SIEB & MEYER

CNC 8x.00

SMDDE

Communication between CNC and external applications

P-TD-0000310.5
2017-08-23

Copyright

Translation of the original instructions, Copyright © 2017 SIEB & MEYER AG

All rights reserved.

This manual or extracts thereof may only be copied with the explicit authorization of SIEB & MEYER AG.

Trademarks

All product, font and company names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

SIEB & MEYER worldwide

For questions regarding our products and technical problems please contact us.

SIEB & MEYER AG
Auf dem Schmaarkamp 21
21339 Lueneburg
Germany

Phone: +49 4131 203 0
Fax: +49 4131 203 2000
support@sieb-meyer.de
http://www.sieb-meyer.com

SIEB & MEYER Shenzhen Trading Co. Ltd.
Room A208, 2/F,
Internet Innovation and Creation services base
Building (2),
No.126, Wanxia road, Shekou, Nanshan district,
Shenzhen City, 518067
China

Phone: +86 755 2681 1417 / +86 755 2681 2487
Fax: +86 755 2681 2967
sm.china.support@gmail.com
http://www.sieb-meyer.cn

SIEB & MEYER Asia Co. Ltd.
4 Fl, No. 532, Sec. 1
Min-Sheng N. Road
Kwei-Shan Hsiang
333 Tao-Yuan Hsien
Taiwan

Phone: +886 3 311 5560
Fax: +886 3 322 1224
smasia@ms42.hinet.net
http://www.sieb-meyer.com

SIEB & MEYER USA
3975 Port Union Road
Fairfield, OH 45014
USA

Phone: +1 513 563 0860
Fax: +1 513 563 7576
info@sieb-meyerusa.com
http://www.sieb-meyer.com

1 Introduction

The CNC allows execution of certain actions by an external application. Instructions can for example be transferred to the CNC or values can be inquired by different data objects.

⚠ **NOTICE**

Damage to the machine / wrong production results

The uncontrolled remote control of the machine can cause damage to the machine!

Always ensure clear states of the CNC and the machine when commands are transmitted.

This manual describes communication commands, command objects and data objects that allow data exchange with the CNC.

- These functionalities allow for example inquiring events or transmitting instructions to the CNC.
- The connection to the CNC can be realized via different transmission protocols (see chapter 2 "Data Transmission", page 9).

☞ For necessary settings in the used CNC, consider the corresponding CNC manuals. In a CNC 8x.00, for example, the CNC command NONDDE must be active for correct receipt of commands.

2 Data Transmission

The connection to the CNC software can be realized via different transmission protocols.

2.1 Transmission Protocols

The CNC supports the common transmission protocols.

DDE (Dynamic Data Exchange)

The DDE-protocol is locally available in the operating systems Windows (from version 2.0) and OS/2.

- Commands and data can be exchanged via DDE, if both application programs are running.
- The application that requires data (client) sends a request to the application that provides data (server = CNC).
- The data provided this way are permanently linked with the server. This ensures that the client application has access to the current data at any time.
- All commands and data are transmitted as ASCII text.
- The CNC 4x.00 (CNC 46.00, CNC 48.00 and SYSTEM 56.00) supports one running connection.
- The CNC 8x.00 (CNC 82.00 and CNC 84.00) supports up to 4 running connections.
- The usual commands EXECUTE, REQUEST, ADVISESTART, ADVISESTOP and ADVISE are available as communication commands (see chapter 3 "Communication Commands (commands)", page 17).

NetDDE

The NetDDE protocol corresponds to the local DDE protocol (see above). Only communication is realized via the network.

- The installation of the DDE interface in a network is described in the manual "*Software Configuration*".
- The information for the DDE protocol apply.

TCP/IP (Transmission Control Protocol/Internet Protocol)

Since the TCP/IP protocol does not require a certain operating system, it should be preferred to the DDE protocol.

- The computers participating in a network are identified via IP addresses.
- The connection is established and interrupted according to standards. The content of the protocol also meets the standards.
- The transmission consists of XML strings, only.
- The usual commands EXECUTE, REQUEST, ADVISESTART, ADVISESTOP and ADVISE are available as communication commands (see chapter 3 "Communication Commands (commands)", page 17).

2.2 Overview

The following table lists the available transmission commands and objects:

Terms and Definitions	Meaning
Service Only DDE	SMSERVER
Devices (topics) Only DDE	CNC1, CNC2, CNC3, CNC4, CNC5, CNC6, CNC7, CNC8
Communication Commands (commands)	"ADVISE: Transmit Content of a CNC Object", page 18 "ADVISESTART: Start Automatic Advice Note", page 19 "ADVISESTOP: Stop Automatic Advice Note", page 20 "EXECUTE: Execute Statement", page 21 "REQUEST: Transmit Request", page 22
Command objects (command items)	"CHANGECLIENT: Switch Screen Page", page 26 "CNCCOMMAND: Execute CNC Command", page 28 "CNCKEY: CNC Keystroke", page 29 "CLRNEXT: Clear Program Name", page 27 "COMMAND: Transmit Command Object", page 30 "INFLGCLR: Clear Loader Input Flags", page 31 "INFLGSET: Set Loader Input Flags", page 31 "PCCOMMAND: Execute PC Command", page 34 "PCKEY: Transmit PC Keystroke", page 33 "PROGRAM: Transfer File Name", page 35 "RUNTIMEVALUE: Current Value of an Object", page 37 "SETHANDLE: Define Data Object", page 38 "STARTEXE: Start Application", page 39
Data objects (data items)	"@: Set Data Object for CNC Handle", page 42 "ACTPROGRAM: File Name of the Current Program", page 43 "CNCERROR: DNC Message", page 44 "CNCSTATUS: CNC Status", page 45 "CNCTOOLS: Current Tool Data", page 46 "COMMSTATUS: Execution Status of a CNC Command", page 47 "DATETIME: Current Date and Time", page 48 "DNCM: Current Mode of the DNCM Command", page 48 "DUTY: Load of CNC", page 49 "FTASTEN: State of the Function Keys", page 50 "INFLAGS: State of the Loader Input Flags", page 50 "NEXTPROGRAM: File Name of Next Part Program", page 51 "OUTFLAGS: State of Loader Output Flags", page 53 "PCSTATUS: Status of Communication Command", page 54 "SCREENSAVER: Current Event Message", page 54 "SYSSTATUS: Status of the Last Communication", page 55 "USERLEVEL: Current Access Authorization", page 56 "VERSION: SIEB & MEYER TCP/IP Version", page 57 "XYPOSITION: Current Table Position", page 58

2.3 TCP/IP Transmission Structure

Communication is realized via ASCII format, only.

- The communication process always includes a request package and a response package.
- To acknowledge the receipt of a request package the response package may be identical with the request package.
- If the value of a data object is requested, the response package already includes the data value.
- If, however, the CNC receives an unknown data object, the response package includes the unknown data object and an empty data value (""). A corresponding check of the response package can avoid the waiting time, until the timeout time is passed.

The content of the communication package consists of XML elements

- XML elements are enclosed by angle brackets ("<…>").
- Any XML element consists of one pair of elements. The closing element is labeled by a slash ("</…>").
- XML elements consist of the element name (e.g. "<EXECUTE>") and optional attributes (e.g. "<EXECUTE Value=…>").Attributes are enclosed by quotation marks.

The communication always consists of two transmission packages.

Package of the sender:

```
<SMDNCPACKET Value="packetId">
  <CNC Value="cncNumber">
    <communication command Value="communicationId">
      <item Value="item value">
        item data
      </item>
    </communication command>
  </CNC>
</SMDNCPACKET>
```

Response package of the receiver:

```
<SMDNCPACKET Value="packetId">
  <CNC Value="cncNumber">
    <communication command Value="communicationId">
      <item Value="item value">
        itemData
      </item>
    </communication command>
  </CNC>
</SMDNCPACKET>
```

Element/value	Explanation
<SMDNCPACKET>	Root element
packetId	The attribute “Value” transmits an identification number (*packetId*) as value. Specific notes: ▸ The sender and receiver manage different identification numbers. ▸ Identification numbers can be used for monitoring the order of received packages, for example. ▸ The CNC does not evaluate the identification number.

Element/value	Explanation
	▸ The CNC generates a serial number as identification number for every package. ▸ Note: There will be gaps in the serial numbers, if several client processes are running.
<CNC>	CNC address element
cncNumber	The attribute “Value” transmits the number of the addressed CNC (default = 1).
<*communication command*>	This element includes the communication command. The communication command defines the content of the communication package. Depending on the communication command this package must be programmed in the element *item* of a command object (*command item*) or data object (*data item*).
communicationId	The attribute “Value” transmits a definite number (*communicationId*). Specific notes: ▸ Response packages can only be identified definitely via this command ID. ▸ The ID will be returned by the CNC in a response package. ▸ The ID can be structured individually. ▸ Example: The ID consists of a code number for the command (01=EXECUTE, 02=REQUEST etc.) and a serial number (4-digit): Following this rule, the number for the 15th EXECUTE instruction is "010015".
<*item*>	The element *item* includes a command object (*command item*) or a data object (*data item*). ▸ *command item*: This element is used with the communication command EXECUTE, only. ▸ *data item*: This element is used with the communication commands ADVISE, ADVISESTART, ADVISESTOP und REQUEST, only.
item value	The attribute *item value* can include a value. Example: The CNC transmits by default the ADVISE state of a data object.
item data	Transmission value or return value of a data object

For further support, please contact the SIEB & MEYER service (e-mail: cnc-service@sieb-meyer.de).

