files are used to specify whether internal or external binding is desired for the IDL compiler. If the host computer is to communicate with multiple machines, external binding must be used.

The supplied ACF files contain the corresponding instruction for both types of binding. One of them is commented out, however. Ensure that the desired form is active and the undesired form is a comment. The IDL compiler generates the client and server stubs as well as header files from these files.

If Microsoft Windows (>= WINDOWS 95) is used on the host computer, the VC++ development system is sufficient. It contains an IDL compiler and the other files that are necessary for RPC.

Call of Microsoft IDL compiler

MIDL SCHOST.IDL /osf	For RPCs to the host computer, e.g. `R_MACHINE_H`
MIDL SCHMACH.IDL /osf	For RPCs to SINCOM, e.g. `T_MACHINE_M`

The /osf parameter must be specified in order to be DCE-compatible.

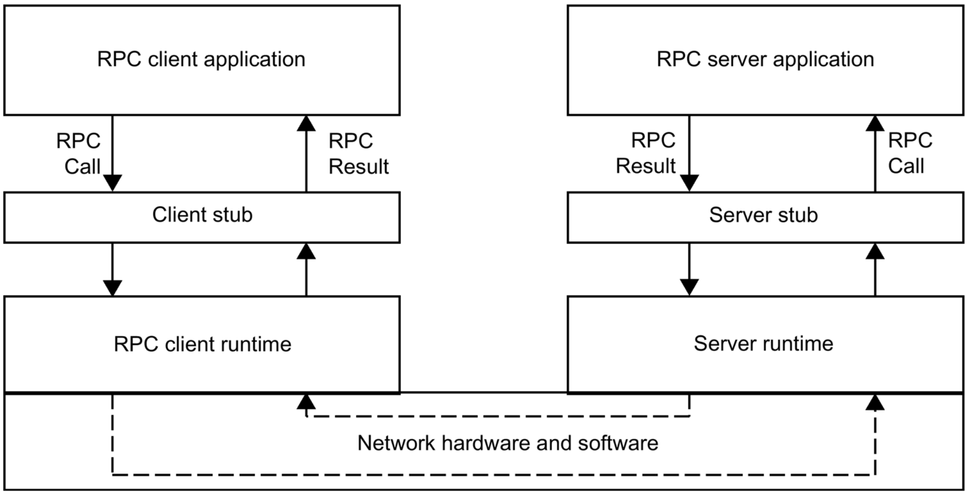
A DCE-RPC development system is required for other operating systems.

Note

No SUN RPC

- Do not use a SUN RPC development system!
- Use a DCE-RPC development system.

The "example" directory of the RPCTEST installation contains the sources of a VC++ 6.0 sample program as a template for an RPC SINUMERIK host computer application as well as a brief description of the most important program sections.



A.1.2 Functions for the host computer - SCHOST.IDL

Function

Table A- 1 SCHOST.IDL

```
[ uuid(d3d7d860-c15a-11d0-a0cb-00a0244ce687),
 version(1.0),
 pointer_default(unique)
]
interface SINCOMHOST
{
const long maxWPCLen = 6;
const long maxMPos = 3;
long R_MACHINE_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in] long MachineMode,
  [in] long MachineStatus,
  [in, string] unsigned char* pszNCProgramm,
  [in] long ClampCubeSide,
  [in] long DockPos[maxMPos],
  [in] long DockPosStatus[maxMPos],
  [in] unsigned char pszWPC[maxMPos][maxWPCLen],
  [in] long WPCStatus[maxMPos],
  [in] long ResInt1,
  [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
const long maxTPos = 2;
long R_TPS_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long MachineMode,
  [in] long MachineStatus, [in] long TpOStatus,
  [in] long DockPos[maxTPos], [in] long DockPosStatus[maxTPos],
  [in] unsigned char pszWPC[maxTPos][maxWPCLen],
  [in] long ResInt1, [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
const long maxAlarms = 10;
long R_REPORT_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long Typ,
  [in] long Number[maxAlarms],
  [in] long Time[maxAlarms],
  [in] char Flag[maxAlarms],
  [in] long ResInt1,
  [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
long R_MESSAGE_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in, string] unsigned char* pszMessage,
  [in] long ResInt1, [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
long T_DATA_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SFkt,
  [in, string] unsigned char* pszName1,
  [in, string] unsigned char* pszName2 );
long R_DATA_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SFkt,
  [in, string] unsigned char* pszName1,
  [in, string] unsigned char* pszName2,
  [in] long Date, [in] long LastFile );
long R_VAR_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long VarMode,
  [in, string] unsigned char* pszVarSet,
  [in, string] unsigned char* pszVarDescr,
  [in, string] unsigned char* pszVarData );
long R_DDEDATA_H([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in, string] unsigned char* pszData );
void Shutdown_H(void);
}
```

A.1.3 Functions for SINUMERIK - SCMACH.IDL

Function

Table A- 2 SCMACH.IDL

```
[ uuid(d6542300-c15a-11d0-a0cb-00a0244ce687),
  version(1.0),
  pointer_default(unique)
]
interface SINCOMMACHINE
{
long T_MACHINE_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum);
long T_TPS_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum);
long T_DATA_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SFkt,
  [in, string] unsigned char* pszName1,
  [in, string] unsigned char* pszName2 );
long T_VAR_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long VarMode,
  [in, string] unsigned char* pszVarSet,
  [in, string] unsigned char* pszVarDescr );
long R_NC4WPC_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in, string] unsigned char* pszWPC,
  [in, string] unsigned char* pszNCProg,
  [in] long Date, [in] long NCPLength,
  [in] long ClampCubeSide,
  [in] long TpFlag,
  [in] long NCExtern,
  [in] long ResInt1,
  [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
long R_REPORT_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in] long Typ,
  [in] long Number,
  [in] long ResInt1,
  [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
long R_MESSAGE_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in, string] unsigned char* pszMessage,
  [in] long ResInt1, [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
long R_DATA_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SFkt,
  [in, string] unsigned char* pszName1,
  [in, string] unsigned char* pszName2,
  [in] long Date, [in] long LastFile );
long R_VAR_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long VarMode,
  [in, string] unsigned char* pszVarSet,
  [in, string] unsigned char* pszVarDescr,
  [in, string] unsigned char* pszVarData );
long R_DDEDATA_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in, string] unsigned char* pszApplication,
  [in, string] unsigned char* pszTopic,
  [in, string] unsigned char* pszItem,
  [in, string] unsigned char* pszData );
long C_DELETE_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SFkt,
  [in, string] unsigned char* pszName1,
  [in, string] unsigned char* pszName2 );
long C_MODE_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long Mode );
long C_SYNCH_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum, [in] long SynchFlag);
long C_TPORDER_M([in, string] unsigned char* pszHost,
  [in, string] unsigned char* pszMachine,
  [in] long OrderNum,
  [in] long SDockPos,
  [in] long DDockPos,
  [in, string] unsigned char* pszWPC,
  [in] long WPCTyp,
  [in] long BufferFlag,
  [in] long Priority,
  [in] long ChainNum,
  [in] long Vehicle,
  [in] long ResInt1,
  [in] long ResInt2,
  [in, string] unsigned char* pszResByte );
void Shutdown_M(void);
}
```

A.2 OEM interface HMI <=> NC/PLC

The following chapter of this appendix is an excerpt from the user instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package, Part 1"; Edition 02.2004.

For reasons of completeness, references to additional Chapters of these user instructions are included in this documentation. As a consequence, you can only find the corresponding information in the user instructions.

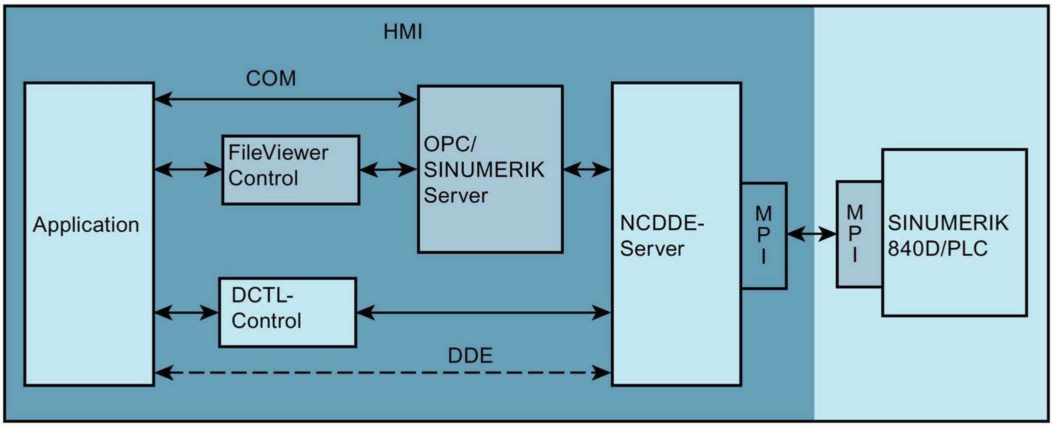
Overview

Three different services are available to the developer for communication:

Variable service	Access to NC, PLC and drive data	
	via	OPC DataAccess
	or	DCTL Control
Domain service	Copying files between the HMI and NC:	
	File access via data management	
	with	IMC File
	and	IADS
	or	FileViewer Control
Program instance service	Start functions in the NC	
	with	IMC command

A.2.1 General

Communication between applications and NC/PLC takes place via the OPC-COM or SINUMERIK-COM interfaces. For compatibility reasons, direct access via the NCDDE server is also still possible for the time being. For new developments, access via the new COM interfaces should always be used.



The NCDDE server can be configured using initialization files. This allows the user to adapt the NCDDE server to his development environment.

- Can testing be done on a controller?
- Is one or more NCUs available whose data is to be accessed?

Note

No time guarantee

As a result of the Windows environment, there is no time guarantee for communication with SINUMERIK 840D. For this reason, real-time tasks cannot be implemented in the HMI. They must be handled in the NCU area using the NC OEM package.

A.2.2 DDE basics

Overview

The Windows operating system provides application developers with the DDE (Dynamic Data Exchange) mechanism as a communication tool between Windows processes.

Main DDE features

DDE characterizes the dynamic data exchange under Windows with the following attributes:

- DDE is communication between Windows applications.
- DDE runs between two processes according to the client-server model.
- One process is a client: it requests data from another program (the server).
- One process is a server: it supplies the data to the other program (the client).
- Connection is established by the client.
- A program can be both a client and a server.
- The Windows protocol is used.

DDE connection establishment

In order to establish the connection to a DDE server, the developer of the client must know the following:

• Link Server	Name of the DDE server
• Link Topic	Topic
• Link Item	Data that is to be accessed
• Link Mode	Connection mode

DDE Link Mode

• Request	The client requests the data only once.
• Warmlink	The server informs the client if the data has changed. The client can then fetch the data.
• Hotlink	If the data has changed, the server automatically sends the current value of the data to the client.
• Poke	The client instructs the server to write data.
• Execute	The client instructs the server to execute a command.

A.2.3 NCDDE server configuration

A.2.3.1 Initialization file "MMC.INI"

Description

The NCDDE server is initialized using the "[GLOBAL]" section in the "MMC.INI" file.

This file is located in the "\MMC2" directory of the OEM system. Link Server and Link Topic that the local NCDDE server is to establish a connection to is defined here.

You can find explanations for the terms "Link Server" and "Link Topic" in Chapter: DDE basics (Page 285).

The NCDDE server can take on four basic configurations depending on the installation scope:

- Establishment of connection to an NC
- Establishing a connection to one or several NCs

  For information on M:N functionality, see Chapter: Connection to several NCs (Page 290).
- Local operation on a PC

  Gives the developer the option of testing his application locally on a PC without a connection to an NC. In this case, the NCDDE server outputs substitute values that are defined with the "NEW" command and that can be changed with the "ANIMATE" command in such a way that it creates the impression of an active NC.

  For information on the "NEW" and "ANIMATE" commands, see Chapter: Further commands of the NCDDE server (Page 318)
- Local operation on a PC with NC simulator.

  Gives the developer the option of testing his application locally on a PC without a connection to an NC. The NC simulator enables simulation of a NC-like behavior for the HMI.

NcddeServiceName

DDC Link service name of the DDE server It is preset to "ncdde".

Note

Examples

All examples in section OEM interface HMI <=> NC/PLC (Page 284) are based on "NcddeServiceName= ncdde". If the name is different, these examples must be modified accordingly. Otherwise, they will not work.

Ncdde-MachineName

The NCU name for the standard applications is entered here. If "MachineSwitch" is entered here, switchover between individual NCUs is possible.

For information on M:N functionality, see Chapter: Connection to several NCs (Page 290).

NcddeDefault-MachineName

The initialization for the M:N functionality is set with this, i.e. the connection to this NCU is established when the HMI powers up.

Ncdde-MachineNames

The names of the NCUs to which a connection can be established are entered here. For each NCU name entered here, the "MMC.INI" file must contain a section with this name.

NcddeStartupFile

NSK file that is to be loaded when the NCDDE server starts.

For information on the NSK file, see Chapter "Command files of the NCDDE server (Page 289)".

NcddeMachineNamesAdd1

Identification of an installed NC simulation.

If no simulation software has been installed, this entry is irrelevant.

The following example shows the parameter assignment of the "MMC.INI" file based on installation on a PC without NC and without simulation software.

Table A- 3 Example: Extract from the "MMC.INI" file

```
[GLOBAL]
; for using M:N function set NcddeMachineName=MachineSwitch
; for working without NC set NcddeMachineName=local
; for working with SIMNC set NcddeMachineName=SIM1
; for connecting to a NC set NcddeMachineName=NCU840D
NcddeMachineName=local
; for using M:N function set NcddeDefaultMachineName=net:NCU_1
; for working without NC set NcddeDefaultMachineName=local
; for working with SIMNC set NcddeDefaultMachineName=SIM1
; for connecting to a NC set NcddeDefaultMachineName=NCU840D
NcddeDefaultMachineName=local
; for using M:N function set NcddeMachineNames=net,NCU840D
; for working without NC set NcddeMachineNames=
; for working with SIMNC set NcddeMachineNames=SIM1
; for connecting to a NC set NcddeMachineNames=NCU840D
NcddeMachineNames=
; for using M:N function set NcddeStartupFile=ncdde5.nsk
; for working without NC set NcddeStartupFile=ncdde202.nsk
; for working with SIMNC set NcddeStartupFile=sim1dde5.nsk
; for connecting to a NC set NcddeStartupFile=ncdde5.nsk
NcddeStartupFile=ncdde202.nsk
```

Namespaces

The NCDDE server does not differentiate variables into "namespaces" in LOCAL operation.