Example: Requesting CNC data

In this example the client requests the name of the current production program.

Request package of the client

- The package is generated by the client (in this case: 124).
- The CNC address is "1".
- The REQUEST command requests the value of the data package ACTPROGRAM. The command ID is generated by the client (in this case: 020327). This ID is returned back by the CNC in the response package allowing exact assignment to the request later.
- The data object ACTPROGRAM requests the name of the current production program.

```
<SMDNCPACKET Value="124">
 <CNC Value="1">
  <REQUEST Value="020327">
   <ACTPROGRAM>
   </ACTPROGRAM>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The response package serves as acknowledgement of receipt. The structure corresponds to the structure of the request package.
- The package ID is generated by the CNC (in this case: 332553).
- The REQUEST command includes the ID generated by the client (in this case: 020327). This ensures correct assignment to the request.
- The Value attribute of the data object ACTPROGRAM includes the state of the automatic advice note (in this case: ADVISEOFF). The value of the data object includes the requested file name.

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <REQUEST Value="020327">
   <ACTPROGRAM Value="ADVISEOFF">
    C:\SM_WPROG\DRILL.SM5
   </ACTPROGRAM>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

2.4 Connection of the Camera

Requirement

- *Software version 10.01 or higher*

This section describes how to connect an external camera system.

- The syntax of the transmission commands is described in the manual “Optical Measurement - Protocol”.
- The measuring commands are described in the manual “Optical Measurement - Operation”.

2.4.1 Transmission Diagram

All transmission processes are triggered only by the CNC.

- The server is always the camera system.
- The client is always the CNC.
- In case of an error the transmission of a package is repeated up to 4 times. If the transmission is still not correct, the process is canceled.
- In a CNC 8x.00 the used interface is reserved only during one transmission.
- In case of an error the transmission of a package is repeated up to 4 times. If the transmission is still not correct, the process is canceled.
- In CNC 4x.00 two waiting times (timeouts) are distinguished:
  - waiting time while transmitting the package = 2 s (e.g. wait for “ACK”). this waiting time can be set with the CNC command CATMt.
  - waiting time for measuring response = endless (can only be canceled with Escape key)
- A CNC 8x.00 uses a waiting time (Timeout) for the following cases (adjustable with the CNC command CATM*t*, default value = 600 s):
  - waiting time while transmitting the package
  - waiting time for measuring response
  - waiting time for establishing the connection = double CATM duration (only for TCP/IP)
- Length values (positions, measurement results, depth values, diameter values and radius values) are transmitted as micrometers (µm). Example: The value 23.45 mm is transmitted as 23450.
- Transferring parameters to the camera system is triggered
  - by the M75 command in the part program (e.g. M75,ABC)
  - by the sequence command SEND CAM in a sequence (only CNC 8x.00)
- The CNC uses the structure "Send strong" for the transmission.
- For analyzing purposes, the data transmission is additionally provided via the output debug string interface (CNC 8x.00).

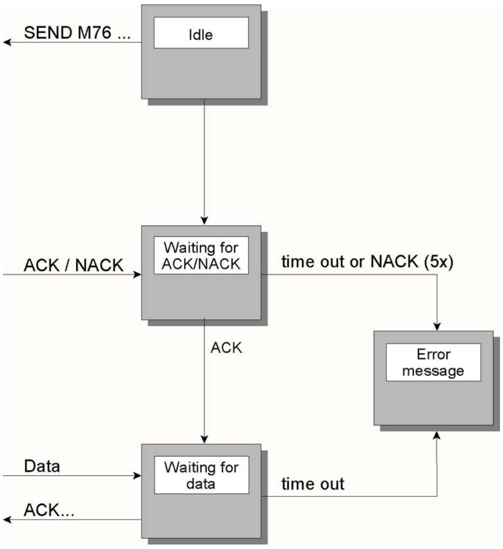


Fig. 1: Schematic diagram of a transmission

The figure shows the page for transmitting the M76 command.

- *After sending the data, the client waits for an acknowledgement of receipt (ACK).*
- *Afterwards, the client waits for a response package including the M76 command.*
- *Finally, the client sends an acknowledgement of receipt to the server (ACK).*

2.4.2 TCP/IP (Camera System)

Data are transmitted via a network interface.

- TCP/IP is only supported by the CNC 8x.00.
- The connection is configured by the machine manufacturer in the parameter editor
- If several camera systems are connected (e.g. machine with several independent work areas), a separate connection must be configured for every system.
- Machine parameter IPADR = TCP/IP address. The address can be taken from the network connection settings of the computer on which you have installed the camera software. If the computer is a CNC computer, use the address “127.0.0.1”.

- Machine parameter PORT = port number. A separate port must be selected for every camera system! If, for example, two cameras are installed in one machine, they are identified via the port number when communicating with the camera system.

2.4.3 Serial Connection

Data are transmitted via a serial interface. Specific notes:

- The connection is configured by the machine manufacturer in the parameter editor
- If several camera systems are connected (e.g. machine with several independent work areas), a separate connection must be configured for every system.
- Default setting: 9600 baud, 8 data bits, 1 stop bit, no parity bit.
- A hardware handshake with RTS/CTS is required.

3 Communication Commands (commands)

A communication command (command) transmits a communication object to the CNC (command object or data object).

Requirement

To allow communication with the CNC ensure that the receipt of external statements (CNC command NONDDE).

A communication object defines the function and the content of a data package.The table lists all available communication objects (in alphabetical order). The column “Object” specifies the type of the data object.

- For details on the command objects refer to chapter 4 "Command Objects (command items)", page 25.
- For details on the data objects refer to chapter 5 "Data Objects (data items)", page 41.

Command	Object	Explanation
ADVISE	Data object	The CNC sends the content of a data object to the client, because the value of the data object has changed.
ADVISESTART	Data object	Statement to the CNC to activate the automatic advice note for a data object.When the content of the data object changes, the CNC sends an appropriate advice note.
ADVISESTOP	Data object	Statement to the CNC to deactivate the automatic advice note for a data object.
EXECUTE	Command object	The client sends a statement to the CNC.
REQUEST	Data object	The client requests a data value from the CNC.

Example: requesting a CNC data object via TCP/IP

In this example the client requests the name of the current production program.

Request package of the client

```
<SMDNCPACKET Value="124">
 <CNC Value="1">
  <REQUEST Value="020327">
   <ACTPROGRAM>
   </ACTPROGRAM>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The response package serves as acknowledgement of receipt. The structure corresponds to the structure of the request package.
- The acknowledgement already includes the requested value. In this case the file name.
- The state of the automatic advice note of the CNC object is returned in the data object (in this case: <ACTPROGRAM Value="ADVISEOFF">).

```
<SMDNCPACKET Value="332553">
```

```
  <CNC Value="1">
    <REQUEST Value="020327">
      <ACTPROGRAM Value="ADVISEOFF">
        C:\SM_WPROG\DRILL.SM5
      </ACTPROGRAM>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

ADVISE: Transmit Content of a CNC Object

Transmit content of a CNC object

Command	Description	Version
ADVISE *data item*	The CNC sends the value of a CNC object.	10.08

Argument	Description
data item	Data object

The communication command ADVISE transmits the value of a CNC object for which the automatic advice note is activated. Specific notes for TCP/IP:

- The communication command <ADVISE> includes the attribute “Value”. The value is a number for the definite identification of the statement.
- The data object includes the value of the monitored CNC object and the state of the automatic advice note (Value=ADVISEON).

Example (TCP/IP)

In this example the state of the CNC status has changed. Since the automatic advice note is activated, the CNC starts the transmission.

Automatic advice note of the CNC:

```
<SMDNCPACKET Value="725483">
  <CNC Value=1>
    <ADVISE Value="050824">
      <CNCSTATUS Value="ADVISEON">
        AR00:07:35,AH001600,AP100,ZS00000001,MOWAIT,EC0048;3378
      </CNCSTATUS>
    </ADVISESTART>
  </CNC>
</SMDNCPACKET>
```

Response package of the client:

```
<SMDNCPACKET Value="123">
  <CNC Value=1>
    <ADVISE Value="050824">
      <CNCSTATUS Value="ADVISEON">
        AR00:07:35,AH001600,AP100,ZS00000001,MOWAIT,EC0048;3378
      </CNCSTATUS>
    </ADVISE>
  </CNC>
</SMDNCPACKET>
```

Related topics

SMDDE object "ADVISESTART: Start Automatic Advice Note", page 19

SMDDE object "ADVISESTOP: Stop Automatic Advice Note", page 20

ADVISESTART: Start Automatic Advice Note

Start automatic advice note

Command	Description	Version
ADVISESTART *data item*	Activate automatic advice note for a CNC object.	10.08

Argument	Description
data item	Data object

The communication command ADVISESTART instructs the CNC to activate the automatic advice note for a data object. Specific notes for TCP/IP:

- The communication command ADVISESTART includes the attribute “Value”. The value is a number for the definite identification of the statement.
- The request package includes an empty data object.
- The attribute “Value” of the data object in the response package includes the new state of the automatic advice note (Value="ADVISEON").

Example (TCP/IP)

In this example the CNC shall send an advice note whenever the CNC status changes.

Request package of the client

```
<SMDNCPACKET Value=125>
  <CNC Value=1>
    <ADVISESTART Value=030328>
      <CNCSTATUS>
      </CNCSTATUS>
    </ADVISESTART>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value=332553>
  <CNC Value=1>
    <ADVISESTART Value=030328>
      <CNCSTATUS Value="ADVISEON">
      </CNCSTATUS>
    </ADVISESTART>
  </CNC>
</SMDNCPACKET>
```

When the CNCSTATUS changes the CNC sends an ADVISE package to the client. Every ADVISE package must be acknowledged by the client.

Related topics

SMDDE object "ADVISE: Transmit Content of a CNC Object", page 18

SMDDE object "ADVISESTOP: Stop Automatic Advice Note", page 20

ADVISESTOP: Stop Automatic Advice Note

Stop automatic advice note

Command	Description	Version
ADVISESTOP *data item*	Deactivate automatic advice note for a CNC object	10.08

Argument	Description
data item	Data object

The communication command ADVISESTOP instructs the CNC to deactivate the automatic advice note for a data object.

- The communication command ADVISESTART includes the attribute “Value”. The value is a number for the definite identification of the statement.
- The request package includes an empty data object.
- The attribute “Value” of the data object in the response package includes the new state of the automatic advice note (Value="ADVISEON").

Example (TCP/IP)

In this example the CNC shall not send an advice note whenever the CNC status changes.

Request package of the client

```
<SMDNCPACKET Value="126">
  <CNC Value=1>
    <ADVISESTOP Value="040329">
      <CNCSTATUS>
      </CNCSTATUS>
    </ADVISESTOP>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332553">
  <CNC Value=1>
    <ADVISESTOP Value="040329">
      <CNCSTATUS Value="ADVISEOFF">
      </CNCSTATUS>
    </ADVISESTOP>
  </CNC>
</SMDNCPACKET>
```

Related topics

SMDDE object "ADVISE: Transmit Content of a CNC Object", page 18

SMDDE object "ADVISESTART: Start Automatic Advice Note", page 19

EXECUTE: Execute Statement

Execute statement

Command	Description	Version
EXECUTE *command item*	Statement to the CNC	10.08

Argument	Description
command item	Statement

The communication command EXECUTE transmits an statement to the CNC.

- Requirement: The CNC must allow the receipt of external instructions (CNC command NONDDE).
- TCP/IP: The communication comand EXECUTE includes the attribute “Value”. The value is a number for the definite identification of the statement.
- All statements that can be executed with the communication comand EXECUTE are described in section chapter 4 "Command Objects (command items)", page 25

Example (TCP/IP)

In the example the absolute traveling plane is set to 13.0 mm. Requirement: The command execution via TCP/IP is enabled in the CNC (CNC command NONDDE). After the package has been transmitted, it is checked, whether or not the command has been executed.

Request package of the client

```
<SMDNCPACKET Value="123">
 <CNC Value="1">
  <EXECUTE Value="010326">
   <CNCCOMMAND>
    H13.
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The acknowledgement only signals that the command has been received correctly.
- It does not include any information on whether the command has already been executed.

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <EXECUTE Value="010326">
   <CNCCOMMAND Value="83">
    H13.
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Request package of the client

```
<SMDNCPACKET Value="124">
  <CNC Value="1">
    <REQUEST Value="020327">
      <COMMSTATUS>
      </COMMSTATUS>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

The response package of the CNC includes the data value (in this case: "OK")

```
<SMDNCPACKET Value="332554">
  <CNC Value="1">
    <REQUEST Value="020327">
      <COMMSTATUS Value="ADVISEOFF">
        OK
      </COMMSTATUS>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

REQUEST: Transmit Request

Transmit request

Command	Description	Version
REQUEST *data item*	Request data value of a CNC object.	10.08

Argument	Description
data item	Data object

The communication command REQUEST transmits a request to the CNC. Specific notes for TCP/IP:

- The communication comand EXECUTE includes the attribute “Value”. The value is a number for the definite identification of the statement.
- The request package includes an empty data object.
- In the response package the data object includes the data value and the attribute “Value” includes the state of the automatic advice note (e.g. Value=ADVISEOFF).

Example (TCP/IP)

In the example the absolute traveling plane is set to 13.0 mm. Requirement: The command execution via TCP/IP is enabled in the CNC (CNC command NONDDE). After command transmission, the CNC checks, whether or not the command has been executed.

Request package of the client

```
<SMDNCPACKET Value="123">
  <CNC Value="1">
    <EXECUTE Value="010326">
      <CNCCOMMAND>
        H13.
```

```
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The acknowledgement only signals that the command has been received correctly.
- It does not include any information on whether the command has already been executed.

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <EXECUTE Value="010326">
   <CNCCOMMAND Value="83">
    H13.
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Request package of the client

```
<SMDNCPACKET Value="124">
 <CNC Value="1">
  <REQUEST Value="020327">
   <COMMSTATUS>
   </COMMSTATUS>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

The response package of the CNC includes the data value (in this case: "OK")

```
<SMDNCPACKET Value="332554">
 <CNC Value="1">
  <REQUEST Value="020327">
   <COMMSTATUS Value="ADVISEOFF">
    OK
   </COMMSTATUS>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

4 Command Objects (command items)

Command objects (command items) are statements to the CNC for executing an action.

Requirement

To allow communication with the CNC ensure that the receipt of external statements (CNC command NONDDE).

Command objects can be programmed with the communication command EXECUTE, only. For details on the command objects refer to chapter 4 "Command Objects (command items)", page 25.

Name	Content
EXECUTE	The client sends a statement to the CNC.

Specific notes for TCP/IP:

- The request package includes the command object and an optional transmission value.
- The response package is identical to the request package.
- The transmission parameter is of the data type STRING.
- The data object “SysStatus” includes the information, whether the transmission an the command system of a transmitted command are correct.
- The result is returned in a separate data package. For more detailed information refer to the description of the appropriate command.
- For examples, refer to the description of the appropriate command and the section chapter 6 "Examples", page 59.

Example: execution of a CNC command (TCP/IP)

In this example the CNC shall execute the CNC command "H13.".

Command package of the client

```
<SMDNCPACKET Value="124">
 <CNC Value="1">
  <EXECUTE Value="010327">
      <CNCCOMMAND>
    H13.
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The response package serves as acknowledgment of receipt. The XML structure corresponds to the request package.
- The transmitted command is included in the acknowledgement.

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <EXECUTE Value="010327">
```

```
   <CNCCOMMAND>
    H13.
   </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

CHANGECLIENT: Switch Screen Page

Switch screen page

Command	Description	Version
CHANGECLIENT *page*	Switch page in the CNC user interface	10.08

Argument	Description
page	Name of screen page

Return values

Return value	Explanation
SYSSTATUS	Information, whether the command was transmitted correctly
PCSTATUS = 0	Screen page was switched
PCSTATUS = 1	Screen page was not switched

The command object CHANGECLIENT displays another screen page of the CNC.

- The data object PCSTATUS includes the information, whether the command was executed correctly.
- You will find a selection of the CNC screen pages in section 7.C.3 "Names of CNC Screen Pages", page 93.

Example (TCP/IP)

The dynamic tools page shall be displayed. After sending a command, the data objects are set by the CNC.

Switching takes two steps: First, the 1st tool page (TOOLS) is displayed. Then, it is switched to the desired page (TOOLSDYN).

Request package of the client

`<EXECUTE><CHANGECLIENT>TOOLS</CHANGECLIENT></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><CHANGECLIENT>TOOLSDYN</CHANGECLIENT></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Related topics

SMDDE object "PCSTATUS: Status of Communication Command", page 54

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

CLRNEXT: Clear Program Name

Clear program name

Command	Description	Version
CLRNEXT	Clear the name in the buffer.	10.08

Argument	Description
–	No transfer parameter required.

Return values

Return value	Explanation
SYSSTATUS	Information, whether the command has been transmitted correctly.
NEXTPROGRAM	File name of the next part program

The command object CLRNEXT clears the entry of the next part program in the CNC.

Example (TCP/IP)

The name of the next part program is transmitted to the buffer memory of the CNC. If the operator activates a certain machine function, the part program shall be removed from the buffer storage.

`<ADVISESTART><FTASTEN></FTASTEN></ADVISESTART>`	Activate monitoring of the FTASTEN string.
`<EXECUTE><PROGRAM>ABC.SM5</PROGRAM></EXECUTE>`	Load part program into the main memory.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Automatic advice note of the CNC after a function key was pressed. The position of the number "1" identifies the number of the function key.

<ADVISE><FTASTEN>0000100000100000</FTASTEN></ADVISE>	Advise note of the CNC

`<ADVISE><FTASTEN>0000100000100000</FTASTEN></ADVISE>`	Confirm ADVISE message.

The statement to clear the part program from the memory follows the evaluation of the string.

`<EXECUTE><CLRNEXT></CLRNEXT></EXECUTE>`	Clear the part program from the memory.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Related topics

CNC command "FTASTEN: State of the Function Keys", page 50

CNC command "NEXTPROGRAM: File Name of Next Part Program", page 51

CNC command "PROGRAM: Transfer File Name", page 35

CNCCOMMAND: Execute CNC Command

Execute CNC command

Command	Description	Version
CNCCOMMAND *cmd*	Execute CNC command in the CNC.	10.08

Argument	Description
cmd	Name of the CNC command

Return values

The execution state provides information on the command execution.