"Namespace" is understood to mean a differentiation by TOPIC.
Thus, if a variable has been created for TOPIC LOCAL and the same variable has been created for TOPIC Sim0, these variables are not differentiated by the NCDDE. As a result of this, for example, the current block display no longer works in simulation if you ever change to a screen with current block display after MACHINE because local variables are created there from the program and these local variables redefine the "simulation variables".

A.2.3.2 Command files of the NCDDE server

NSK files

The command files with extension ".NSK" contain, for example, the link Items that the NCDDE connections refer to. These files can contain commands that are described in Chapter: Further commands of the NCDDE server (Page 318). The data Link Items that can be accessed are described in these files. Other .nsk files can also be included using CALL. This approach enables structuring. The following example shows a link Item (LastError) and the structuring of the global variables for HMI with CALL instructions.

Note

Loading of user-related NSK files

The user has the option of loading his own NSK files with CALL instructions. The NSK files can be created with a MAP function; see Chapter: MAP functions between domains (Page 312)).

Table A- 4 Example: File NCDDE311.NSK in directory "\mmc2"

```
REM NSK ROOT FOR 840D
===========================================================
REM
REM WRITE-ACCESS FOR NC-BUSADDRESS
LINK(”/Nck/Nck/busAddress”,200,”7 31 0 0 E0# /NC 1 0 11”,10)
LINK(”/Nck/Nck/busState”,300,””,0);
REM
REM ACCESS TO CONNECTION ERROR STATE
LINK(”LastError”,1,””,0);
REM
REM IMPORT 840D BASIC NC VARIABLES
CALL(nc311.nsk)
REM
REM IMPORT 840D BASIC PLC VARIABLES
CALL(plc311.nsk)
REM
REM IMPORT ADDITIONAL LINK VARIABLES
CALL(add311.nsk)
REM
REM IMPORT COMIC STARTS
CALL(comic.nsk)
REM
```

A.2.3.3 Connection to several NCs

M:N functionality

This functionality allows the connection of multiple HMIs with multiple NCUs. For example, you can access data of two NCUs from an HMI. The "NETNAMES.INI" file is interpreted for this basic configuration.

Connection section

The "[conn MMC_1]" section specifies the partners that the HMI can establish communication with.

Network parameters

The transmission rate is defined in the "[param network]" section:

BTSS	1.5 Mbits
MPI	187.5 Kbits

Bus nodes

In the "[param NCU_n]" sections, the bus addresses of the NC and the PLC as well as the name of the NCU are specified.

The NCU must be addressed from the HMI using this name. A description must be available for each NCU.

Table A- 5 Example: The "NETNAMES.INI" file

```
; owner TECHNISCHE Referenz auf Busbeschreibung
; Rechnerspezifisch
[own]
owner= MMC_1
; Description of possible connections
[conn MMC_1]
conn_1= NCU_1
conn_2= NCU_2
; Description of significant net-parameters
; btss =1,5MBit
; mpi =187,5 KBit
[param network]
bus= btss
; Busadressen für alle Busteilenehmer
[param MMC_1]
mmc_address= 1
[param NCU_1]
nck_address= 10
plc_address= 10
name=Standard_Machine
[param NCU_2]
nck_address= 11
plc_address= 11
name=Test_Maschine
```

Application

For OEM applications to behave correctly in an M:N configuration, the following must be observed:

- For each communication with the NC, "machineswitch" should be used exclusively as the Link Topic. This ensures that the application always communicates with the NC that was selected as part of an M:N or NC switchover. If, on the contrary, the name of a particular NCU is specified, e.g. NCU1, when a communication connection is set up, this will not be taken into consideration as part of an M:N switchover. This means that this connection is retained after switchover from NCU1 to another NC on the operator panel. Such "fixed" or "static" connections to a particular NC may only be set up by an application when the operator panel on which the application runs is configured as an M:N server.
- An NC switchover is represented for the OEM application like an NC reset or communication failure. The application should also respond the same as after an NC reset and communication failure. You recognize an NC reset by the setup of a Hotlink to the BTSS variable "/bag/state/opmode". In the case of an NC reset, the hotlink returns the value "#".

- To prevent an M:N or NC switchover from interrupting or terminating communications running in the background, such as file transfers, the NC switchover should be disabled before starting such operations and enabled again after they are finished. Functions "LockCurrentNCU/LockChanMenu" and/or "UnlockCurrentNCU/UnlockChanMenu" are available. While "LockCurrentNCU" disables only the switchover to another NC, channel switchovers on the currently selected NC are still permitted. "LockChanMenu" also prevents the channel switchover.
- If certain services or variables are available only on a particular NCU, before accessing these services or variables a check must be made to determine if there is a connection to the relevant NCU. The NCU to which there is currently a connection can be determined by reading out the "machineswitch" variable (KinkItem).

A.2.4 Structure of a DDE connection

Overview

This section explains the basic establishment of a DDE connection to the NCDDE server with Visual Basic and with Visual C++.

Note

Examples

In the following examples, only the standard Visual Basic Control "LABEL" is used for the DDE communication. However, an OEM application should use the Siemens Control "DCTL" for DDE communication; see Chapter: DCTL.OCX file (Page 323)).

The following requirements must be met for the examples to work:

Development environment:

- Recommendation: MS Visual Basic 4. 0_16
- To enable testing the examples on the SINUMERIK 840D directly from the PC, an MPI connection is needed and the NCDDE server must be configured for NC operation.

  If the NCDDE server is operated without SINUMERIK 840D, not all data accesses are possible.
- The NCDDE server (C:\MMC2\NCDDE.EXE) must be started (e.g. using Explorer or the Start menu)

A.2.4.1 Establishment of a DDE connection with Visual Basic

For standard Visual Basic controls that support DDE, such as

- Label
- Text box
- Picture

Link Service and Link Topic are combined in the "Link Topic" property (attribute). They are separated by the pipe character "|" (e.g. Link Topic "ncdde|local").

One-time import of variables

The following example reads the actual value of the first axis of the first channel in the tool coordinate system once.

For the following example to work, the NCDDE server must be configured for local operation and "NcddeServiceName = ncdde" is required. This means that the NC is not accessed here. "LinkMode" must have the value "2" to read once only.

Note

Link Mode

"LinkMode" must have the value 2 to read once only. The value is then requested from the first channel with the "LinkRequest" method.

Table A- 6 Example: One-time reading of variables

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|local”
       Label1.LinkItem=”/Channel/GeometricAxis/actToolBase-
       Pos[u1,1]”
       Label1.LinkMode = 2
       Label1.LinkRequest
End Su
```

Note

Channel designation

If the channel designation "u1" is not specified, the first channel is automatically accessed.

Update following a change

The following example automatically updates (hotlink) the actual value of the third axis of the second channel in the machine coordinate system in "Label1", i.e. the actual value of this axis is displayed.

Note

Hotlinks

For hotlinks, "LinkMode" must have the value "1".

Table A- 7 Example: Update following a change

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem=”/Channel/MachineAxis/actToolBasePos[u2,3]”
       Label1.LinkMode = 1 ’Hotlink
End Sub
```

Writing NC data

In this example, the client writes the value "4" in the R-parameter R[1] of the first channel.

Note

Writing of data

When data is written (Poke), "LinkMode" must have the value "2". The value is written with "LinkPoke".

Table A- 8 Example: Writing NC data

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/Channel/Parameter/R[1]”
       Label1.LinkMode = 2 ’Manual
       Label1.Caption = ”4”
       Label1.LinkPoke
End Sub
```

Writing PLC data

In this example, the client writes the value "250" in the memory byte 5 of the PLC.

Table A- 9 Example: Writing PLC data

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/PLC/Memory/Byte[5]”
       Label1.LinkMode = 2 ’Manual
       Label1.Caption = ”250”
       Label1.LinkPoke
End Sub
```

Executing a command

To execute the command; see also Chapter: File transfer HMI <=> NC/PLC (Page 304), MAP functions between domains (Page 312), PI services (Page 316), Further commands of the NCDDE server (Page 318) .

The following example starts a file transfer of the "test.mpf" file from the HMI to the NC.

Note

Executing a command

To execute commands (Execute), the "Link Mode" must have a value of "2". With "LinkExecute", the command is executed.

Table A- 10 Example: Executing a command

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkMode = 2
       Label1.LinkExecute ”COPY_TO_NC(””C:\NC\test.mpf””,
       /NC/_N_MPF_DIR/N_TEST_MPF, trans)”
End Sub
```

A.2.4.2 Establishment of a DDE connection with Visual C/C++

Overview

The full functionality of the DDE interface can be used from C/C++. In particular, it is possible to direct asynchronous calls to the DDE interface. Under use of the OEM Visual Basic controls such as DCTL, this is also possible in Visual Basic.

Note

Recommendation

DDE with Visual C/C++ is only recommended for OEM users who are experienced in C programming in Windows and need only conditional integration or no integration in the sequential control system of the OEM package.

DDE access with C/C++

This example shows how a hotlink (Advise) connection is established (with acknowledgment communication) between:

- C/C++ program
- Variable "/Channel/GeometricAxis/toolBaseDistToGo[1]"
- NcddeServiceName = ncdde
- NcddeMachineName = local

The data changes of the variables are received in an XTYP_ADVDATA transaction with a callback routine registered in DDEML.

Table A- 11 Example: Hotlink to C level

```
DWORD            idInst; // created with DdeInitialize
HSZ              hszService, hszTopic, hszItem; // String Handles
HCONV            hConv; // Conversation Handle
HszService       = DdeCreateStringHandle ( idInst , "ncdde" , NULL );
hszTopic         = DdeCreateStringHandle ( idInst , "local" , NULL );
hszItem          = DdeCreateStringHandle ( idInst ,
                   Ä"/Channel/GeometricAxis/toolBaseDistToGo[1]" , NULL );
hConv            = DdeConnect(idInst,hszService,hszTopic,NULL);
                      // Connection establishment to server
                      // Hotlink follows
if ( DdeClientTransaction ( (LPBYTE)NULL , 0 , hConv , hszItem ,
                 ÄCF_TEXT ,XTYP_ADVSTART|XTYP_ACKREQ , 1000 , NULL )
                 Ä==TRUE) { } // Hotlink establishment successful
```

A.2.4.3 Structure of a DDE connection from MS Excel

Overview

You can establish Advise (Hotlink) connections to the variable interface of the NCDDE server based on cell formulas in Excel.

Syntax in an EXCEL cell

Within a cell

= NcddeServiceName|NcddeMachineName!Variablen

Displaying a PLC bit in Excel

The following example shows an Advise (Hotlink) connection between a cell in Excel (German version) and bit 3 in byte 9 of data block 100.

The variable name is: "/Plc/DataBlock/Bit[c100,9.3]" of an "ncu840D" machine connected via the NCDDE server.

Table A- 12 Example: Displaying a PLC bit in MS Excel

	A			A
1	=ncdde|ncu840D!'/Plc/DataBlock/Bit[c100,9.3]'		1	1

The cell formula is shown on the left and the resulting, constantly up-to-date display is shown on the right.

A.2.5 Variable service

Overview

Variables of the NC are accessed using OPC data or the DCTLControl. For compatibility reasons, variables can still be accessed using DDE. The variable services of the NCDDE server enables two types of data access:

- Single access
- Array access

The variables can also be characterized in the Link Item by an additional data format and possibly also by an array area. This enables you to request the data from the NCDDE server in such a way that a conversion is usually no longer necessary.

Note

Further information

For a complete description of the variables that can be accessed, refer to Chapter: Online help for variables (Page 344).

Formats of the NCDDE variables

Formatting instructions for NCDDE variables are inserted at the end of the Link Item. The internal data preparation allows the formatting of fixed-point values, floating-point values and texts. An extended 'printf' format of the C language is used for describing the formatting. The syntax of an NCDDE format specification is:

Format:	""	<Params>	<‘printf’ format>
Params	'!' 'b'	<Params>	// Conversion to a bit string
	'!' 'd'	<Params>	// d for double as 64-bit floating
	'!' 'l'	<Params>	// l for long as 32-bit fixed-point //value
	'!' 't'	<Params>	// t for text as string
	'!' '#'	<Params>	// # , Index of variable //access as 32-bit fixed-point value

For additional information on the data type of the corresponding DDE variables, see Chapter: Applications of DCTL.OCX (Page 329) .

Note

Display of incorrect data

If the types of the data selection and the read-out variable value do not match, automatic adaptation of the data format does not occur. That is, incorrect data is displayed.

Formatting of a numerical value

Here, the actual value of the second axis is read from the NC and displayed with a maximum of 11 places before the decimal point and with 3 places after the decimal point. Without a formatting instruction, only 3 places after the decimal point are displayed.

Example: Formatting of a maximum of 11 decimal places before the decimal point and of 3 fixed places after the decimal point

```
Sub Form_Load ( )
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/Channel/MachineAxis/actToolBasePos[2]
                       Ä(””!d%11.3lf””)”
       Label1.LinkMode = 2 ’Manual
       Label1.LinkRequest
End Sub
```

Conversion to hexadecimal number

In this example, the memory byte 5 is read and then displayed as a two-digit hexadecimal number with leading zeros.

Example: Conversion to hexadecimal number

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/PLC/Memory/Byte[5] (””!l%02lx””)”
       Label1.LinkMode = 2 ’Manual
       Label1.LinkRequest
End Sub
```

Conversion to bit string

In this example, bit memory–byte 5 is read, and then output as 32-bit string.