Data objects	Value	Explanation
SYSSTATUS		Information, whether the command was transmitted correctly (see data object SYSSTATUS).
COMMSTATUS	BUSY	The CNC command is being executed.
	OK	The execution of the CNC command is finished.
	ERROR	The execution of the CNC command caused an error.

The command objects CNCCOMMAND executes a CNC command in the CNC. The string "@@@" suppresses possible inquiries.

Example: Move the machine axes and show tool parameters.

In this example the Z-axes are retracted and then, the XY-axes are moved to the park position. During positioning, the CNC user interface is switched to the page “Tool parameters (dyn.)”.

- To get an automatic advice note (ADVISE) when the park position is reached, ADVISESTART must be activated for the data object “CommStatus”.
- During the positioning the data object COMMSTATUS is BUSY
- After the park position has been reached, the data object COMMSTATUS is OK.

`<ADVISESTART><COMMSTATUS></COMMSTATUS></ADVISESTART>`	Monitor COMMSTATUS.
`<EXECUTE><CNCCOMMAND>H</CNCCOMMAND></EXECUTE>`	CNC command for Z-axis retraction
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`...`	Wait, until the Z-axes have reached the target position (ADVISE[COMMSTATUS=OK])
`<EXECUTE><CNCCOMMAND>P</CNCCOMMAND></EXECUTE>`	CNC command for XY-axis positioning
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><CHANGECLIENT>TOOLS</CHANGECLIENT></EXECUTE>`	CNC command for Z-axis retraction
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><CHANGECLIENT>TOOLSDYN</CHANGECLIENT></EXECUTE>`	CNC command for Z-axis retraction
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`...`	Wait, until the XY-axes have reached the target position (ADVISE[COMMSTATUS=OK])

Related topics

"COMMSTATUS: Execution Status of a CNC Command", page 47

"SYSSTATUS: Status of the Last Communication", page 55

CNCKEY: CNC Keystroke

CNC Keystroke

Command	Description	Version
CNCKEY *button*	Simulate keystroke in the CNC.	10.08

Argument	Description
button	Keyname

Return values

Return value	Explanation
SYSSTATUS	Information, whether the command was transmitted correctly.

The command object CNCKEY triggers a keystroke in the CNC An overview of all keynames is provided in section 7.C.4 "CNC Keynames", page 95.

Example: interrupt execution

Simulate STOP key in the CNC.

`<EXECUTE><CNCKEY>STOP</CNCKEY></EXECUTE>`	Execute CNC keystroke.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Example: automatic user log-in

In this example the user is automatically logged in the CNC. For this purpose, several dialog boxes are confirmed in the page. After login, the page “Work [Work - Status]” appears and the execution is started.

`<EXECUTE><CHANGECLIENT>LOGIN</CHANGECLIENT></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><PCKEY>L</PCKEY></EXECUTE>`	Keystroke "L" (call log off dialog)
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Check, whether the keystroke was executed.
`<EXECUTE><PCKEY>J</PCKEY></EXECUTE>`	Keystroke "J" (confirm dialog)
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Check, whether the keystroke was executed.
`<EXECUTE><PCKEY>USER\n</PCKEY></EXECUTE>`	Transfer user name (\n=acknowledgement)

`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Check, whether the text was entered.
`<EXECUTE><PCKEY>****\n</PCKEY></EXECUTE>`	Transfer secret word (\n=acknowledgement)
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Check, whether the text was entered.
`<EXECUTE><CHANGECLIENT>DEFAULT</CHANGECLIENT></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><CNCKEY>START</CNCKEY></EXECUTE>`	START key
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Related topics

SMDDE object "CHANGECLIENT: Switch Screen Page", page 26

SMDDE object "PCKEY: Transmit PC Keystroke", page 33

SMDDE object "PCSTATUS: Status of Communication Command", page 54

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

COMMAND: Transmit Command Object

Transmit command object

Command	Description	Version
COMMAND *cmd*	Transmit command object	10.08

Argument	Description
cmd	Command object with parameter

The command object COMMAND is a universal command object, used for transmitting another command object.

- The command object COMMAND includes names and parameters of the command object to be transmitted.
- A potentially necessary parameter is separated by an equal sign.

Example (TCP/IP)

The dynamic tools page shall be displayed. After the command has been sent, the data objects are set by the CNC.

`<EXECUTE><COMMAND>CHANGECLIENT=TOOLS</COMMAND></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, if the command has been received
`<EXECUTE><COMMAND>CHANGECLIENT=TOOLSDYN</COMMAND></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, if the command has been received

INFLGCLR: Clear Loader Input Flags

Clear loader input flags.

Command	Description	Version
INFLGCLR *flags*	Clear loader input flags.	10.08

Argument	Description
flags	Flag list

Return values

Return value	Explanation
SYSSTATUS	State of the command execution
INFLAGS	State of the loader input flags

The command object INFLGSET clears one or several loader input flags.

- The flag states are defined in a string.
- CNC 4x.00: 32 flags (the string must have 32 signs)
- CNC 8x.00: 64 flags (the string must have 32 or 64 signs)
- The 1st sign of the string influences the flag LIFL1.
- The last sign of the string influences the highest significant flag.
- Every sign defines the state of a flag.
  - Sign = "1": The bit is cleared.
  - Sign = "0": The bit remains unchanged.

Example

The flags LIFL2 and LIFL3 shall be cleared in a CNC 84.00. All other flags keep their momentary state.

`<EXECUTE><INFLGCLR>0110..0000</INFLGCLR></EXECUTE>`	Clear one/several flags (all in all 64 signs)
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

State of the flags after command execution:

Before	After
NOLIFL1	NOLIFL1
NOLIFL2	NOLIFL2
LIFL3	NOLIFL3
LIFL4	LIFL4

Related topics

SMDDE object "INFLAGS: State of the Loader Input Flags", page 50

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

INFLGSET: Set Loader Input Flags

Set loader input flags

Command	Description	Version
INFLGSET *flags*	Set loader input flags.	10.08

Argument	Description
flags	Flag list

Return values

Return value	Explanation
SYSSTATUS	State of the command execution
INFLAGS	State of the loader input flags

The command object INFLGSET sets one or several loader input flags.

- Flag states are defined in a string.
- CNC 4x.00: 32 flags (the string must have 32 signs)
- CNC 8x.00: 64 flags (the string must have 32 or 64 signs)
- The 1st sign of the string influences the flag LIFL1.
- The last sign of the string influences the highest significant flag.
- Every sign defines the state of a flag.
  - Sign = "1": The bit is set.
  - Sign = "0": The bit remains unchanged.

Example

The flags LIFL2 and LIFL3 shall be set in a CNC 84.00. All other flags keep their momentary state.

```
<EXECUTE><INFLGSET>0110..0000</
INFLGSET></EXECUTE>
```

Set one/several flags (all in all 64 signs)

```
<REQUEST><SYSSTATUS></SYSSTATUS></
REQUEST>
```

Check, whether the command was received.

State of the flags after command execution:

Before	After
NOLIFL1	NOLIFL1
NOLIFL2	LIFL2
LIFL3	LIFL3
LIFL4	LIFL4

Related topics

SMDDE object "INFLAGS: State of the Loader Input Flags", page 50

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

LOAD: Load File

Load file

Command	Description	Version
LOAD *name*	Load file.	??.??
LOAD *name*|*type*	Load file.	??.??

Argument	Description
name	File name
type	File type

Return values

Return value	Explanation
PCSTATUS = 1	File was loaded correctly
PCSTATUS = 2	Error while loading the file

The command object LOAD loads a file.

- The file name must include the full path.
- The file content is applied to the CNC independently from the file type. The rules for applying the content include the interpretation of the file content (conversion of length-dependent values etc.) and the use within the CNC (tool data, part program etc.).
  - If the transmission parameter is *type* is defined, the rules of this type apply.
  - If the file extension corresponds to a preset type (see section 7.E "File Types", page 99), the rules of this file type apply.
  - If the file has another type and no file type is defined, the content of the file is interpreted as part program and the PGM type rules apply.
- Already existing data in the working memory are overwritten/added.

PCKEY: Transmit PC Keystroke

Transmit PC keystroke

Command	Description	Version
PCKEY *key*	Trigger keystroke on the PC.	10.11

Argument	Description
key	Keyname

Return values

Return value	Explanation
SYSSTATUS	Information, whether the command was transmitted correctly
PCSTATUS = 0	The command has not been executed.
PCSTATUS = 1	Data object that the command has been executed.

The command object PCKEY causes a keystroke on the server PC.

- The data object PCSTATUS includes the information, whether the command was executed correctly.
- An input acknowledgment (Return key) is transmitted with the string “\n”.

Example: automatic user log-in

In this example the user is automatically logged in the CNC. For this purpose, several dialog boxes are confirmed in the page. After login, the page “Work [Work - Status]” appears and the execution is started.

`<EXECUTE><CHANGECLIENT>LOGIN</CHANGECLIENT></EXECUTE>`	Switch screen page
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<EXECUTE><PCKEY>L</PCKEY></EXECUTE>`	Keystroke "L" (call log off dialog)

<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.
<REQUEST><PCSTATUS></PCSTATUS></REQUEST>	Check, whether the keystroke was executed.
<EXECUTE><PCKEY>J</PCKEY></EXECUTE>	Keystroke "J" (confirm dialog)
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.
<REQUEST><PCSTATUS></PCSTATUS></REQUEST>	Check, whether the keystroke was executed.
<EXECUTE><PCKEY>USER\n</PCKEY></EXECUTE>	Transfer user name (\n=acknowledgement)
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.
<REQUEST><PCSTATUS></PCSTATUS></REQUEST>	Check, whether the text was entered.
<EXECUTE><PCKEY>****\n</PCKEY></EXECUTE>	Transfer secret word (\n=acknowledgement)
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.
<REQUEST><PCSTATUS></PCSTATUS></REQUEST>	Check, whether the text was entered.
<EXECUTE><CHANGECLIENT>DEFAULT</CHANGECLIENT></EXECUTE>	Switch screen page
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.
<EXECUTE><CNCKEY>START</CNCKEY></EXECUTE>	START key
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Check, whether the command was received.

Related topics

SMDDE object "CHANGECLIENT: Switch Screen Page", page 26

SMDDE object "PCKEY: Transmit PC Keystroke", page 33

SMDDE object "PCSTATUS: Status of Communication Command", page 54

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

PCCOMMAND: Execute PC Command

Execute PC command

Command	Description	Version
PCCOMMAND *cmd*	Trigger command on the server PC.	10.08

Argument	Description
cmd	Name of the command

Return values

Return value	Explanation
SYSSTATUS	Information, whether the command was transmitted correctly.
PCSTATUS = 0	The command has not been executed.
PCSTATUS = 1	The command has been executed.

The command object PCCOMMAND releases a command on the PC. The string ".NODIALOG" suppresses possible inquiries.

Example

The operating system shall be shut down. Thereby possible inquiries shall be suppressed.

`<EXECUTE><PCCOMMAND>SHUTDOWN.NODIALOG</PCCOMMAND></EXECUTE>`	Shut down operating system
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Related topics

SMDDE object "PCSTATUS: Status of Communication Command", page 54

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

PROGRAM: Transfer File Name

Transfer file name

Command	Description	Version
PROGRAM *file*	Transfer file name or bar code string	10.08

Argument	Description
file	The structure of the argument *file* depends on the current mode: ▸ NOJOBL: *file* = file name ▸ JOBL: *file* = bar code string ▸ The sequence commands CHEK BCOD and CHEK SMDE always expect a bar code string

Return values

Return value	Explanation
ACTPROGRAM	File name of the current part program

Job list mode is off (CNC command NOJOBL)

The command object PROGRAM transmits the name of a part program to the CNC. The CNC configuration determines how the command works:

- Preparation memory off (CNC command NOBELA):
  - The command object PROGRAM loads a part program into the main memory.
  - Any part program in the main storage will be overwritten.
- Preparation memory on (CNC command BELA):
  - The command object PROGRAM loads a part program into the preparation memory.
  - Any part program in the preparation memory will be overwritten.
  - The data object NEXTPROGRAM allows inquiring the file name of the part program in the preparation memory.

Job list mode is on (CNC command JOBL)

The command object PROGRAM transfers a string to the CNC that corresponds to a bar code string. Two parameters of the bar code string are evaluated.

- First parameter of the string: file name of the part program. If the file name consists of an empty string, the job list entry is cleared.
- Fifth parameter of the string: number of the job list entry.

- All other parameters are ignored.

Sequence commands CHEK BCOD, CHEK SMDE

If the sequence command CHEK BCOD or CHEK SMDE waits for a string, the transmitted string must correspond to a bar code string. This behavior does not depend on the CNC configuration (CNC commands JOBL, BELA).

- In the sequence it must be ensured that the bar code string is transmitted within the waiting time of 10 s.
- The CNC and the client are synchronized by setting and checking the loader flags (LIFL and LOFL).
- For detailed information on programming the sequence command CHEK BCOD or CHEK SMDE refer to the manual CNC 8x.00 − Sequences.

The sequence command CHEK BCOD is only available for CNC 84.00. As from software version 11.13.003 the sequence command CHEK SMDE is additionally available for all control systems of the series CNC 8x.00. It checks, if the name of a loaded part program is transmitted as bar code string, if the part program was transferred via DDE protocol, page 9. Both sequence commands, CHEK SMDE and CHEK BCOD, have the same function and same options.

Example: Load part program into the main memory

Requirement

The default mode must be active (CNC command NOJOBL).

The name of the next part program shall be loaded into the main memory. The content of the main memory will be cleared before.

<EXECUTE><CNCCOMMAND>CM@@@</CNCCOMMAND></EXECUTE>	Clear part program from main memory
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Inquiry, whether the previous command has been executed (response=DONE?)
<EXECUTE><PROGRAM>abc.sm5</PROGRAM></EXECUTE>	Load part program into memory
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Inquiry, whether the previous command has been executed (response=DONE?)

Example: Bar code string

Requirement

The job list mode must be active (CNC command JOBL).

The name of the part program at the 3rd place of the job list shall be deleted. Then, the name of a part program shall be entered at the 2nd place of the job list. Note: The dollar sign ($) is used as separator within the bar code string. This is set with the CNC command BACS$ (for example in the Startup file).

<EXECUTE><PROGRAM>$$$$3</PROGRAM></EXECUTE>	Am empty file name is entered at the 3rd place of the job list.
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Inquiry, whether the previous command was executed (response=DONE?)
<EXECUTE><PROGRAM>C:\PRG\PRGR.SM5$$$$2</PROGRAM></EXECUTE>	Enter file name at the 2nd place of the job list
<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>	Inquiry, whether the previous command was executed (response=DONE?)

Related topics

SMDDE object "ACTPROGRAM: File Name of the Current Program", page 43

SMDDE object "NEXTPROGRAM: File Name of Next Part Program", page 51

RUNTIMEVALUE: Current Value of an Object

Current value of an object

Command	Description	Version
RUNTIMEVALUE *term*	Determines the current value of any expression.	10.11

Argument	Description
term	Any expression

Return values

Return value	Explanation
PCSTATUS	Result value

The command object allows requesting the value of any data object from the CNC.

- The result is saved in the data object PCSTATUS.
- The command allows requesting internal variables or calculating arithmetic terms.
- More information on possible request values can be found in the section “Detecting CNC values”.

Example: Determine current X-position

In this example the current X-position of the machine table is requested.

`<EXECUTE><RUNTIMEVALUE>%AuXPos</RUNTIMEVALUE></EXECUTE>`	Request current X-position.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Request result

The response package of the CNC includes the data value.

<REQUEST><PCSTATUS>23456</PCSTATUS></REQUEST>

Example: Calculate arithmetic value

In this example an arithmetic value shall be calculated.

`<EXECUTE><RUNTIMEVALUE>10+2</RUNTIMEVALUE></EXECUTE>`	Transfer arithmetic expression
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><PCSTATUS></PCSTATUS></REQUEST>`	Request result

The response package of the CNC includes the data value.

<REQUEST><PCSTATUS>12</PCSTATUS></REQUEST>

Related topics

SMDDE object "PCSTATUS: Status of Communication Command", page 54

SAVE: Save File

Save file

Command	Description	Version
SAVE *name*	Save the indicated file.	??.??
SAVE *name|type*	Save the indicated file.	??.??

Argument	Description
name	File name
type	File type

Return values

Return value	Explanation
PCSTATUS = 1	The file was saved correctly.
PCSTATUS = 2	Error while saving the file

The command object SAVE saves a file on a data medium.

- The file name must include the full path.
- The fie is saved independently from the file type (conversion of length-dependent values et.) and the use within the CNC (tool data, part program etc.).
  - If the transmission parameter is *type* is defined, the rules of this type apply.
  - If the file extension corresponds to a preset type (see section 7.E "File Types", page 99), the rules of this file type apply.
  - If the file has another extension for which no file type has been defined, the part program will be saved.
- An already existing file is overwritten.

SETHANDLE: Define Data Object

Define data object

Command	Description	Version
SETHANDLE @*handle*	Assign a CNC handle to an @-data object.	10.08

Argument	Description
handle	Number of the @-data object (0 to 9)

The command object SETHANDLE assigns a CNC handle to an @-data object.

- The max. number of @ data objects is 10 (@0 to @9).
- An @-data object can be used like a normal data object.
- For more information on handle identifiers please contact SIEB & MEYER.

Example

In the example the state of the CNC command NDDE shall be inquired. First, an @ data object must be defined for the NDDE handle.Then, the setting of the CNC command NDEE can be determined with the help of the @-data object.

`<EXECUTE><SETHANDLE>0=%FL_JL_NDDE_</SETHANDLE></EXECUTE>`	@0-data object=NDDE
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><@0></@0></REQUEST>`	Request setting for the @0-data object.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response of the CNC includes the data value (0 = NDDE is not active). 1 = NDDE is active).

<REQUEST><@0>1</@0></REQUEST>

STARTEXE: Start Application

Start application

Command	Description	Version
STARTEXE *file*	Start application on the server.	10.08

Argument	Description
file	File name of the application

The command object STARTEXE starts an application on the server.

Example

Execute the file NOTEPAD.EXE.

`<EXECUTE><STARTEXE>c:\windows\notepad.exe</STARTEXE></EXECUTE>`	@0-data object=NDDE
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Related topics

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

5 Data Objects (data items)

Data objects (data items) include information inquired by the client or transmitted automatically by the CNC.

Requirement

Communication with the CNC is only possible, if the receipt of external statements is activated (CNC command NONDDE).

Data objects can only be programmed with the communication commands listed in the table. For details on the data objects refer to chapter 5 "Data Objects (data items)", page 41.