Example: Conversion to bit string

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/PLC/Memory/Word[5] (””!b%16.16s””)”
       Label1.LinkMode = 2 ’Manual
       Label1.LinkRequest
End Sub
```

Result: 10101010101010101

Reading a string from the PLC

In this example, 10 bytes are read starting from byte 20 from data block 81 and then output as a zero-terminated string.

Example of reading a string from the PLC

```
Sub Form_Load ()
       Label1.LinkTopic = ”ncdde|ncu840d”
       Label1.LinkItem = ”/PLC/DataBlock/Byte[c81,20,#10] (””!l%lc””)”
       Label1.LinkMode = 2 ’Manual
       Label1.LinkRequest
End Sub
```

Result: e.g. Hallo

A.2.5.1 Single access

If you work within the sequential control system, then it is advantageous to use global variable "g_chNCDDEServiceName" as Link Topic.

It always contains the NCDDEServiceName entered in the "MMC.INI" file and the NcddeMachineName separated by a pipe character ("|").

Additional information about the sequential control system can be found in Chapter 7.

Single access to three variables

Reading of the names of the first three machine axes

Table A- 13 Example: Single access to three variables

```
Sub Form_Load
       achsname(0).LinkTopic = g_chNCDDEServiceName
       achsname(0).LinkItem = ”/Channel/MachineAxis/name[1]”
       achsname(0).LinkMode = 2 achsname(0).LinkRequest
       achsname(1).LinkTopic = g_chNCDDEServiceName
       achsname(1).LinkItem = ”/Channel/MachineAxis/name[2]”
       achsname(1).LinkMode = 2
       achsname(1).LinkRequest
       achsname(2).LinkTopic = g_chNCDDEServiceName
       achsname(2).LinkItem = ”/Channel/MachineAxis/name[3]”
       achsname(2).LinkMode = 2
       achsname(2).LinkRequest
End Sub
```

PLC bit access

You can use the following Link Item to access bit 4 in input byte 2: /Plc/Input/Bit[2.4].

PLC byte access

You can use the following Link Item to access output byte 4 /Plc/Output/Byte[4].

PLC word access

The following Link Item can be used to access bit memory word 4. /Plc/Memory/Word[8]

Additional access operations are described in Chapter 12.1.5.

A.2.5.2 Array access to data

Application

Always use array access if multiple data is required from one area. This reduces the load on the NCDDE server, and improves the performance of your own applications. The Table 19-13 Example: Single access to three variables (Page 300) is a poor example of this.

Note

Advantages of array access

Array accesses accelerate data access and improve the speed of the overall system because the communication resources required are considerably reduced.

Syntax

Notation of array areas: Variablenname[c, u, StartIndex, [EndIndex]]

Parameter

Table A- 14 Parameters for data accesses

Name	Description
Variable name	Name of the NC/PLC variables; see Chapter 11: User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".
c	Column index for access to NC variables; see Chapter: 11 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1". c stands for column and applies only to multidimensional arrays. Identifies the data block that is to be accessed when accessing PLC data blocks.
u	Unit index only for NC variable (e.g. channel) u stands for unit
StartIndex	Index of the variable to be read. For array accesses this is the first of the values to be read.
EndIndex (optional)	Only for array accesses: Specifies the number of values to be read.

Array access to axis names

This example reads the first three axis names from the NC. The result is a string that contains these axis names, e.g. X1,Y1,Z1 in the form "X1Y1Z1". With the Visual Basic functions "Trim$" and "Mid$", the individual axis names are then extracted from the result string.

Table A- 15 Example: Array access to axis names

```
m_a_namen.LinkTopic = g_chNCDDEServiceName
m_a_namen.LinkItem = "/Channel/MachineAxis/name[u1,1,3]"m_a_namen.LinkMode
= 2
m_a_namen.LinkRequest
'Extract individual values from the value array
achsname1.Caption = Trim$(Mid$(m_a_namen.Caption,1,2))
achsname2.Caption = Trim$(Mid$(m_a_namen.Caption,4,2))
achsname3.Caption = Trim$(Mid$(m_a_namen.Caption,7,2))
```

Array access to axis names

Reading out of axis names of two axes of the second channel, beginning with axis 3. The names of axes three and four are read. Except for the following line, it corresponds toTable 19-13 Example: Single access to three variables (Page 300).

Table A- 16 Example: Array access to axis name

```
...
LinkItem = ”/channel/machineaxis/name[u2,3,4]”
...
```

Array access to PLC data

This example reads bytes 2 to 4 of DB 8 from the PLC: These are bytes 2, 3, 4 as two-digit hexadecimal numbers. The individual bytes are then separated with the Visual Basic functions "Trim$" and "Mid$".

Table A- 17 Example: Array access to PLC data

```
Label1.LinkTopic= ”ncdde|ncu840d”
Label1.LinkItem = ”/PLC/Datablock/Byte[c8,2,4](””!l%02lx””)”
Label1.LinkMode = 1 ’hotlink
'Extracting individual hexadecimal stings from the byte string
byte_1 = Trim$(Mid$(Label1.Caption,1,2))
byte_2 = Trim$(Mid$(Label1.Caption,3,2))
byte_3 = Trim$(Mid$(Label1.Caption,5,2))
```

PLC access with specification of number

This example reads out five words of DB 8 starting from word 2 from the PLC as a four-digit hexadecimal number. The individual words are separated by "_".

Table A- 18 Example: Array access to PLC data

```
Label1.LinkItem = ”/PLC/Datablock/Word[c8,2,#5](””!l%04lx_””)”
```

Array access to R parameters

This example writes the three R parameters R3, R4 and R5 with the values:

- R3 = 2.2
- R4 = 3.5
- R5 = 4.9

Table A- 19 Example: Array access to R parameters

```
Label1.LinkTopic = ”ncdde|ncu840d”
Label1.LinkItem = ”/CHANNEL/PARAMETER/R[U1,3,5]”
Label1.LinkMode = 2 ’Manual
Label1.Caption = ”:2.2:3.5:4.9”
Label1.LinkPoke
```

A.2.6 File transfer services (domain services)

Overview

The file transfer services are used for data transfer between the HMI and NC/PLC areas (domains). IMC File, IADSl and IMC domain are available for the transfer between HMI and NC/PLC. For compatibility reasons, data can still be transferred using DDE. A total of five commands are available for this, as shown in the following table. They operate as a background activity. As of SW version 3.3, extended copying functions are available between the individual areas. These are especially well suited for editing of programs in the NC. A description of the new functions can be found in Chapter 9.6.2.

Table A- 20 Parameters for data accesses

Name	Description
COPY_FROM_NC	Transfer from the NC to the HMI
COPY_FROM_NC_BINA RY	Transfer from the NC (PLC) to the HMI
COPY_TO_NC	Transfer from HMI to the NC
COPY_TO_NC_BINARY	Transfer from HMI to the NC (PLC)
MAP_ACC_NC	Load ACC files from NC kernel and preparation for DDE interface

The status of a data transfer can be tracked using a status variable.

A.2.6.1 File transfer HMI <=> NC/PLC

Description

These functions enable the transfer of data/files between the HMI and NC/PLC.

Application

These functions are suited for transferring part programs and workpieces to the NC and of SIMATIC S7 and C programs to the PLC. Functions without the "BINARY" extension can be used to transfer files such as part programs to the NC. The NCDDE server supplements the data to be transferred by adding a block header containing the block size and block date as well as the area path.

Note

Application

- Used for transferring data to the NC.
- Cannot be used for transferring data to the PLC because an NC block header is always added to the data stream.

BINARY functions

Functions with the "BINARY" extension can be used to transfer files such as part programs to the NC. The NCDDE server transfers the data without an NC block header.

Note

Application

- Can be used for transferring data to the PLC and NC.
- PLC blocks are always transferred to the passive file system of the PLC. But they are not yet active at that point. The passive blocks must then be activated.

Syntax

The copy functions are written as a string in the following form:

COPY_FROM_NC (WinFile,NcFile,TransferState)

COPY_TO_NC (WinFile,NcFile,TransferState)

COPY_FROM_NC_BINARY (WinFile,NcFile,TransferState)

COPY_TO_NC_BINARY (WinFile,NcFile,TransferState)

Parameters

Table A- 21 Parameters for data accesses

Name	Description
WinFile	Source or destination of information in the HMI area
NcFile	File name for the NC/PLC environment
TransferState	Variable that identifies connection status

Parameter WinFile

Describes the source or the destination of the information on the HMI page. The first character indicates the type of information. This parameter is the default file name in the Windows environment. It should always include the drive designation, path name and file name.

Example: C:\NC\test.MPF

Piping with the WinFile parameter

If the first character of WinFile is the @ character, the parameter is interpreted as a pipe name. In combination with COPY_TO_NC, the "Copy using pipes" service can be executed.

Note

Application

Suitable for writing and reading blocks up to a size of 500 bytes. The NCDDE server rejects larger blocks.

During the transfer to the NC/PLC (download), DDE Pokes fill the pipeline and thus ensure direct transfer to the NC/PLC. An empty Poke indicates the end of the transfer. During the transfer from NC/PLC (upload), DDE Requests empty the pipeline that is filled for the active transfer. If a Request delivers empty data, this indicates the end of the transfer.

Shared memory access with the WinFile parameter

If WinFile begins with a # character followed by a hexadecimal number, this is interpreted as Windows shared memory that is assigned to the global heap. The memory allocated with the Windows function "Global/Alloc" must be initialized with the following structure. The user data follows after this header. The following example shows the use from Visual Basic.

Table A- 22 Example: Shared memory access with WinFile

`struct NCDDE_DOMAINMAP_HEADER`	
`{unsigned short handle;`	// handle of buffer (HGLOBAL) (is preset // by the client)
`unsigned short header_size;`	// length of header (is preset by the client)
`unsigned long shared_size;`	// usable length of data area (is preset // by the client)

`unsigned long fill_count;`	// number of valid bytes in data area (is preset
	// by the client when downloading and set by
	// the server when uploading)

`unsigned long state;`	// corresponds to the transfer status variable of the
	// transfer command
	// < 100: transfer is active, "state" indicates what percentage
	// of the transfer has been completed
	// ==100: transfer completed successfully
	// > 100: transfer was canceled with error,
	// "state" indicates the ncdde error code
	// (is set by the server)

`unsigned long file_mod_time;`	// file modification time value:
	// 0 means current time (preset by the
	// client when downloading and set by the server when uploading)

`unsigned long server_private;`	// server-specific data (set by the server)
`unsigned long client_private;`	// client-specific data (set by the client)

`unsigned long magic;`	// signature for an additional type test, value always
	// NCDDE_MAGIC = 0xF6F7F8F9 (preset by the
	// client)

`};`	

Parameter NcFile

The "NcFile" parameter is the file name for the NC/PLC environment. It is composed of two parts: a configurable path name that is needed for addressing the relevant NC and a domain path of the NC environment. Domains in the NC are addressed via the NCDDE server using an NC file name.

/NC	PLC or NC area specification
/_N_MPF_DIR	Path specification for NC
/_N_WS03_MPF	File name

Parameter TransferState

The "TransferState" parameter is the name of an NCDDE local variable (variable type: Fixed) that is to be used for returning the status of the transfer running in the background. This variable is also generated by the NCDDE server, if necessary. The TransferState variable identifies the transfer status:

Table A- 23 Identification of status

Status of transfer	Value	Meaning
Transfer is being started	0	The Open protocol (opening of the file) is running with the CNC.
Transfer is running	1 ... 98	Transfer is running. The value indicates the percentage of the file that has already been transferred (see note).
Transfer is being ended	99	The Close protocol (closing of file) is running with the CNC.
Transfer successful	100	The order has been completed without error.
Transfer stopped with error number	>100	Transfer stopped. TransferState contains the reported error code. This corresponds to the value of the LastError variable; see Chapter: 11.7 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".

The range of values has been chosen so that values ≤ 100 indicate normal conditions and all other values indicate error conditions.

Note

Variable value

When the variable has a value of 1 ... 99, it cannot be used for another file transfer.

Stopping a data transfer

In order to stop a running data transfer, the transfer variable must be written with a valid error value. That is, each byte of the transfer variable defined as "LONG" (4 bytes) must have a value not equal to 0.

Example of a valid error value: 16909060

Visualization

For visualization of the transfer status, the TransferState variable can be used, e.g. in a bar display, or similar via an Advise/Hotlink connection.

Note

Transfer in BINARY mode/with pipes

For a transfer in the BINARY mode and data transfer with pipes, no information on the block size is available. Therefore, information regarding the current percentage of transferred data cannot be supplied to the Transfer State variable: It then stays constant at 50%.

For very short files, the display may jump abruptly from 1 to 99. This is a basic problem with Hotlinks, however, due to the fact that the client/application does not retrieve data fast enough from the NCDDE server.

Upload of a part program

The following example copies the part program "BSP.MPF" to the "test.mpf" file in the "C:\NC" directory. The "test.mpf" file is newly created. The "BSP.MPF" part program must be present in the NC.

Table A- 24 Example: Upload of a part program

```
Sub Form_Load ()
     Label1.LinkTopic = ”NCDDE|ncu840d”
     Label1.LinkMode = 2
     Label1.LinkExecute ”COPY_FROM_NC (C:\nc\test.mpf,
                        Ä/NC/_N_MPF_DIR/_N_BSP_MPF,trans)”
End Sub
```

Download of a part program

The following example copies a file named "test.mpf" from the "C:\NC" directory to the NC in the "_N_MPF_DIR" directory. The part program in the NC is named "BSP.MPF".

Table A- 25 Example: Download of a part program

```
Sub Form_Load ()
     Label1.LinkTopic = ”NCDDE|ncu840d”
     Label1.LinkMode = 2
     Label1.LinkExecute ”COPY_TO_NC(C:\NC\test.MPF,
                        Ä/NC/_N_MPF_DIR/_N_BSP_MPF,trans)”
End Sub
```

Download of a part program with piping

The following example shows the use of the pipe mechanism:

The PIPE1.MPF file is created in the NC and the NC block "G01 F11111 X5555" is written to it.

Table A- 26 Example: Download of a part program with piping