Name	Content
ADVISE	The CNC sends the content of a data object to the client.
ADVISESTART	Instruction to the CNC to activate the automatic advice note for a data object. Whenever the content of the data object changes, the CNC sends an appropriate advice note.
ADVISESTOP	Instruction to the CNC to deactivate the automatic advice note for a data object.
REQUEST	The client requests a data value from the CNC.

Specific notes for TCP/IP:

- The request package includes the data object. One data object corresponds to a an element in the XML structure.
- The response package includes the return value in the data object. Besides, the state of the automatic advice note (Value="ADVISEON" or Value="ADVISEOFF") is returned in the attribute “Value”.
- The content of a data object is always a string.
- Note: A numerical value is also transmitted as string (e.g. the data content = "1" corresponds to the ASCII value 49 (=0x0031).
- An automatic advice note can be defined for every data object (communication commands ADVISESTART and ADVISESTOP).
- In the CNC 8x.00 the ADVISE settings of all data objects are listed on the screen page “System - [TCP/IP]”.

Example: Requesting a CNC object (TCP/IP)

In this example the client requests the name of the current production program.

Request package of the client

```
<SMDNCPACKET Value="124">
 <CNC Value="1">
  <REQUEST Value="010327">
   <ACTPROGRAM>
   </ACTPROGRAM>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The response package serves as acknowledgment of receipt. The XML structure corresponds to the request package.
- The acknowledgment already includes the requested value. In this case the file name.

- The state of the automatic advice note of the CNC object is returned in the attribute “Value” of the data object (in this case: <ACTPROGRAM Value="ADVISEOFF">).

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <REQUEST Value="010327">
   <ACTPROGRAM Value="ADVISEOFF">
    C:\SM_WPROG\DRILL.SM5
   </ACTPROGRAM>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

Related topics

SMDDE object "ADVISESTART: Start Automatic Advice Note", page 19

SMDDE object "ADVISESTOP: Stop Automatic Advice Note", page 20

@: Set Data Object for CNC Handle

Set data object for CNC handle

Command	Description	Version
@*n*=%*handle*	Set data object *n* for a CNC handle.	At sign

Argument	Description
n	Data object 0 to 9
handle	CNC handle

Return values

Return value	Explanation
value	Response of the CNC, depends on the CNC handle

The data objects @*n* allow executing any CNC handles.

- Before a data object @*n* is used, a CNC handle must be assigned with the command object SETHANDLE first.
- Then, all communication commands (EXECUTE, REQUEST etc.) can be used depending on the assigned CNC handle .

Example

In the example the state of the CNC command NDDE shall be requested. First, an @ data object must be defined for the NDDE handle. Then, the setting of the CNC command NDDE can be determined by using the @ data object.

`<EXECUTE><SETHANDLE>0=%FL_JL_NDDE_</SETHANDLE></EXECUTE>`	@0 data object = NDDE
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><@0></@0></REQUEST>`	Request setting for the @0 data object
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response of the CNC includes the data value (0 = NDDE is not active; 1 = NDDE is active).

<REQUEST><@0>1</@0></REQUEST>

Related topics

SMDDE object "SETHANDLE: Define Data Object", page 38

SMDDE object "SYSSTATUS: Status of the Last Communication", page 55

ACTPROGRAM: File Name of the Current Program

File name of the current program

Command	Description	Version
ACTPROGRAM	File name of the current part program	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
file	File name

The data object ACTPROGRAM includes the name of the currently started part program.

Example

The file name of the part program in the main memory shall be determined.

`<REQUEST><ACTPROGRAM></ACTPROGRAM></REQUEST>`	Determine file name.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response of the CNC includes the file name:

<REQUEST><ACTPROGRAM>C:\SMWDATA\PRODUCTION\ABC.SM5</ACTPROGRAM></REQUEST>

Related topics

"NEXTPROGRAM: File Name of Next Part Program", page 51

"PROGRAM: Transfer File Name", page 35

BATCH: Current Batch Number

Current batch number

Command	Description	Version
BATCH	Current batch number	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
value	Batch number

The BATCH data object includes the current batch number.

Example

The current batch number shall be requested during an execution.

`<REQUEST><BATCH></BATCH></REQUEST>`	Determine batch number.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

When the batch "P0034" is being executed, the BATCH data object includes the following string:

<REQUEST><BATCH>P0034</BATCH></REQUEST>

CNCERROR: DNC Message

DNC message

Command	Description	Version
CNCERROR	DNC message	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
DncMessage	(Only CNC 4x.00) The content is *message;number*
DncMessage	(Only CNC 8x.00) The content depends on the DNCM setting.

The data object CNCERROR includes the content of a DNCM message.

- CNC 4x.00: The content of the data object always corresponds to the setting of the CNC command DNCM1. Example: If the execution has been interrupted by pressing the STOP key, the data object includes the text "STOP;3016".
- CNC 8x.00: The content of the data object corresponds to the DNCM setting. Example: If the CNC command DNCM3 is active, the data object includes the text "DRILLPROGRAMM.SM5 * STOP;3016" after the STOP key has been pressed.
- For detailed information refer to the appendix section 7.A "CNC commands", page 73.

Example

The DNC messages shall automatically be sent by the CNC.

`<ADVISESTART><CNCERROR></CNCERROR></ADVISESTART>`	Activate automatic sending of the DNC message.

```
<REQUEST><SYSSTATUS></SYSSTATUS></
REQUEST>
```

Check, whether the command was received.

The following message is sent after the STOP key is pressed (DNCM3 mode):

<ADVISE><CNCERROR>DRILLPROGRAMM.SM5 * STOP;3016</CNCERROR></ADVISE>

Related topics

CNC command "DNCM0: DNC Transmission Mode 0", page 73

CNC command "DNCM1: DNC Transmission Mode 1", page 75

CNC command "DNCM2: DNC Transmission Mode 2", page 76

CNC command "DNCM3: DNC Transmission Mode 3", page 78

CNC command "DNCM4: DNC Transmission Mode 4", page 81

CNCSTATUS: CNC Status

CNC status

Command	Description	Version
CNCSTATUS	CNC status	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation	
AR*hh:mm:ss*	Current running time ("AR00:35:12" for example means that 53 minutes and 12 seconds have passed since the execution has been started).	
AH*n*	Current drill stroke counter ("AH1397" for example means that the tool life counter of the tool in the spindle has a value of 1397 strokes).	
AB*n*	Current block number (e.g. "AB371" means that the 371st program line is being executed).	
AS*n*	Current step number ("AS3" for example means that the 3rd program section is being executed).	
AP*n*	Current progress of the program execution in percent (e.g. "AP42" means that 42 % of the run are finished).	
ZS*zAchsen*	Mask of the selected Z-axes ("ZS0100" for example means that the 2nd axis is selected in a 4-axes machine).	
MO*mode*	Machine status	
	mode = IDLE	Not in Automatic mode
	mode = WAIT	Wait for next program
	mode = WORK	The execution runs.
	mode = STOP	Program stopped
	mode = ALAM	Machine error
	mode = SERV	Service
EC*event*	Event existing of event number and event message	
FN*file*	File name of the current part program	

The data object CNCSTATUS includes the DNC monitoring data. Requirement: The periodical transmission of messages is activated in the CNC (CNC command DNCO).

Example

The CNC status shall be requested during a running execution.

`<REQUEST><CNCSTATUS></CNCSTATUS></REQUEST>`	Determine CNC.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response of the CNC includes the current data.

<REQUEST><CNCSTATUS>AR00:00:28,AH000294,AP000,ZS00111111,MOWORK, EC0000,FNC:\SM_WPROG\ABC.SM3</CNCSTATUS></REQUEST>

CNCTOOLS: Current Tool Data

Current tool data

Command	Description	Version
CNCTOOLS	Current tool data	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
T*t*	Tool number
D*d*[,E*e*]	The diameter and the optional tool type (CNC command TTYP must be activated).
N*n*	Preset tool life
B*b* or C*c*	Current tool life of the drill bit [strokes] or cutter tool life [m]
EN*n*	Total number of the equipped tool of the tool number T
EU*n*	Number of the already executed tools of the tool number T
ER*n*	Number of the still required tools of the tool number T for the current cycle
NN*n*	Total preset tool lives of the equipped tools of the tool number T
NU*n*	Former tool life of all tools of the tool number T
NR*n*	The still required tool life of the tool number T for the current cycle
E*m*	Tool type, if the tool type management is activated (CNC command TTYP)

The data object CNCTOOLS includes information on the tool currently mounted in the spindle.

Example

The current data shall be determined for the tool in the spindle.

`<REQUEST><CNCTOOLS></CNCTOOLS></REQUEST>`	Determine tool data.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response of the CNC includes the current data.

<REQUEST><CNCTOOLS>T4,D0.800,N3000,B125,EN1,EU0,ER0,NN3000,NU125,NR0</CNCTOOLS></REQUEST>

COMMSTATUS: Execution Status of a CNC Command

Execution status of a CNC command

Command	Description	Version
COMMSTATUS	Execution status of a CNC command	10.11

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
OK[:*status*]	The command execution is finished.
ERROR[:*error*]	The command execution caused an error.
BUSY[:*command*]	The command is being executed.

The data object COMMSTATUS includes the status of the last executed command “CNCCOMMAND”.

- This way, it can be evaluated, whether the execution of the command has been finished.
- Additional information possibly transmitted by the CNC will be separated from the status by a colon. Note: The squared brackets in the object table label optional contents. The transmitted string does not include any brackets.
- The ERROR message corresponds to the message in the entry window of the CNC commands (in case of wrong entry).

Example

The CNC command H250. shall be transmitted.

`<EXECUTE><CNCCOMMAND></CNCCOMMAND></EXECUTE>`	Transmit CNC command
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.
`<REQUEST><COMMSTATUS></COMMSTATUS></REQUEST>`	Check, whether the command was received.

Since the allowed H-value has been exceeded, the response from the CNC includes an error message:

<REQUEST><COMMSTATUS>ERROR:value range of parameter violates H25.000</COMMSTATUS></REQUEST>

Related topics

SMDDE object "CNCCOMMAND: Execute CNC Command", page 28

DATETIME: Current Date and Time

Current date and time

Command	Description	Version
DATETIME	Current date and time	10.08

Argument	Description
–	No parameter required.
–	No parameter required.

Return values

Return value	Explanation
datetime	Date and time

The data object DATETIME includes the current date and the current time.

Format	Explanation
tt.mm.jj	Date
hh:mm:ss	Time

Example

The current date and time shall be determined.

`<REQUEST><DATETIME></DATETIME></REQUEST>`	Determine date and time.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

On December 1, 2010, at 03:36 PM the response from the CNC includes the following string:

<REQUEST><DATETIME>01.12.10 15:36:00</DATETIME></REQUEST>

DNCM: Current Mode of the DNCM Command

Current mode of the DNCM command

Command	Description	Version
DNCM	Current mode of the DNCM command	10.01.001

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
value	Mode number

The data object DNCM includes the currently set mode of the CNC command DNCM.

Example

The current DNCM setting shall be determined.

`<REQUEST><DNCM></DNCM></REQUEST>`	Determine DNCM setting
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

If the CNC command DNCM4 is active, the response from the CNC includes the following string:

<REQUEST><DNCM>4</DNCM></REQUEST>

Related topics

SMDDE object "CNCERROR: DNC Message", page 44

DUTY: Load of CNC

Load of CNC

Command	Description	Version
DUTY	Load of CNC	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
value	Percentage value

The data object DUTY includes the load in percent. This value corresponds to the value “Load” in the page “System [Current]”.

Load = execution time / time passed since working shift began

Example

The current machine load shall be determined.

`<REQUEST><DUTY></DUTY></REQUEST>`	Determine current machine load.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

Four hours and 35 minutes have been passed since the working shift began.During this time, the execution was started for a period of 3 hours and 47 minutes. The load is 83%. *Note: Before calculating this value manually, you must convert the minutes in percent values (p = min/60).* The response from the CNC includes the load.

<REQUEST><DUTY>83</DUTY></REQUEST>

FTASTEN: State of the Function Keys

State of the function keys

Command	Description	Version
FTASTEN	State of the function keys	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation	CNC
keyString	Message = state of 16 function keys	CNC 4x.00
keyString	Message = string of a button	CNC 8x.00

CNC 4x.00

The data object FTASTEN includes the state of the function keys on the PC keyboard.

- The string has 16 signs.
- Every sign shows the state of a function key.
  - Sign = 1: The key is pressed.
  - Sign = 0: The key is not pressed.

CNC 8x.00

The data object FTASTEN includes the string defined for the currently pressed button.

- The strings for the buttons are configured in the file "UPDATE.MEN".
- A string must have 16 signs. Only the signs "0" and "1" are allowed.
- The string can include several "1" signs. This way, 65535 combinations are allowed.

Example

Whenever a function key is pressed the CNC shall send a string.

`<ADVISESTART><FTASTEN></FTASTEN></ADVISESTART>`	Monitor buttons.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

After a button has been pressed, the response from the CNC includes the following string:

<ADVISE><FTASTEN>0000001000000001000000</FTASTEN></ADVISE>

Related topics

SMDDE object "PCKEY: Transmit PC Keystroke", page 33

INFLAGS: State of the Loader Input Flags

State of the loader input flags

Command	Description	Version
INFLAGS	State of the loader input flags	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation	CNC
flags	32 signs	CNC 4x.00
flags	64 signs	CNC 8x.00

The data object INFLAGS includes the state of the loader input flags.

- All states are listed in one string.
- The length of the string depends on the number of available flags in the CNC.
- Every sign represents the state of a flag.
  - Sign = "1": The bit is set
  - Sign = "0": The bit is cleared
- The first sign includes the state of the flag LIFL1.
- The last sign includes the state of the highest significant flag.
- The flags are set or cleared via data package or CNC command:
  - Data package with command object: INFLGSET sets and INFLGCLR clears a flag.
  - CNC commands (*requirement: user level 3 or higher*): LIFL sets a flag and NOLIFL clears a flag.

Example:

The states of the loader input flags shall be determined.

`<REQUEST><INFLAGS></INFLAGS></REQUEST>`	Determine the state of the flags.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response from the CNC includes the following string:

<REQUEST><INFLAGS>110000000000000000000000000000000000000000000000
0000000000000000</INFLAGS></REQUEST>

The content of the string corresponds to the following CNC command settings:

- LIFL1 = ON
- LIFL2 = ON
- All other flags are off.

Related topics

SMDDE object "INFLGCLR: Clear Loader Input Flags", page 31

SMDDE object "INFLGSET: Set Loader Input Flags", page 31

SMDDE object "OUTFLAGS: State of Loader Output Flags", page 53

NEXTPROGRAM: File Name of Next Part Program

File name of the next part program

Command	Description	Version
NEXTPROGRAM	File name of the next part program	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
file	File name

The data object NEXTPROGRAM includes the name of the part program to be executed next.

Example

The file name of the part program to be executed next shall be determined.

`<REQUEST><NEXTPROGRAM></NEXTPROGRAM></REQUEST>`	Determine file name.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response from the CNC includes the following string:

<REQUEST><NEXTPROGRAM>C:\SM_WPROG\ABC.SM3</NEXTPROGRAM></REQUEST>

Related topics

SMDDE object "ACTPROGRAM: File Name of the Current Program", page 43

SMDDE object "PROGRAM: Transfer File Name", page 35

OPID: OPID String

OPID string

Command	Description	Version
OPID	OPID string	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
text	OPID string

The OPID data object includes the OPID string. The OPID string is defined with the CNC command OPID.

Example

The current OPID string shall be determined.

`<REQUEST><OPID></OPID></REQUEST>`	Determine OPID string.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

If the CNC command OPID,M13 is active, the data object OPID includes the following string:

<REQUEST><OPID>M13</OPID></REQUEST>

OUTFLAGS: State of Loader Output Flags

State of the loader output flags

Command	Description	Version
OUTFLAGS	State of the loader output flags	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation	CNC
flags	32 signs	CNC 4x.00
flags	64 signs	CNC 8x.00

The data object OUTFLAGS includes the state of the loader output flags.

- All states are listed in one string.
- The length of the string depends on the number of available flags in the CNC.
- Every sign represents the state of a flag.
  - Sign = "1": The bit is set
  - Sign = "0": The bit is cleared
- The first sign includes the state of the flag LOFL1.
- The last sign includes the state of the highest significant flag.
- The flags are set or cleared via CNC command or FIIPS command.
  - CNC commands (*requirement: user level 3 or higher*): LOFL sets a flag and NOLOFL clears a flag.
  - FIPS commands: SET LOFL sets a flag and CLR LOFL clears a flag.

Example:

The states of the loader output flags shall be determined.

`<REQUEST><OUTFLAGS></OUTFLAGS></REQUEST>`	Determine the state of the flags
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response from the CNC includes the following string:

<REQUEST><OUTFLAGS>1100000000000000000000000000000000000000000000000000000000000000</OUTFLAGS></REQUEST>

The content of the string corresponds to the following CNC command settings:

- LOFL1 = ON
- LOFL2 = ON
- All other flags are off.

Related topics

SMDDE object "INFLAGS: State of the Loader Input Flags", page 50

SMDDE object "INFLGCLR: Clear Loader Input Flags", page 31

SMDDE object "INFLGSET: Set Loader Input Flags", page 31

PCSTATUS: Status of Communication Command

Status of a communication command

Command	Description	Version
PCSTATUS	Status of a communication command	10.11

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
value	Status value

The answers of a communication process are provided in the PCSTATUS data object (see table).

Command object	Meaning
CHANGECLIENT	0 = page was not switched
	1 = page was switched
PCCOMMAND	0 = command was not executed
	1 = command was executed
PCKEY	0 = keystroke was not executed
	1 = keystroke was executed
RUNTIMEVALUE	Result

Examples

Examples can be found in the description of the corresponding command.

Related topics

SMDDE object "CHANGECLIENT: Switch Screen Page", page 26

SMDDE object "PCCOMMAND: Execute PC Command", page 34

SMDDE object "PCKEY: Transmit PC Keystroke", page 33

SMDDE object "RUNTIMEVALUE: Current Value of an Object", page 37

SCREENSAVER: Current Event Message

Current event message

Command	Description	Version
SCREENSAVER	Current event message	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
message	Color design and content of the event message *background color text color*\n*message*

The data object SCREENSAVER includes the background color, the text color and the text of the event message currently shown on the action bar.

- The RGB color values are transmitted as decimal values.
- The color values are separated by a blank character.
- The text of the message is separated from the color values by the signs “\n”.
- The text of a message can additionally include the signs “\n”. These signs cause a line break in the text shown in the action bar.
- This allows for example to show this message as screensaver.

Example

The current OPID string shall be determined.