```
Sub Form_Load ()
     ’Start pipe
     Label1.LinkTopic = "NCDDE|ncu840d"
     Label1.LinkMode = 2
     Label1.LinkExecute "COPY_TO_NC(@pipe,
                        Ä/NC/_N_MPF_DIR/_N_PIPE1_MPF,trans)"
     ’Write pipe
     Label2.LinkTopic = "NCDDE|NCU840D"
     Label2.LinkMode = 2
     Label2.LinkItem = "@pipe"
     Label2.Caption = "G01 F11111 X5555"
     Label2.LinkPoke
     ’End pipe
     Label2.Caption = ""
     Label2.LinkPoke
End Sub
```

Download of a SIMATIC S7 block to the PLC

Transferring the "OB1.PLC" block to the PLC's passive file system

Note

PLC blocks

PLC blocks are always transferred to the passive file system of the PLC. But they are not yet active at that point. The passive blocks must then be activated; see: Table 19-37 Example: Activating OB 1 (Page 317).

Table A- 27 Example: Download of an S7 block to the PLC

```
 Label1.LinkItem = "ncdde|ncu840d"
 Label1.LinkMode = 2
  Label1.LinkExecute "COPY_TO_NC_BINARY(C:\TMP\OB1.PLC,
                     Ä/PLC/_0800001P, trans)"
```

A.2.6.2 Extended data transfer HMI <=> NC/PLC

Description

These functions allow data to be transferred between the NC/PLC and HMI.

Application

These functions are especially well-suited for transferring individual blocks and program sections and for editing of part programs on the NC.

Note

Difference between the variants

The difference between the "normal" and "BINARY" variants of the functions is described in Chapter: File transfer HMI <=> NC/PLC (Page 304).

Syntax

The extended copy functions are written as a string in the following form:

COPY_FROM_NC	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_FROM_NC(_BINARY)	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_TO_NC	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)
COPY_TO_NC(_BINARY)	(WinFile,NcFile,seekPos,seekLen,compareString,skipCount)

Parameters

The parameters are described in the table below:

Table A- 28 Parameters of COPY_TO/FROM_NC commands

Name	Description
WinFile	Source or destination of information in the HMI area
NcFile	File name for the NC/PLC environment
seekPos	Seek pointer: Start point of the copy operation, identifier B for block or C for character
seekLen	Window size: Area to be transferred, identifier B for block or C for character
compareString	Search string, up to 32 characters long
skipCount	Number of found search strings to be skipped

The commands return only after complete processing of all subcommands. Errors that are detected during execution of the command can be analyzed based on the LastError variable.

The following example shows a typical application of the extended commands.

File transfer of a program section

File transfer of the first 1024 bytes of the "TP1.MPF" part program to the "test.dat" file in the "C:\NC" directory.

Table A- 29 Example: File transfer of a program section

```
Sub Form_Load ( )
     Label1.LinkTopic = "NCDDE|NCU840D"
     Label1.LinkMode = 2
     Label1.LinkExecute"COPY_FROM_NC(C:\NC\test.dat,
                        Ä/NC/_N_MPF_DIR/_N_TP1_MPF,1 ,1024 , , 0
)"
End Sub
```

Transfer of individual blocks

Pipe transfer of blocks 2 to 4 to the X.MPF part program. Existing blocks are overwritten.

Table A- 30 Example: Transfer of individual blocks

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute " COPY_TO_NC_BINARY ( @xpipe ,
               Ä/NC/_N_MPF_DIR/_N_X_MPF , B2 , 3 , , 0 )"
End Sub
```

Transfer of a block

Text transfer (max. text length of 200 bytes) in the 2nd block of the TEST.MPF part program. The second block is overwritten.

Table A- 31 Example: Transfer of a block

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2 Label1.LinkExecute "COPY_TO_NC ( ""!This is the
content of the
       2nd block"" , /NC/_N_MPF_DIR/_N_TEST_MPF, B2 ,1 , , 0 )"
End Sub
```

A.2.6.3 MAP functions between domains

MAP_ACC_NC

Description

This function allows global user data (GUDs) and machine data of the NC to be communicated to the NCDDE server. This data is saved in files with the extension ACC. Files with the extension ACC are stored in the NC and contain access descriptions of the variables.

Application

With the MAP_ACC_NC command, it is possible to read ACC files from the NC and prepare them for the NCDDE interface. This means that the corresponding connections to this data are created in/communicated to the NCDDE server.

Note

Application

Allows users to communicate new NC data to the NCDDE server. Otherwise these variables/data could not be accessed.

The command behaves like the COPY_FROM_NC command, plus it also decodes the information transferred from the ACC file and prepares it so that it can be represented on the DDE interface.

Syntax

The syntax of the call is as follows:

MAP_ACC_NC

(WinFile, NcFile, TransferState, Area, DataBlock, Timeout, Prefix)

Parameters

The parameters are described in more detail in the following table.

The first three parameters correspond to those of the other domain services. They are also listed here for sake of completeness.

Table A- 32 Parameters of the MAP_ACC_NC command

<table>
<tr><th>Name</th><th colspan="2">Description</th></tr>
<tr><td>WinFile</td><td colspan="2">Source or destination of information in the HMI area</td></tr>
<tr><td>NcFile</td><td colspan="2">File name for the NC/PLC environment</td></tr>
<tr><td>TransferState</td><td colspan="2">Variable that identifies connection status</td></tr>
<tr><td rowspan="10">Area</td><td colspan="2">Area address of the ACC data; see Chapter: 11.1.1, Table 11-1 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".
Here is a complete overview:</td></tr>
<tr><td>Area</td><td>Area address</td></tr>
<tr><td>NC</td><td>0</td></tr>
<tr><td>Mode group</td><td>1</td></tr>
<tr><td>Channel</td><td>2</td></tr>
<tr><td>Axis</td><td>3</td></tr>
<tr><td>Tool/magazine</td><td>4</td></tr>
<tr><td>Feed drive</td><td>5</td></tr>
<tr><td>Main spindle drive</td><td>6</td></tr>
<tr><td>Reserved</td><td>7</td></tr>
<tr><td rowspan="6">DataBlock</td><td colspan="2">Block for the variable service: Numerical hexadecimal value from 00 to FF; see Chapter: 11.3.1 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".
For example (excerpt):</td></tr>
<tr><td>Block designation</td><td>Number (DataBlock)</td></tr>
<tr><td>System data (Y)</td><td>10</td></tr>
<tr><td>Global user data (GUD)</td><td>17</td></tr>
<tr><td>Tool data for OEM (TU)</td><td>24</td></tr>
<tr><td>Magazine directory (TMV)</td><td>2B</td></tr>
<tr><td>Timeout</td><td colspan="2">Time monitoring of the NCK-HMI [s] transaction in seconds</td></tr>
<tr><td>Prefix</td><td colspan="2">Any string placed in front of the ACC variable</td></tr>
</table>

Note

Parameter WinFile

If the WinFile parameter is a file with the extension .NSK, the domain service also generates an NSK file containing the assigned LINK commands in addition to an ACC file.

ACC files

`/NC/_N_NCK_GD2_ACC`	;	Global NC user variable MGUD
`/NC/_N_CH02_GUD_ACC`	;	global user variable in second channel
`/NC/_N_AX_SEA_ACC`	;	Axis-specific setting data
`/NC/_N_CH_TEA_ACC`	;	Channel-specific NC machine data

Creating of connections for drive machine data

`MAP_ACC_NC`	Command header
`L:\MMC2\NCMDACC.NSK`	File name in the Windows environment
`/NC/_N_VS_DIR/_N_VS_TEA_ACC`	NC domain
`trans`	TransferState variable
`5`	Area Here: Numeral 5 for area address of feed drive
`7F`	DataBlock Here: address 7F for drive service values block
`10`	Time monitoring Here: 10 seconds
`/ACC/driveVSA/MD/`	Prefix Here: string that is to be used later to access the data

Table A- 33 Example: Creating connections for drive machine data

```
Sub Form_Load ()
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute "MAP_ACC_NC (L:\MMC2\NCMDACC.NSK,
       /NC/_N_VS_DIR/_N_VS_TEA_ACC,trans,5,7F,10,/ACC/driveVSA/MD/)"
End Sub
```

Access to created connections

Access to a link created previously in the example with the following components:

`/ACC/driveVSA/MD/`	Prefix of the previous call of the MAP command
`$MD_TORQUE_THRESHOLD_X[1]`	Name of the machine data, beginning with $.

Table A- 34 Example: Access to created connections

```
Sub Form_Load ()
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkItem= "/ACC/driveVSA/MD/$MD_TORQUE_THRESHOLD_X[1]"
       Label1.LinkRequest
End Sub
```

Access to global user variable

Access to global user variable.

Examples of some MAP_ACC_NC commands

For these commands, note that the comma between the "WinFile" and "NcFile" parameter must be followed by a blank space.

Table A- 35 Example: Examples of MAP_ACC_NC commands

```
All machine data:
MAP_ACC_NC(c:\tmp\c.nsk,/NC/_N_COMPLETE_TEA_ACC,trans,0,1A,10,/MD/)
All NCK machine data:
MAP_ACC_NC(c:\tmp\nc.nsk, /NC/_N_NC_TEA_ACC, trans,0,1A,10,/NC/)
Channel machine data of channel 1:
MAP_ACC_NC(c:\tmp\ch1.nsk, /NC/_N_CH1_TEA_ACC, trans,2,1A,10,/CH1/)
All axis-specific machine data:
MAP_ACC_NC(c:\tmp\ax.nsk, /NC/_N_AX_TEA_ACC, trans,3,1A,10,/AX/)
All NC-global setting data:
MAP_ACC_NC(c:\tmp\sea.nsk, /NC/_N_NC_SEA_ACC, trans,0,16,10,/SEA/)
All axis-specific setting data:
MAP_ACC_NC(c:\tmp\axs.nsk, /NC/_N_AX_SEA_ACC, trans,3,16,10,/AXSEA/)
All NC-global user data:
MAP_ACC_NC(c:\tmp\gud.nsk, /NC/_N_NC_GUD_ACC, trans,0,17,10,/GUD/)
All channel-specific user data:
MAP_ACC_NC(c:\tmp\gud.nsk, /NC/_N_CH_GUD_ACC, trans,2,17,10,/GUD/)
All NC-global user data 1 (=SGUD):
MAP_ACC_NC(c:\tmp\gd1.nsk, /NC/_N_NC_GD1_ACC, trans,0,17,10,/GUD1/)
All channel-specific user data 1 (=SGUD):
MAP_ACC_NC(c:\tmp\gd1.nsk, /NC/_N_CH_GD1_ACC, trans,2,17,10,/GUD1/)
All NC-global user data 2 (=MGUD):
MAP_ACC_NC(c:\tmp\gd2.nsk, /NC/_N_NC_GD2_ACC, trans,0,2D,10,/GUD2/)
All channel-specific user data 2 (=MGUD):
MAP_ACC_NC(c:\tmp\gd2.nsk, /NC/_N_CH_GD2_ACC, trans,2,2D,10,/MGUD/)
All NC-global user data 3 (=UGUD):
MAP_ACC_NC(c:\tmp\gd3.nsk, /NC/_N_NC_GD3_ACC, trans,0,2E,10,/GUD3/)
All NC-global user data 4 (=GUD4):
MAP_ACC_NC(c:\tmp\gd4.nsk, /NC/_N_NC_GD4_ACC, trans,0,2F,10,/GUD4/)
All NC-global user data 5 (=GUD5):
MAP_ACC_NC(c:\tmp\gd5.nsk, /NC/_N_NC_GD5_ACC, trans,0,30,10,/GUD5/)
All NC-global user data 6 (=GUD6):
MAP_ACC_NC(c:\tmp\gd6.nsk, /NC/_N_NC_GD6_ACC, trans,0,31,10,/GUD6/)
All NC-global user data 7 (=GUD7):
MAP_ACC_NC(c:\tmp\gd7.nsk, /NC/_N_NC_GD7_ACC, trans,0,32,10,/GUD7/)
All NC-global user data 8 (=GUD8):
MAP_ACC_NC(c:\tmp\gd8.nsk, /NC/_N_NC_GD8_ACC, trans,0,33,10,/GUD8/)
All NC-global user data 9 (=GUD9):
MAP_ACC_NC(c:\tmp\gd9.nsk, /NC/_N_NC_GD9_ACC, trans,0,34,10,/GUD9/)
```

A.2.7 PI services

Overview

The IMC command is available for executing program instance (PI) services on the NC/PLC. For compatibility reasons, orders to the NC and PLC via DDE can still be determined. The PI services are grouped in PI.hlp.

The PI services of the NCDDE server include:

PI_START	Order to the NC to execute a command
PI_START_BINARY	Order to PLC to execute a command
PI_STOP	Order to NC to stop the execution of a command
PI_STOP_BINARY	Order to PLC to stop the execution of a command
PI_RESUME	Order to NC to resume a stopped command
PI_RESUME_BINARY	Order to PLC to resume a stopped command

PI_START(_BINARY)

Description

This function allows an order to be issued from the HMI to the NC.

Application

These functions are suitable for starting orders in the NC. Non-binary transfer is suitable for transferring data to the NC. The binary transfer is suitable for transferring data to the PLC, NC and drives.

Syntax

The command line for calling a PI service is structured as follows:

PI_START (server name, parameter 1, parameter 2 ... parameter n, PI name)

PI_START_BINARY (server name, parameter, PI name)

The PI name for NC begins with _N_ followed by six characters. Slightly modified conventions apply to the PLC.

Parameters

The parameters are described in detail in the online help because their meanings can differ significantly depending on the respective PI service.

Selecting a part program

This example shows how the PI order "SELECT" (select program for processing for a channel) selects the "BSP.MPF" part program. Note here that the area path, and not the NC file path, must be entered in this command.

Table A- 36 Example: Selecting a part program

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute
"PI_START(/NC,201,/_N_MPF_DIR/_N_BSP_MPF,_N_SELECT)"
End Sub
```

Activating OB 1

Activate OB1, which is already located in the passive file system of the PLC:

Table A- 37 Example: Activating OB 1

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute "PI_START_BINARY( /PLC,
       ""@1d1@1d0@@0800001P"",_INSE)"
End Sub
```

Stopping the selection of a part program

This example shows how the PI order "SELECT" (select program for processing in a channel) is stopped for the "BSP.MPF" part program.

Table A- 38 Example: Stopping selection of a part program

```
Sub Form_Load ( )
     Label1.LinkTopic = "NCDDE|NCU840D"
     Label1.LinkMode = 2
     Label1.LinkExecute "PI_STOP(/NC,201,/_N_MPF_DIR/
                         Ä_N_BSP_MPF, _N_SELECT)"
End Sub
```

Stopping activation of OB 1

Table A- 39 Example: Stopping activation of OB 1

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute "PI_STOP_BINARY( /PLC,
       Ä""@1d1@1d0@@0800001P"", _INSE)"
End Sub
```

A.2.8 Further commands of the NCDDE server

Overview

The following table provides an overview of the other NCDDE server commands:

Table A- 40 Other NCDDE server commands

Command	Description
NEW	Create local variables
FREE	Delete variables
ANIMATE	Continuously change a "local" variable
CALL	Execute NCDDE commands in files
PLC_MEMORYRESET	Reset the PLC memory

NEW

Description

Creates a local/internal variable in the NCDDE server that can then be accessed.

Application

With the "NEW" command, an NCDDE-local/internal variable is created. There is no communication with the NC when this variable is accessed. If a variable called VarName already exists, it is deleted beforehand (behavior same as "FREE" command, see below).

Syntax

```
NEW       (VarName , value)
```

Parameters

Table A- 41 Parameters of NEW

Parameter	Syntax	Description
VarName	<String>	Name of the variable that is to be created
Value	<Parameter>	Initialization value for the variable

Creating an internal variable

Creates the "test" variable in the NCDDE server and initializes this with the value 10.0.

Table A- 42 Example: Creating an internal variable

```
Sub Form_Load ( )
Label1.LinkTopic = "NCDDE|NCU840D"
Label1.LinkMode = 2
Label1.LinkExecute " NEW ( test , 10.0 )"
End Sub
```

FREE

Description

Deletes a variable in the NCDDE server

Application

The "FREE" command deletes a variable created with the "NEW" and "LINK" commands.

If the variable is currently being used as a status variable by a file transfer service, then the "FREE" command is rejected.

If Advise Links (Hotlinks) to this variable exist, they are removed. Other transactions with the CNC are canceled.

Syntax

FREE (VarName)

Parameters

Table A- 43 Parameters of FREE

Parameter	Syntax	Description
VarName	<String>	Name of the variable that is to be deleted

Deleting an internal variable

Deletes the "test" variable in the NCDDE server.

Table A- 44 Example: Deleting an internal variable

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute " FREE( test )"
End Sub
```

ANIMATE

Description

Causes the NCDDE server to continuously change a local variable created with "NEW". Numerical values are incremented in a cycle of approximately 1 second.

Application

Can be used for testing your application.

Syntax

ANIMATE (VarName)

Parameters

Table A- 45 Parameters of ANIMATE

Parameter	Syntax	Description
VarName	<String>	Name of the variable that is to be deleted

Changing an internal variable

Continuously changes the "test" variable in the NCDDE server.

Table A- 46 Example: Changing an internal variable

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute " ANIMATE( test )"
End Sub
```

CALL

Description

Serves the interpretation of a command file.

Application

The CALL command executes NCDDE commands stored in files. Each line of the file is passed to the NCDDE server as a command. The file may contain comments and blank lines. For NCDDE command files, the name extension .NSK must be used uniformly.

Note

Application

Allows creation of the NCDDE server with the variables it needs.

Syntax

CALL (FileName)

Parameters

Table A- 47 Parameters of CALL

Parameter	Syntax	Description
VarName	<String>	Name of the NCDDE command file

For example, see file "\MMC2\ NCDDE311.NSK"

PLC_MEMORYRESET

Description

The PLC_MEMORYRESET command of the NCDDE server resets the PLC memory. /PLC must be specified as the area address.

Application

Reset the PLC memory

Syntax

PLC_MEMORYRESET (AreaAdr)

Parameters

Table A- 48 Parameters of PLC_MEMORYRESET

Parameter	Syntax	Description
AreaAdr	<String>	Area address

Reset the PLC memory

Resets the PLC memory; the PLC must be stopped beforehand

Table A- 49 Example: Reset the PLC memory

```
Sub Form_Load ( )
       Label1.LinkTopic = "NCDDE|NCU840D"
       Label1.LinkMode = 2
       Label1.LinkExecute "PLC_MEMORYRESET(/PLC)"
End Sub
```

A.2.9 OEM Visual Basic Controls (OCX files)

Overview

Some shortcomings of DDE communication with standard Visual Basic controls are overcome with these controls.

Note

Recommendation

When developing an OEM application, use of these OEM Visual Basic controls is recommended when accessing the NCDDE server.

Standard controls such as Label and TextBox offer DDE communication. However, these have some shortcomings:

- loss of events. It is not ensured that the change procedure in the VB program is started when the value of a DDE variable changes while Link Mode = 1. (can only be remedied by polling the value via "Timer control")
- DDE functionality cannot be nested. In a DDE change procedure, additional DDE functionality of the controls cannot be activated (can only be remedied by a timer or similar)
- Only synchronous transactions are implemented. The response times for setting up Hotlinks and for Requests are quite long. This is especially true if these actions are distributed across several CPUs (NC, PLC).
- Resource consumption is high. A DDE connection is set up for each control instance that uses DDE. Each connection uses two window handles and thus scarce user resources.
- For NCDDE, LastError is not handled in a user-friendly fashion. For communication with the NC via NCDDE, this DDE variable "LastError" provides detailed diagnostics of errors that have occurred. This variable is specific to each DDE conversation and only valid if the DDE Return is "DDE_FNOTPROCESSED".

A.2.9.1 DDECTL.VBX file

With the switchover of VB applications from 16 bit to 32 bit, DDECTL.VBX is no longer applicable; see also Chapter: "Porting VB applications from 16 bit to 32 bit".

A.2.9.2 DDECTL.VBX file

With the switchover of VB applications from 16 bit to 32 bit, DCTL.VBX has become DCTL.OCX; see the following chapter and Chapter: "Porting VB applications from 16 bit to 32 bit".

A.2.9.3 DCTL.OCX file

Overview

The Visual Basic DCTL.OCX control is a graphic control with extended DDE functionality. It is similar to the standard Label control, but offers several additional advantages:

- Minimum use of Windows resources:

  DDE Request, DDE Poke and DDE Execute require Windows resources only temporarily. Altogether, all DDE Hotlinks of a Windows process with the DCTL.OCX control need only one Windows handle
- Close connection to the NCDDE server:

  For example, this control supplies the LastError value following unsuccessful DDE transactions
- Higher speed:

  Applications run faster because multiple/simultaneous transactions with a server are possible.
- Faster output:

  Through optimized screen output and index filtering, a faster display on the screen is possible. This also simplifies the BASIC programming.
- Avoidance of side effects:

  Typical side effects of Visual Basic controls, such as termination of programmed connections by pressing the Escape key, are avoided.

This section examines the properties of the new control followed by the additional events. Finally, some examples show possible uses of the DCTL.OCX.

Properties

Most properties of the DCTL.OCX control correspond to those of the standard controls of Visual Basic. These include:

- Style properties
- Color properties
- Base properties
- Drag properties
- Font properties

Some properties differ from those of the standard controls of Visual Basic:

- DDE properties
- HorAlignment property
- VertAlignment property
- WordBreak property
- TabSize property

- LastError property
- Data property
- DataToCaption property
- LinkCmd property
- LinkNext property
- LinkFilter property.

DDE properties

DDE properties include:

- Link Item
- Link Topic (the preset NCDDE by default)
- Link Timeout (for synchronizing LinkCmd).

HorAlignment property

This property specifies the horizontal alignment of the heading.

Table A- 50 Horizontal alignment

Value	Property
LeftJustify	Left-justified (default)
RightJustify	Right-justified
HorCenter	Centered

VertAlignment/ Multiline property

This property specifies the vertical alignment of the heading. Alternatively, a multi-line display can also be set. In the case of a multi-line display, the line breaks are defined by the WordBreak property.

Table A- 51 Vertical alignment

Value	Property
VertCenter	Vertically centered (default)
TopJustify	At the top
BottomJustify	At the bottom
MultiLine	Multi-line

WordBreak property

If the VertAlignment/Multiline property is set to Multiline (value = 3), the WordBreak property then defines the type of line break.

Table A- 52 Type of line break

Value	Property
False	Line break by CR/LF (carriage return/line feed)
True	Automatic line break when the word no longer fits in the line. Line break by CR/LF is also possible here.

TabSize property

This property specifies the tab size. The default value is eight characters. Up to 255 characters is possible.

LastError property

LastError property This property enables the output of error messages. At the beginning of a DDE connection with a server, the LastError error variable is reset (LastError = 0). If an error occurs during the transaction and it is detected by the DCTL control, this control determines a detailed error code that can be accessed using the LastError property.

Note

Decoding

The DCTL control does not decode errors that are transferred as data: These include # characters or characters from the NCDDE server.

A description of the LastError variable for the DCTL control is provided in Chapter 11.7.1.

Data property

The "Data" property serves as an argument for the following DDE transactions:

Table A- 53 Arguments of the DDE transactions

DDE transaction	Argument
Request	Requested variable value when property DataToCaption = False is set.
Advise Link	Updated values when property DataToCaption = False is set.
Poke	Value to be transferred
Execute	Instruction to be executed

DataToCaption property

The "DataToCaption" property defines the destination of the data of a DDE transaction.

Table A- 54 Type of line break

Value	Meaning
False	Data destination is Caption property.
True	Data destination is Data property.

LinkCmd property

Changes of the "LinkCmd" property start the DDE activities of the DCTL control. If no activity is present, LinkCmd = 0.

Table A- 55 Property of LinkCmd property

Ser. No.	Change to	DDE activity	End by
1	Advise Link	Advise Link is established. Return after establishment of Advise Link. Advise Link can be cleared using a Stop command.	Stop
2	Advise Link_NotifyData	Same as no. 1, plus action (1) when DDE data arrives.	Stop
3	Advise Link_NotifyDataWhenVisible	Same as no. 1, plus action (2) when DDE data arrives.	Stop
4	Advise LinkAsync	Initiates establishment of Advise Link. Return before establishment of Advise Link. Advise Link can be cleared using a Stop command.	Stop
5	Advise LinkAsync_NotifyData	Same as no. 4, plus action (1) when DDE data arrives.	Stop
6	Advise LinkAsync_NotifyDataWhenVisible	Same as no. 4, plus action (2) when DDE data arrives.	Stop
7	Stop	Removes an Advise Link. Return after removal of the Advise Link.	Itself
8	StopAsync	Initiates removal of Advise Link. Return before removal of Advise Link.	Sync
9	StopAsync_Notify	Same as no. 8, plus action (1) after Advise Link has been removed.	Sync
10	StopAsync_NotifyWhenVisible	Same as no. 8, plus action (2) after Advise Link has been removed.	Sync
11	Request	Reading of a DDE variable. Return after variable has been read.	Itself
12	RequestAsync	Initiate reading of a DDE variable. Return before variable has been read.	Sync
13	RequestAsync_Notify	Same as no. 12, plus action (1) after variable has been read.	Sync

Ser. No.	Change to	DDE activity	End by
14	RequestAsync_NotifyWhenVisible	Same as no. 12, plus action (2) after variable has been read.	Sync
15	Execute	Sends an instruction to the server. Return after execution is complete.	Itself
16	ExecuteAsync	Initiates an instruction to the server. Return before execution is complete.	Sync
17	ExecuteAsync_Notify	Same as no. 16, plus action (1) after execution is complete.	Sync
18	ExecuteAsync_NotifyWhenVisible	Same as no. 16, plus action (2) after execution is complete.	Sync
19	Poke	Writing of a DDE variable. Return after variable has been written. The value is written from data (not from caption).	Itself
20	PokeAsync	Initiates writing a DDE variable. Return before execution is complete.	Sync
21	PokeAsync_Notify	Same as no. 20, plus action (1) after writing is complete.	Sync
22	PokeAsync_NotifyWhenVisible	Same as no. 20, plus action (2) after writing is complete.	Sync
23	Sync	Completes asynchronous instructions in the same way as synchronous instructions. No execution when no asynchronous instruction is being used.	Itself

Actions

The actions named in the above table are:

- Action (1)

  An attempt is made to call the DdeNotify event procedure. If Visual Basic does not call an event procedure at this time - or the parameter of the event procedure has not been changed - the DCTL control attempts ten times per second to send this event as long as the parameter of the DdeNotify event procedure does not change.

- Action (2)

  The DCTL control calls the DdeNotify event procedure when it receives a paint message from Windows. To ensure that these Paint messages are also generated, the pixel in the top left corner of the control is invalid as long as the parameter of DdeNotify does not change. This mechanism also suppresses displays when the control is not visible.

Note

New DDE activities

New DDE activities should not be begun until previous activities are finished. This is possible with the parameters named in the right column (End by).

The Hotlinks of all DCT controls in the same window share a DDE connection when they have the same LinkTopic property. The DDE connections of the other activities (except Hotlink) are created dynamically and removed again. This behavior, together with the fact that a DCTL control has no window, significantly reduces the demands on Windows resources.

Note

Change of the LinkCmd property

A change of the LinkCmd property leads to evaluation of the Link Topic, Link Timeout and Link Item properties. That is, errors relating to these variables will occur when the LinkCmd property changes. They must therefore be intercepted here.

LinkNext property

The optionally assignable LinkNext property holds the name and optionally the index of another DCTL control. If LinkNext is not empty, the DCTL control checks a string communicated via an Advise Link for index specifications of the NCDDE (five numerals followed by a colon). It divides the text into indexed text components and forwards them to a chained list of DCTL controls. The control whose LinkFilter property corresponds to this index receives the text. Texts not distributed in this way are lost.

LinkFilter property.

The LinkFilter value can range from 0 to 65535. Its use is presented in the above LinkNext property section.

Events for DCTL.OCX

Most events of the DCTL control are identical to those of other standard controls of Visual Basic, such as:

- Click
- DblClick
- MouseDown
- MouseMove
- MouseUp
- DragDrop
- DragOver
- KeyDown
- KeyPress
- KeyUp.

DdeNotify event

The DdeNotify event has been implemented especially for DDE communication. It indicates the arrival of new Advise Link data or the completion of an asynchronous DDE transaction. Its use is described in the description of the LinkCmd property (actions (1) and (2)).

Syntax

Sub ctlname_DdeNotify (Flag As Integer)
With the Flag argument that indicates to the DCT control when the event has actually arrived at the basic level.
The Flag value is expected to change at each call of the event procedure. That is because the DCTL control continues to initiate the DdeNotify event until the Flag argument changes. If this change does not take place, a cooperative permanent activity results that burdens the system.

A.2.9.4 Applications of DCTL.OCX

Reading and displaying a variable

A DDE variable is to be read immediately and displayed on the screen. For this purpose, a DCTL control (e.g. called DCTL1) should then be positioned at a suitable location on the screen. The associated code then appears as the following:

Table A- 56 Example: Reading and displaying a variable

```
Sub Form_Load ( )
       Dctl1.LinkItem = "/Channel/Parameter/R[1]" ’ the variable name
       Dctl1.DataToCaption = TRUE ’that’s default, can be omitted
       Dctl1.LinkCmd = 11 ’ commands the reading
       ’ here Dctl1.Caption holds the value of the DDE variable
End Sub
```

Reading of a variable to the Data property

A DDE variable is to be read and processed immediately without being displayed on the screen. For this purpose, a DCTL control (e.g. named DCTL2) of type Label is then to be arranged in a form. The associated code then appears as the following:

Table A- 57 Example: Read to the Data property

```
Sub Form_Load ( )
       Dctl1.LinkItem = "/Channel/Parameter/R[1]" ’ the variable name
       Dctl1.DataToCaption = FALSE ’ routing data to the Data property
       Dctl1.LinkCmd = 11 ’ commands the reading
       ’ here Dctl1.Data holds the value of the DDE variable
End Sub
```

Writing a DDE variable

A DDE variable is to be written. For this purpose, a DCTL control (e.g. named DCTL3) of type Label is then to be arranged in a form. The associated code then appears as the following:

Table A- 58 Example: Write a variable

```
Sub Form_Load ( )
       Dctl1.LinkItem = "/Channel/Parameter/R[1]" ’ the variable name
       Dctl1.Data = 12 ’ the value
       Dctl1.LinkCmd = 19 ’ commands the writing
       ’ here the NC variable is already successfully set to 12
End Sub
```

Executing a command

A DDE command is to be communicated to a server. For this purpose, a DCTL control (e.g. named DCTL4) of type Label is then to be arranged in a form. The associated code then appears as the following:

Table A- 59 Example: Executing a command

```
Sub Form_Load ()
       Dctl1.Data = "Pi_start(/NC,001,_N_SET_OF)" ’ the command
       Dctl1.LinkCmd = 15 ’ sends the command
       ’ here the command is already successfully executed
End Sub
```

Displaying a DDE Hotlink

The value of a DDE variable is to be displayed on the screen. For this purpose, a DCTL control (e.g. called DCTL4) should then be positioned at a suitable location on the screen. A hotlink must be setup that runs as a background activity of the DCTL control.

The associated code then appears as shown in the following example: However, it is also possible to execute the coded property setting at the time of the design.

Table A- 60 Example: Hotlink with DCTL

```
Sub Form_Load ( )
Dctl1.LinkItem = "/Channel/Parameter/R[1]" ’ the variable name
Dctl1.DataToCaption = TRUE ’ that’s default, can be omitted
Dctl1.LinkCmd = 4 ’ initiates the creation of a hotlink
End Sub
```

Acceleration through parallel work

Several independent DDE activities must be executed when a form is loaded. In this case, a fast loading of the form is desirable. This is best achieved with DDE activities running in parallel – in interaction with the DCTL control.

A suitable code is shown in the next example.

Table A- 61 Example: Acceleration through parallel work

```
Sub Form_Load ( )
’ start reading variable 1
Dctl1.LinkItem = "/Channel/Parameter/R[1]" ’ the variable name
Dctl1.LinkCmd = 12
’ initiates the reading
’ start reading variable 2
Dctl2.LinkItem = "/Channel/Parameter/R[2]" ’ the variable name
Dctl2.LinkCmd = 12
’ initiates the reading
’ start reading variable 3
Dctl3.LinkItem = "/Channel/Parameter/R[3]" ’ the variable name
Dctl3.LinkCmd = 12
’ initiates the reading
’ start a hotlink into display
Dctl4.LinkItem = "/Channel/Parameter/R[4]" ’ the variable name
Dctl4.DataToCaption = TRUE
’ that’s default, can be omitted
Dctl4.LinkCmd = 4
’ creates the hotlink
’ start a command execution
Dctl5.Data = "Pi_start(/NC,001,_N_SET_OF)" ’ the command
Dctl5.LinkCmd = 16
’ commands execution
’ here the tree variable accesses, the hotlink creation and the ’ ’command
are
working in parallel. You can not be sure that any of ’ ’ them has complet-
ed.
Dctl1.LinkCmd = 23
’ wait until variable 1 read
Dctl2.LinkCmd = 23
’ wait until variable 2 read
Dctl3.LinkCmd = 23
’ wait until variable 3 read
Dctl5.LinkCmd = 23
’ wait until command executed
’ here the variable accesses and the command have completed, the hotlink
will
’ show it’s value on screen as soon as possible.
End Sub
```

Acceleration through text arrangement

A large amount of data must be read and displayed at high frequency. The BASIC language should not be used for a display. In addition, the scope of the transferred data should be minimized. On the NCDDE side, array access to data and a combination of array access with data preparation ensures that these requirements are met. The DCTL control provides a multiline display and index filter functions for this.

Table A- 62 Example: Acceleration through text arrangement

```
’NCDDE array access with "Field" data preparation - Dctl index filtering:
’ high frequency display of 5 values in 5 different controls
Dctl1.LinkItem = "/Channel/Parameter/R[1,5](!""!d%12.5g"")" ’ variable
Dctl1.LinkFilter = 1         ’      index of accepted data
Dctl1.LinkNext = "Dctl2"      ’     linkage to the next control
Dctl2.LinkFilter = 2   ’            index of accepted data
Dctl2.LinkNext = "Dctl3"      ’     linkage to the next control
Dctl3.LinkFilter = 3         ’      index of accepted data
Dctl3.LinkNext = "Dctl4"     ’      linkage to the next control
Dctl4.LinkFilter = 4          ’     index of accepted data
Dctl4.LinkNext = "Dctl5"     ’      linkage to the next control
Dctl5.LinkFilter = 5         ’      index of accepted data
Dctl1.LinkCmd = 4            ’      initiates the creation of a hotlink
NCDDE array access - Dctl multiline display:
’ high frequency display of 5 values in a column
Dctl1.LinkItem = "/Channel/Parameter/R[1,5](""!d%12.5g"
Dctl1.LinkItem = Dctl1.LinkItem + Chr$(13) + Chr$(10)+""")"
Dctl1.DataToCaption = TRUE ’ that’s default, can be omitted
Dctl1.VertAlignment = 3 ’ multiline selection
Dctl1.LinkCmd = 4 ’ initiates the creation of a hotlink
```

Notification after a change

If the layout of a screen display depends on a variable that is accessed via DDE, this variable is fetched via a Hotlink in the DCTL control. With the notification of a variable change, the screen content can then be rearranged. Because this operation is very time consuming, it should only be executed when the form is visible.

Table A- 63 Example: Notification after a change

```
Sub Form_Load ( )
‘basic code that creates a hotlink with notification “when visible"
Dctl1.LinkItem = “/Channel/Parameter/R[1]" ‘the variable name
Dctl1.LinkCmd = 6 ’ initiates the creation of a hotlink
‘handler for the notification event
End Sub
Sub Dct1_DdeNotify ( Index As Integer, Flag As Integer )
Flag = Flag + 1 ‘Flag MUST change
... ‘rearrangement to be done
End Sub
```

Error handling

The typical error handling for reading, writing and executing is shown here.

Table A- 64 Example: Error handling

```
On Error Goto TypicalErrorHandling
Dctl1.LinkCmd = 11 ’ a DDE activity
...
TypicalErrorHandling:
Select Case Dctl1.Lasterror \ 16777216’ selection by error source
       Case 2 ’ MPI level error
       ... ’ e. g. no connection to NC
       Case 3 , 5 ’ NC/PLC level error
       ... ’ e. g. non-existent variable
       Case 7 ’ Dctl level error
       Select Case Dctl1.Lasterror MOD 256 ’ selection by error code
       Case 7 ’ Dctl level timeout occurred
       ...
       Case Else ’ other Dctl level errors
       ...
       End Select
       Case Else ’ other error sources
       ...
End Select
...
```

A.2.10 Diagnostics - Options for NCDDE access

A.2.10.1 Test functions of NCDDE server

Overview

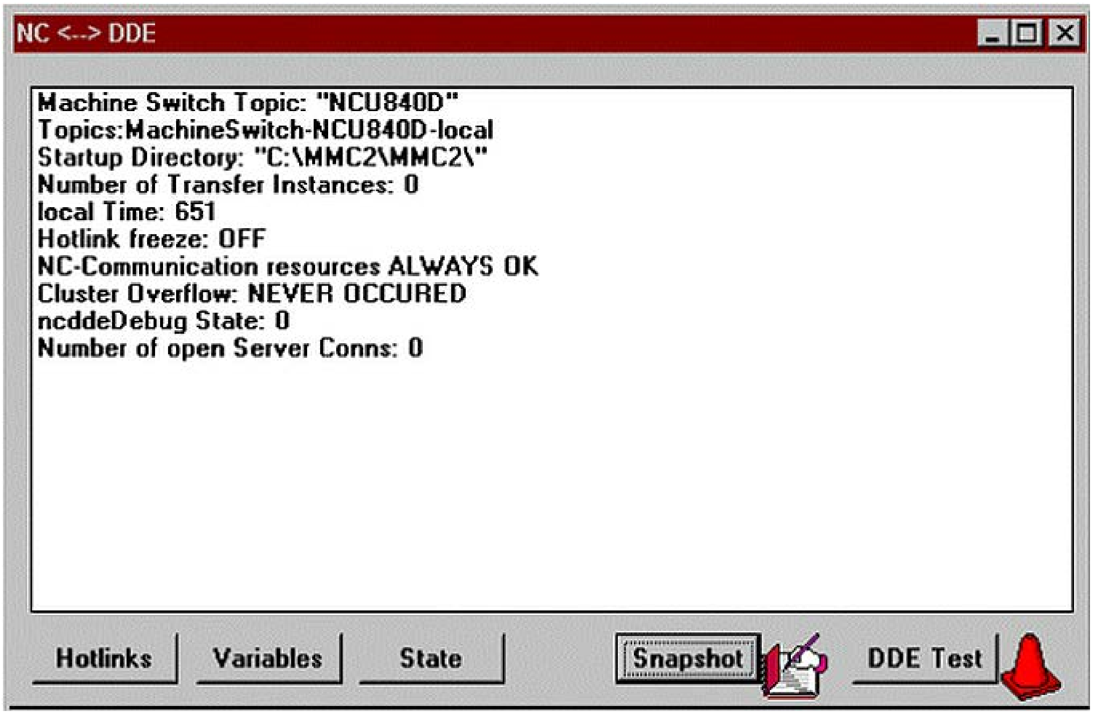
The test functions of the NCDDE server provide information, including information about the local and external tags that were known at the time that the file was created in the NCDDE server.

These test functions are called as follows:

1. Start the NCDDE server in the SINUMERIK 840D MMC-OEM program group.
2. Use the <ALT> and <TAB> keys to switch to the NCDDE program - NC Communication DDE-Server.

   An icon is displayed.
3. Click on the icon.

   The "NC <..> DDE" window appears.



```
Machine Switch Topic: "NCU840D"
Topics:MachineSwitch-NCU840D-local
Startup Directory: "C:\MMC2\MMC2\"
Number of Transfer Instances: 0
local Time: 651
Hotlink freeze: OFF
NC-Communication resources ALWAYS OK
Cluster Overflow: NEVER OCCURED
ncddeDebug State: 0
Number of open Server Conns: 0
```

The following functions are intended mainly for troubleshooting in the environment of the NCDDE server.

Hotlinks

The Advise Links (hotlinks and warning links) are summarized in the following table.

Table A- 65 Hotlinks

Column	Information		
1	PDU reference	Internal value: possible PDU reference for communication with NC and PLC.	
2	Advise Link	LOCAL	Link to local variable
		REMOTE	Link to external variable
		PILED	Duplicate external Advise Link, was switched to another order.
3	Update time	Time of the last Refresh PDU in an internal time measure of the NCDDE server.	
4	LastError variable	LastError data; see: Chapter 11.7 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1". It does not always conform to the value that can be queried on the DDE interface of the server, because the last error of multiple transactions for a connection can be queried there.	
5	Variable name	Variable identifier; see: Chapter 11 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".	

Variables

Displays the variables for which the NCDDE server holds a connection and the location of the storage:"LOCAL" or "PLC/NC".

Snapshot

When this button is pressed, a file named "NCDDE_X.TXT" is generated, which contains the status, Hotlinks and variables of the NCDDE server.

DDE test

Starts a test program "DDETEST.EXE" with the following functions:

Table A- 66 DDE test commands

Command	Action	Description
Passive	None	Initial state, no function active
Hotlink	Start	Establish Advise Link
Request	DoIt	Read variable
Poke	DoIt	Write variable
Execute	DoIt	Executes a service

The connected NC should be marked under service topic, e.g. NCDDE|NCU840D.

"DEFAULT_NC" reads the settings from file "MMC.INI".

The Command function is toggled between the five options by clicking on it.

Descriptions of the LastError error messages are provided in Chapter 11.7.

A.2.10.2 Connection status

Variable NCState

The server publishes the connection state to the CNC via the NCState variable, locally available on the server.

This variable exists immediately after the server starts. It only differs from other variables local on the server by the fact that it cannot be changed via the DDE interface. The variable can assume the following values:

Table A- 67 Values of NCState variables

Value	Description
0	Normal operation
1	Some connections to the CNC have an error condition
2	All connections to the CNC have an error condition
3	Interpretation of the startup file
4	Server initialization

A.2.10.3 Handling error cases

Error acknowledgments from the NC

Error conditions such as lack of resources, access protection violation, incorrect operating states etc. are communicated to the NC via acknowledgments. If the NCDDE server cannot process these error conditions, then the corresponding Request, Peak and Execute transactions of the DDE interface are terminated with an error, i.e. a result is not obtained.

Variable LastError

Detailed diagnostics is provided using the LastError variable, to which a connection existed for the last transaction. It can be read out using the LastError Link Item.

The variable is set to zero once it has been read out. It indicates the last error registered in the NCDDE server. Variable LastError comprises four bytes, where for each byte, in descending order (high byte è low byte), the following error groups can be found:

- Higher-level error class, error source
- Error region
- Error class
- Error code

The significance of the individual error codes are listed in Chapter 11.7: User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1".

Connection terminated to the NC

When the connection is interrupted, the NCDDE server negatively acknowledges active Request, Poke and Execute transactions.

While the connection is interrupted, it rejects the execution of transactions; at the same time, it continually attempts to re-establish a connection to the NC. The connection state is published in the NCState variable that exists locally on the server.

Handling Advise Links

If, for Advise Links, the connection to the NC is interrupted, then the value supplied from the NCDDE server corresponds to the "#" character. Once the connection has been re-established, Advise Links are again setup on the CNC.

Lack of adequate resources in the NCDDE server

If a lack of adequate resources occurs in the NCDDE server, then the corresponding transactions at the DDE interface are terminated with an appropriate error identifier.

A.2.11 Configuring the NCDDE server for network access operations

Only for development purposes.

A.2.12 Expanded functionality of the NCDDE server

A.2.12.1 Multi-variable service

Overview

The multivariable service allows DDE access to several variables with one NCDDE order. It accelerates access to several individual variables; however, it can only be used for read and write operations (not hotlink). The items of the appropriate individual variables/array access operations separated with "|" should be specified as item. The data supplied for read access operations are tightly compressed. Just the same as before for array access operations, separators must be parameterized using format data - or via a new access modification; see Chapter: New access modifications (Page 339). For write access operations, the first character of the supplied data is interpreted as separator for the individual data sets.

Limits

- A maximum of eight tightly compressed PDUs are issued for each order. This means that normally more than 100 individual access operations can be handled in one order. (The specific number must be empirically determined.)
- All PDUs are sent to a target address. As a consequence, PLC and NC access operations cannot be mixed in one order. Further, it is not permissible to mix channel-specific access operations for different channels. (NC requirement). The same applies when accessing drive-specific variables.
- Only real variables (OPI interface/PLC-BUB) can be addressed using the multivariable service. Therefore, accessing date and time, system status list, directory information, ... is not possible.
- Caution, the DDE item size is limited to 255 characters. If the item string exceeds this size, then the item must be indirectly specified; see Chapter: Indirect item output (Page 338).

Example for reading and writing using the Item multivariable service:

/channel/parameter/r[1,2](|“!l%ld“)|/channel/parameter/r[10](|)

Data, e.g.: |1|2|10.000000

A.2.12.2 Indirect item output

The indirect item specification permits items exceeding 255 characters (up to 4 KB). The content of a local NCDDE variable can be used as item for DDE access operations. In this specific case, the name of the local variable must be specified as item with a ">" character as prefix.

Example for accessing R10:

Exec:	NEW(x,"/channel/parameter/r[10]")
Item:	>x
Data:	e. g. 10.000000

Note

Data length

The data length is limited to 4 KB when writing variables, and when executing commands in the NCDDE server. Error message 0X01050414 is output when this value is exceeded.

A.2.12.3 New access modifications

The access can be modified as follows using control characters "|" and "^", attached in round parenthesis to the item string:

"I" For CF_TEXT read access operations, inserts a "|" character before each individual piece of data.
For write access operations, this is not evaluated; see the example in Chapter: Multi-variable service (Page 337).

"^" The hotlink switch off (DEBA/DEBR) is not effective for a variable designated in this way.

A.2.13 Access to global user variables GUD, SGUD, MGUD, UGUD, GD3 ... GD9

Overview

Global data is available for the NC - and for a channel.

NC-specific global user variables exist once in the control system. They are suitable for settings independent of the specific channel - or for program coordination.

Channel-specific global user variables are available once for each channel. They are suitable for channel-specific settings, or for transferring data between programs, which run in a channel.

The same technique can also be applied to local user data. In this case, the same statements essentially apply.

In order that the NCDDE server can access user variables, these must first be defined and then activated.

To cluster variables, only the NSK files have to be created and subsequently integrated. This is realized in five steps:

1. Create a definition file
2. Copy this definition file to directory /_N_DEF_DIR of the NC
3. Activate the user data as *.ACC file with INITIAL.INI
4. Create the *.NSK file using the MAP command
5. Transfer the created *.NSK file into the NSK file of the NCDDE server

Definition file

Global user variables are defined in definition files (blocks) with defined names:

- _N_GUD_DEF for GUD
- _N_SGUD_DEF for GD1 = SGUD global Siemens data
- _N_MGUD_DEF for GD2 = MGUD global machine builder data
- _N_UGUD_DEF for GD3 = UGUD global user data
- _N_GUD4_DEF to _N_GUD9_DEF for GD4 to GD9

These blocks are located in the permanent directory /_N_DEF_DIR of the NC. The total number files for global data depends on general machine data 18118 (MM_NUM_GUD_MODULES); for additional information, see the commissioning instructions.
4 is set as default value.

Defining global data

Global data are defined using:

Definition header	DEF
Area	NC or CHAN
Number type	For example, REAL or INT
Variable name	Variable name, e.g. RETRACT
Parameter	Parameters in square parentheses
Comment	Comment text beginning with a semicolon

Additional information and data can be taken from the Programming Instructions.

Creating definition files

Definition files can be created in the NC or in the MMC.

In the NC:

A definition file for global variables, which can be created at the part program level in the NC, belongs in directory /_N_DEF_DIR, and comprises the following:

- Program identifier in the first line
- Comment line with path data (is evaluated)
- Definitions • concluded with M02, M17 or M30.

Table A- 68 Example: Defining global variable in the NC

```
%_N_MGUD_DEF
; $PATH=/_N_DEF_DIR
DEF NCK REAL RUECKZUG ; definition of global variable for NCK
DEF CHAN INT TABELLE[100] ; definition of channel-specific variable
DEF CHAN REAL BLF_OFFS_X
M17 ; this line must be concluded with RETURN
```

In the HMI:

A definition file for global variable with file name MGUD.DEF, as can be generated in the HMI, is located, for example in directory C:\TMP and comprises:

- Definitions
- Concluded by M02, M17 or M30.

Table A- 69 Example: Defining a global variable in the HMI

```
DEF NCK REAL RUECKZUG ; definition of global variable for NCK
DEF CHAN INT TABELLE[100] ; definition of channel-specific variable
DEF CHAN REAL BLF_OFFS_X
M17 ; this line must be concluded with RETURN
```

Note

Transferring file

This file still has to be transferred by the MMC into directory/_N_DEF_DIR of the NCK. This is done using the domain service
COPY_TO_NC:COPY_TO_NC(C:\TMP\MGUD.DEF,/NC/_N_DEF_DIR/
Ä_N_MGUD_DEF,trans).

Activating user data

User data are activated by copying a file with name "INITIAL.INI" into the NC. This file can be very short: It is completely sufficient to enter M17, followed by RETURN.

For file INITIAL.INI, located in directory C:\TMP, the following applies:
`COPY_TO_NC(C:\TMP\INITIAL.INI, /NC/_N_INITIAL_INI, Ätrans)`

Two ACC files are created with the names:

_N_NC_GD2_ACC	for the global user variables
_N_NC_GD2_ACC	for the channel-specific user variables

(Applicable for the example above, with MGUD = GD2.)

Note

Backing up

Before copying file INITIAL.INI, backup all programs, frames and machine data as the static memory is formatted.

Creating an NSK file for the NC

By calling the MAP command, from the ACC files, NSK files with the same name are generated for global user variables of the NC.

The example shows the call under Visual Basic.

Calling the "MAP_ACC_NC" command

C:\MMC2\MGUD_NCK.NSK:	File name in the Windows environment
/NC/_N_NC_GD2_ACC	: NC domain
trans	: TransferState variable
0	: NC area
2D	: MGUD block type
10	: Time monitoring of the transaction with 10 s
/ACC/NCK/MGUD/:	Any character string that can be selected by the user, which is located in front of the user variable.

Table A- 70 Example: Creating an NSK file for the NC

```
Sub Form_Load ( )
       Label1.LinkTopic = ”NCDDE|MMC2HW0”
       Label1.LinkMode = 2
       Label1.LinkExecute ”MAP_ACC_NC(C:\MMC2\MGUD_NCK.NSK,
               Ä/NC/_N_NC_GD2_ACC, trans, 0, 2D, 10,
               Ä/ACC/NCK/MGUD/)”
End Sub
```

Creating an NSK file for the channel

By calling the MAP command, from the ACC files, NSK files with the same name are generated for global user variables of the channel.

The example shows the call under Visual Basic.

Calling the “MAP command“

C:\MMC2\MGUD_CH.NSK:	File name in the Windows environment
/NC/_N_CH_GD2_ACC	: NC domain
trans	: TransferState variable
2	: Channel area:
2D	: MGUD block type
10	: Time monitoring of the transaction with 10 s
/ACC/CH/MGUD/:	Any character string that can be selected by the user, which is located in front of the user variable.

Table A- 71 Creating an NSK file for the channel

```
Sub Form_Load ()
       Label1.LinkTopic = ”NCDDE|MMC2HW0”
       Label1.LinkMode = 2
       Label1.LinkExecute
”MAP_ACC_NC(C:\MMC2\MGUD_CH.NSK,/NC/_N_CH_GD2_ACC
      ,trans,2,2D,10,/ACC/CH/MGUD/)”
End Sub
```

Note

Formats

The NSK file is created in the binary format (*.MAP) and in the ASCII format (*.NSK).

Transfer into the NSK file of the NCDDE server

Files MGUD_NCK.NSK and MGUD_CH.NSK, created in this example, should be inserted into the NSK file of the NCDDE-NCDDE311.NSK with:

```
REM IMPORT ADDITIONAL USER VARIABLES
CALL(MGUD_NCK.NSK)
CALL(MGUD_CH.NSK)
REM
```

Access to NC user variables

It will be demonstrated how an NC user variable can be read out using the RUECKZUG [RETRACT] variable as example.

Table A- 72 Access to the RUECKZUG [RETRACT] NC user variable

```
Sub Form_Load ( )
       CtlName1.LinkTopic = g_chNCDDEServiceName
       CtlName1.LinkItem = ”/acc/nck/mgud/RUECKZUG”
       CtlName1.LinkMode = 2
       CtlName1.LinkRequest
       CtlName1.LinkMode = 0
End Sub
```

Access to channel user variables

It will be demonstrated how a channel user variable can be read out using BLF_OFFS_X variable as an example.

Table A- 73 Example: Access to channel user variable BLF_OFFS_X

```
Sub Form_Load( )
       CtlName.LinkTopic = g_chNCDDEServiceName
       CtlName.LinkItem = ”/acc/ch/mgud/BLF_OFFS_X[u2]” ’for 2nd channel
       CtlName.LinkMode = 2
       CtlName.LinkRequest
       CtlName.LinkMode = 0
End Sub
```

Note

Further information

Additional information for creating and applying user data is provided in the Commissioning Instructions /IAD/ - and in the Programming Instructions/PA/.

A.2.14 Online help for variables

Overview

The online help for variables supports OEM programming personnel when selecting and defining data from the NC area. It is structured just like all the other help files under Windows, and offers the same functional scope. The variable online help is independent of the OEM HMI package, and is saved under the BTSS_GR.HLP name (with German texts) in directory HLP.

Target systems

The use of the online help for variables is not restricted to OEM programming of the HMI user interface: It is also suitable for configuring the MMC 100 and the NC-Var selector in the PLC programming environment.

Functional scope

The online help for variables provides information on all NC variable; see also: Overview in Chapter 11 User Instructions "SINUMERIK 840D/840Di, SINUMERIK 810/FM-NC, HMI Programming Package Part 1" - and explanations in the List Manual/LIS/.

You can access the information about a variable by scrolling through several description levels:

- Starting with data area with:

  Data area - Block - Variable - Example

  - OR -

- Alphabetically sorted according to the block:

  Block - Variable - Example

  - OR -

- Finding keywords using the FIND function.

  Keywords include:

  – Brief description of the variables (e.g. spindle type)

  – Variable name (e.g. variable spindleType)

  – Brief designation/description of the block (e.g. SSP / for spindle status data).

Accept data

Parts can be copied from the displayed help topic, and inserted in other files. This is especially helpful when directly transferring the examples of the online help for variables into OEM programs.

To do this, proceed as follows:

- Select the "Edit" menu
- Select the "Copy" menu item
- Select the required text using the mouse
- Press the "Copy" button
- Change to other application
- Insert text.

Additional functions

Further, the online help for variables can be used as follows

- Print topics
- Insert comments relating to all topics
- Setting bookmarks to quickly find information that is frequently required.

Note

Comments and bookmarks

The comments for the online help for variables are saved in file BTSS_VAR.ANN (ANN is the abbreviation for Annex) - and the bookmarks in file WINHELP.BMK (BMK is the abbreviation for bookmark) in the Windows directory.

A.2.15 Troubleshooting

No connection to the NC/PLC

- Connecting cable
- MPI drivers must been installed
- Check MMC.INI
- WINSTART.BAT
- S7DPMPI.INI

...didn’t respond to DDE initiate

- Check Link Item
- Has the variable been created?

  Especially for PLC access operations: Has the data block been created?

Form load takes a long time as many hotlinks have to be established

- Use DCTL control
- Setup asynchronous hotlinks

Execute command does not function the first time

Cause

For some commands, the NCDDE server expects that a connection already exists to the NC.

Solution

First create a hotlink to an NC variable.

A.2.16 Determining active bus participants

In the control panel, open the "Set PG/PC interfaces" folder.

In the dialog window that is then displayed, in list "Interface parameterization used" select the active communication interface (e.g. CP5611 (MPI) <Active>) and press the "Diagnostics" button.

Using the "Test" button in the dialog subsequently shown, you can test the functioning of the selected communication interface.

Using the "Read" button, you can identify active bus participants.

From a program-related perspective, active bus participants can be identified by reading out variables /Nck/Nck/BusState.

A 32-bit value is supplied, where the set bits indicate an active bus participant (bit 0 set = active participant at bus address 0, bit 1 set = active participant at bus address 1, ...).

The functionality is available at the MPI and Profibus. The "lower" 32 addresses of the local bus segment are taken into consideration.

A.3 Error numbers/message

A.3.1 Error numbers

Overview

Table A- 74 Error code

Error number	Description
-70	Appears in the log file after expiration of a waiting time after an unsuccessful attempt to send
-97	RPC return value when an RPC arrives while starting.
-98	RPC return value when the RPC SINUMERIK internal order list is full, → wait and then repeat the call
-99	RPC return value when RPC is not supported. E.g. T_TPS_M is sent to a machine.
-100	RPC return value when machine name is incorrect.
-110	RPC return value when host name is incorrect.
-200	RPC return value when same function is already running with R_REPORT_H, when C_TPORDER_M arrives before the previous TPA is complete.
-203	R_REPORT_H when error occurs after R_NC4WPC_M
-250	R_REPORT_H when the file in the data management server cannot be deleted
-262	R_REPORT_H when the file in the data management cannot be loaded
-263	R_REPORT_H when the file in the data management cannot be loaded and selected
-264	R_REPORT_H when the file in the data management cannot be unloaded
-265	R_REPORT_H when the file in the data management cannot be selected
-266	R_REPORT_H when the file in the data management cannot be deselected
-270	R_REPORT_H when time/date cannot be updated
-271	R_REPORT_H when "Set protection level" has not worked
-272	R_REPORT_H when "Reset protection level" has not worked
-300	RPC return value when the file cannot be retrieved from the host computer
-301	R_REPORT_H when a long file name is shortened to 8.3
-302	R_REPORT_H when the file data is set with R_DATA_M (SFkt=1)
-310	R_REPORT_H following error in dh_create
-320	R_REPORT_H following error when program not in data management.
-400	RPC return value when the file cannot be transferred to the host computer
-500	R_REPORT_H when DDE-Connect error occurs for R_DDEDATA_M ()
-510	R_REPORT_H when DDE-Poke error occurs for R_DDEDATA_M ()
-600	R_REPORT_H following tool query with incorrect data structure number
-610	R_REPORT_H following tool query, read error in tool data
-700	R_REPORT_H when TPS acknowledges a TPA with error

Error number	Description
-800	R_REPORT_H when error occurs in T_VAR_M
-805	R_REPORT_H when error occurs in R_VAR_M
-810	R_REPORT_H when the variable set is unknown (SCVARSET.INI)
-820	R_REPORT_H when an error occurs in the variable set
-6003	1st R_REPORT_H after T_DATA_M (SFkt=10), Name1 probably wrong
-6020	R_REPORT_M after T_DATA_H (SFkt =21 ... 23) when the requested tool was not found.

SINUMERIK CMI error numbers

Table A- 75 Error numbers for SINUMERIK CMI

Error number	Description
-262	NC program could not be loaded in the EES memory
-264	NC program could not be unloaded from the EES memory
-265	NC program could not be selected in the EES memory
-266	NC program could not be deselected in the EES memory
-273	NC channels could not be determined
-275	TO areas could not be determined
-300	NC program could not be transferred to the PCU 50

A.3.2 "Orderlist full" message

"Orderlist full" message

The number of RPC calls to be processed simultaneously is limited. If too many RPC calls are sent by the host computer, RPC rejects further calls with return value "-98" (Orderlist full). In this case, the host computer must wait and the rejected RPC calls must be repeated after a certain amount of time.

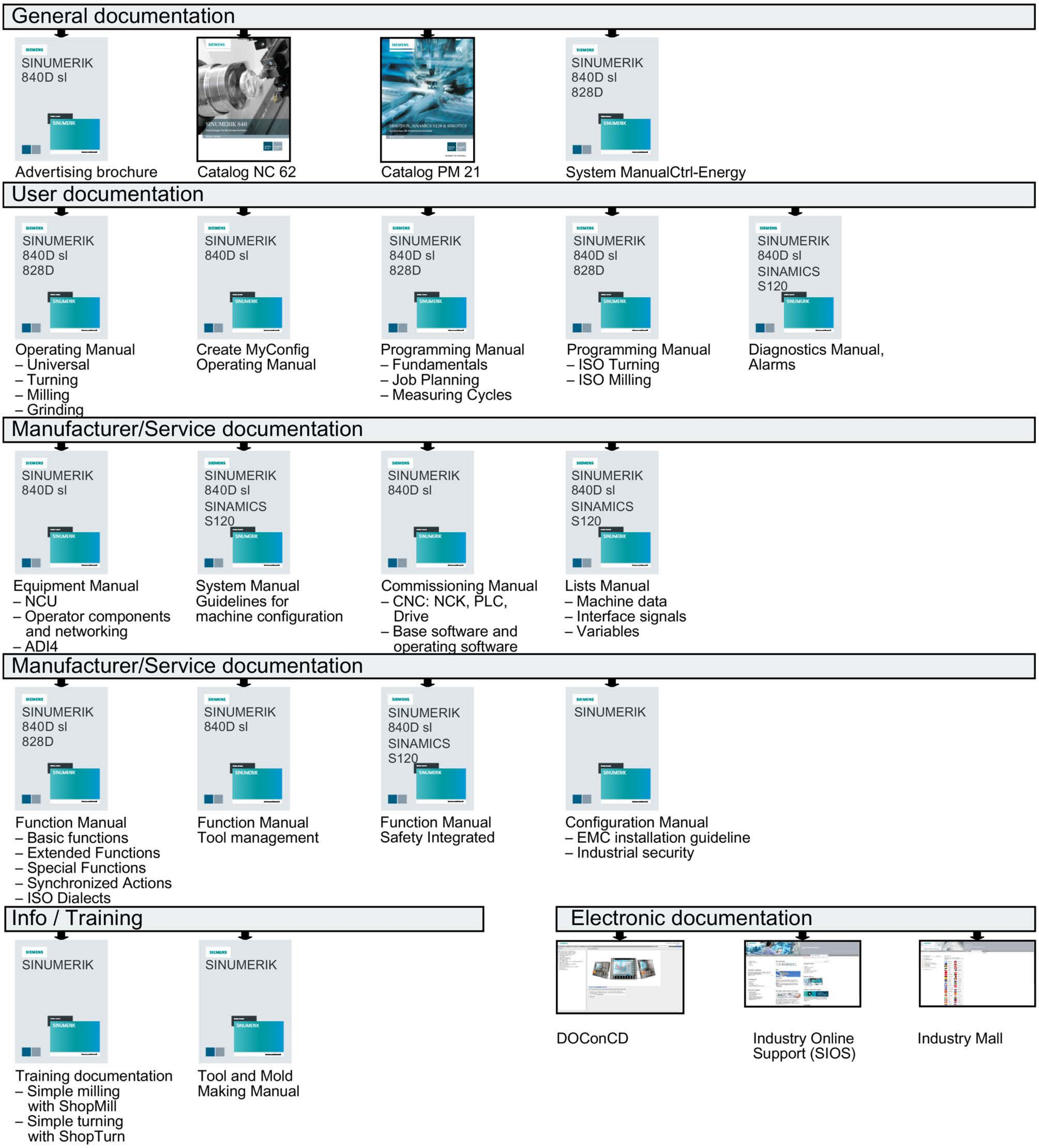
See also section: Error numbers (Page 348)

Note

RPC calls

- The calls must not be automatically resent immediately. This blocks the processing of calls that are still queued in the processing list.
- It is not permissible that more than five RPC calls per host are sent to the machine without response. Otherwise message is output:-98 "Order list full".

A.4 Documentation overview SINUMERIK 840D sl



General documentation

Advertising brochure

Catalog NC 62

Catalog PM 21

System ManualCtrl-Energy

User documentation

Operating Manual
– Universal
– Turning
– Milling
– Grinding

Create MyConfig
Operating Manual

Programming Manual
– Fundamentals
– Job Planning
– Measuring Cycles

Programming Manual
– ISO Turning
– ISO Milling

Diagnostics Manual,
Alarms

Manufacturer/Service documentation

Equipment Manual
– NCU
– Operator components
and networking
– ADI4

System Manual
Guidelines for
machine configuration

Commissioning Manual
– CNC: NCK, PLC,
Drive
– Base software and
operating software

Lists Manual
– Machine data
– Interface signals
– Variables

Manufacturer/Service documentation

Function Manual
– Basic functions
– Extended Functions
– Special Functions
– Synchronized Actions
– ISO Dialects

Function Manual
Tool management

Function Manual
Safety Integrated

Configuration Manual
– EMC installation guideline
– Industrial security

Info / Training

Training documentation
– Simple milling
with ShopMill
– Simple turning
with ShopTurn

Tool and Mold
Making Manual

Electronic documentation

DOConCD

Industry Online
Support (SIOS)

Industry Mall

A.5 List of abbreviations

Admin	Administrator (user role)
AMC	Analyze MyCondition
AMD	Access MyData
AMM/E	Access MyMachine/Ethernet
AMP	Analyze MyPerformance
AMT	Access MyTool ID
API	Application programming interface: interface to the application programming
ASC	Access MyMachine/Ethernet Service Client
ASCII	American Standard Code for Information Interchange
ASP	Application Service Provider-Variante
BA	Operating instructions
OI	Operator interface
CAM	Computer-Aided Manufacturing
CF-Card	CompactFlash Card: Memory card
CMI	Create MyInterface
CNC	Computerized Numerical Control:
CoL	Certificate of License
COM	Communication
COM	Component Object Protocol (programming model of Microsoft®)
CU	Control Unit
CP	Communication Processor
CPU	Central Processing Unit:
DB	Data Block (PLC)
DBB	Data Block Byte (PLC)
DBW	Data Block Word (PLC)
DCE	Distributed Computing Environment
DCOM	Distributed Component Object Model
Dbxy	Data Block (e.g. B 59)
DHCP	Dynamic Host Configuration Protocol: Dynamic assignment of an IP address and other configuration parameters on a computer in a network
DNC	Direct Numeric Control
DockPos	Docking Position
DIR	Directory:
DW	Data Word
EES	Execution from External Storage
FCC	File Client Cache
HC	Host computer
FTP	File Transfer Protocol
ET	Equability axis test
h	Hour
HMI	Human Machine Interface: SINUMERIK user interface

HD	Hard Disk: Hard disk
HTTP	Hypertext Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure,
HW	Hardware
IAC	InterActive Client
IB	Commissioning engineer (user role)
IE	Internet Explorer
IFC	Interface Client
IH	Maintenance
IIS	Microsoft Internet Information Service
CT	Circularity Test
MB	Megabyte
MCIS	Motion Control Information System
MCIS MDA	MCIS Machine Data Aquisition: Machine data management
MCP	Machine Control Panel:
MD	Machine Data
MHComm	Machine Handler Communication
MCS	Machine Coordinate System
MLFB	Machine-Readable Product Code
MMP	Manage MyPrograms
MMP IFC	Manage MyPrograms InterFace Client
MMC	Men-Machine-Communication, Man-machine communication
MMT	Manage MyTools
MO	Machine Operator: Machine operator (user role)
MPI	Multi Port Interface: Multiport Interface
MRL	Manufacturing Resource Library
MS	Microsoft
MSDE	Microsoft Data Engine or Microsoft Desktop Engine database software
MSI	Microsoft Software Installation
MSTT	Machine control panel
NC	Numerical Control: Numerical control
NCK	Numerical Control Kernel: Numeric kernel with block preparation
NCU	Numerical Control Unit: NCK hardware unit
NFS	Network File System
NTLM	NT LAN Manager: Authentication procedure for computer networks
OB	Organization block in the PLC
ODBC	Open Database Connectivity
OEM	Original Equipment Manufacturer
OLE	Object Linking & Embedding
OP	Operation Panel: Operating equipment
OPI	Operation Panel Interface: Interface for connection to the operator panel
OPC	OLE for Process Control
PC	Personal Computer

PCMCIA	Personal Computer Memory Card International Association (memory card standardization)
PCU	PC Unit: Computer unit
PG	Programming device
PDA	Production Data Acquisition: Production data acquisition
PLC	Programmable Logic Control: PLC
PMT	Parts Monitoring & Tracking: Parts tracking
PO	POWER ON
ProToolPro	Configuration tool
PWS	Personal Web Server
RFC	Requests for Comments
RCS	Computer coupling software in MMC102, MMC103
ROM	Read Only Memory
RPC	Remote Procedure Call, function call via network connection
SC	Service Client
SE	Service Engineer
SEC	Service Engineer Client
SEG	Tool presetter
SFI or SFI RM	Shop Floor Integrate or Shop Floor Integrate Resource Management
SI	SINUMERIK Integrate
SK	Softkey
SSL	Secure Socket Layer
SW	Software
TDI	Tool Data Information
TC	Teamcenter
TCP/IP	Transmission Control Protocol / Internet Protocol
TLCA	Tool Library CA: Tool catalog data
TLCU	Tool Library Customer: Customer tool catalog data
TLCUA	Tool Library Customer Assemblies: Complete tools
TPS	Transport system
UAT	Universal Axis Test
UNC	Uniform naming convention
VB	Visual Basic
VC	Visual C
VNCK	Virtual numerical control core
WCS	Workpiece Coordinate System
T	Tool
TLC	Tool Length Compensation
WOP	Workshop-Oriented Programming
WPC	Workpiece Carrier, workpiece holder
WPD	Work Piece Directory: Workpiece directory
TRC	Tool Radius Compensation
WSDL	Web Services Description Language

T	Tool
TO	Tool Offset
MT	Machine Tool
TM	Tool Management
TC	Tool Change
XML	Extensible Markup Language