```
<REQUEST><SCREENSAVER></SCREEN-
SAVER></REQUEST>
```

Determine message text.

```
<REQUEST><SYSSTATUS></SYSSTATUS></
REQUEST>
```

Check, whether the command was received.

The response from the CNC includes the string

<REQUEST><SCREENSAVER>16711680 16776960\nBLOCK\nX100.Y100.</SCREENSAVER></REQUEST>

The parts of the message have the following meaning:

Content of the message	Explanation
16711680	Background color = red (0xFF0000)
16776960	Text color = yellow (0xFFFF00)
\n	Separator (= line break in the CNC display)
BLOCK\nX100.Y100.	Event message which includes a line break (\n)

SYSSTATUS: Status of the Last Communication

Status of the last communication

Command	Description	Version
SYSSTATUS	Status of the last communication	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
DONE	The command has been executed successfully..
WRONG COMMAND	The command is not allowed (e.g. clear memory when the execution is running).
SYNTAX ERROR	Unknown command

Return value	Explanation
PARAMETER ERROR	The command includes wrong parameters (e.g. the flag list of the command object INFLGSET is too short or too long).
OFFLINE	The connection to the CNC module has been interrupted.
BUFFER NOT EMPTY	The command buffer is not free (the last command has not yet been executed).

The data object SYSSTATUS includes information about the communication command sent last.

- This allows for example evaluating, whether the writing and the transmission were correct.
- The content of the status value is cleared by reading-out the status value. If you need the status later in the sequence of your program, you must save it in a STRING variable.

Example

The file name of the part program to be executed next shall be determined.

`<REQUEST><NEXTPROGRAM></NEXTPROGRAM></REQUEST>`	Determine file name.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

If the NEXTPROGRAM command is received correctly the response string from the CNC is:

<REQUEST><SYSSTATUS>DONE</SYSSTATUS></REQUEST>

USERLEVEL: Current Access Authorization

Current access authorization

Command	Description	Version
USERLEVEL	Current access authorization	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
level	Value

The USERLEVEL data object includes the currently set access authorization and the associated access rights.

- The values of the data objects are always of the data type STRING.
- The value 1 (x0001), for example, is transmitted as ACSII sign "1" (x0031).

Example

The access level of the user currently logged on shall be determined.

`<REQUEST><USERLEVEL></USERLEVEL></REQUEST>`	Determine message text.

`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response string from the CNC for user level "3" is:

<REQUEST><USERLEVEL>3</USERLEVEL></REQUEST>

USERNAME: Name of the Current User

Name of the current user

Command	Description	Version
USERNAME	Name of the current user	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
name	Name of the user

The data object USERNAME includes the name of the user logged in the CNC.

Example

The name of the user currently logged on shall be determined.

`<REQUEST><USERNAME></USERNAME></REQUEST>`	Determine message text.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response string from the CNC for the user "SIEB & MEYER" is:

<REQUEST><USERNAME>SIEB & MEYER</USERNAME></REQUEST>

VERSION: SIEB & MEYER TCP/IP Version

SIEB & MEYER TCP/IP version

Command	Description	Version
VERSION	SIEB & MEYER TCP/IP version	10.08

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
version	Version number

The data object VERSION includes the SIEB & MEYER TCP/IP version.

Example

The TCP/IP version shall be determined.

`<REQUEST><VERSION></VERSION></REQUEST>`	Determine message text.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response string of the CNC for version number 1.2 is:

<REQUEST><VERSION>1.2</VERSION></REQUEST>

XYPOSITION: Current Table Position

Current table position

Command	Description	Version
XYPOSITION	Current table position	10.11

Argument	Description
–	No parameter required.

Return values

Return value	Explanation
position	XY coordinates Reference point = program zero

The data object XYPOSITION includes the coordinates of the current table position.

- The position is provided as string (e.g. "X23.456Y65.432").
- A point is used as separator in the coordinate values..
- The format of the coordinate values depends on the current format setting in the CNC (CNC command METR / INCH).
- The program zero results from the sum of the defined offsets (fix offset in the machine parameters, CNC commands FV, FVO, FA, AOFF).

Example

The current XY-coordinates shall be determined.

`<REQUEST><XYPOSITION></XYPOSITION></REQUEST>`	Determine message text.
`<REQUEST><SYSSTATUS></SYSSTATUS></REQUEST>`	Check, whether the command was received.

The response string from the CNC for position X123.456Y54.321 is:

<REQUEST><XYPOSITION> X123.456Y54.321</XYPOSITION></REQUEST>

6 Examples

This section includes examples for a remote control of the CNC and for a query of data.

Notes on the examples:

- A schematic sequence of the commands is listed in the example "Execute part program".
- The examples "Move axes" and "Show tool parameters" represent the transmission packages with all objects.

Handles of the user interface

You can use the function for editing elements of the user interface to display the names of the handles of these elements. You activate the function with user level 7 with the option “Yes” under “Installation settings → Display and help → Editing PAG pages and MEN menus allowed?”.

If this function is active, a dialog will open for editing elements of the user interface when you double-click with the right mouse button on such an element. In the text field “Handle” the name of the handle is shown. You can copy the name and enter it in your request.

6.1 Execute Part Program

In the example, a part program is loaded, executed and cleared again from the main storage.

- The sequence may be incomplete in some details.
- You must adapt the sequence to the specific characteristics of your machine.
- Further checks etc. may be necessary for your production process.

Requirement

The connection to the CNC must be active and the CNC must be online!

`REQUEST CNCERROR`	Check, whether the CNC status "Wait for program;3316" or "Program check;3048" is active.
`EXECUTE CNCCOMMAND CM@@@`	Remove the part program from the main storage.
`EXECUTE PROGRAM`	Transmit the part program to the CNC.
`REQUEST PROGRAM`	Check, whether the correct program is in the main storage of the CNC.
`REQUEST CNCERROR`	Check, whether the CNC status "Wait for program;3155" is active.
`EXECUTE CNCKEY START`	Start execution (START key)
`REQUEST CNCERROR`	Check, whether the execution is finished (status: "End of program;3048")
`EXECUTE CNCCOMMAND CM@@@`	Remove the part program from the main storage.

The numbers of the messages are described in the manual “Event Messages”.

6.2 Move Axes

In this example the Z-axes are retracted and then, the XY-axes move to the park position.

- To get an automatic advice note when the park position is reached, ADVISESTART must be activated for the data object “CommStatus”.
- During positioning “COMMSTATUS is BUSY”.
- After the park position has been reached, the “COMMSTATUS is OK”.

Request package of the client for retracting the Z-axes

```
<SMDNCPACKET Value="123">
  <CNC Value="1">
    <EXECUTE Value="010326">
      <CNCCOMMAND>
        H
      </CNCCOMMAND>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

- The acknowledgment only signals that the command has been received correctly.
- It does not include any information on whether the command has already been executed.

```
<SMDNCPACKET Value="332553">
  <CNC Value="1">
    <EXECUTE Value="010326">
      <CNCCOMMAND VALUE=934>
        H
      </CNCCOMMAND>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Request package of the client for moving to the park position

```
<SMDNCPACKET Value="124">
  <CNC Value="1">
    <EXECUTE Value="010327">
      <CNCCOMMAND Value="84">
        P
      </CNCCOMMAND>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332554">
  <CNC Value="1">
    <EXECUTE Value="010327">
      <CNCCOMMAND Value="84">
      P
    </CNCCOMMAND>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Automatic advice note of the CNC after positioning has been started

```
<SMDNCPACKET Value="332559">
 <CNC Value="1">
  <ADVISE Value="050593">
   <COMMSTATUS Value="ADVISEON">
    BUSY
   </COMMSTATUS>
  </ADVISE>
 </CNC>
</SMDNCPACKET>
```

Response package of the client

```
<SMDNCPACKET Value="143">
 <CNC Value="1">
  <ADVISE Value="050593">
   <COMMSTATUS Value="ADVISEON">
    BUSY
   </COMMSTATUS>
  </ADVISE>
 </CNC>
</SMDNCPACKET>
```

6.3 Show Tool Parameters

In this example the screen page “Tool parameters (dyn.)” appears on the CNC user interface.

Request package of the client for switching the screen page

```
<SMDNCPACKET Value="151">
 <CNC Value="1">
  <EXECUTE Value="010328">
   <CHANGECLIENT Value="324">
    TOOLSDYN
   </CHANGECLIENT>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332569">
 <CNC Value="1">
```

```
    <EXECUTE Value="010328">
      <CHANGECLIENT Value="324">
        TOOLSDYN
      </CHANGECLIENT>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Automatic advice note of the CNC after the park position has been reached

```
<SMDNCPACKET Value="332582">
  <CNC Value="1">
    <ADVISE Value="050594">
      <COMMSTATUS Value="ADVISEON">
        OK
      </COMMSTATUS>
    </ADVISE>
  </CNC>
</SMDNCPACKET>
```

Response package of the client

```
<SMDNCPACKET Value="152">
  <CNC Value="1">
    <ADVISE Value="050594">
      <COMMSTATUS Value="ADVISEON">
        OK
      </COMMSTATUS>
    </ADVISE>
  </CNC>
</SMDNCPACKET>
```

6.4 Requesting Language of the User Interface

In these examples the language of the user interface is requested. The request uses the command object RUNTIMEVALUE.

Determine language of the user interface

With the command object RUNTIMEVALUE you can determine the language of the user interface.

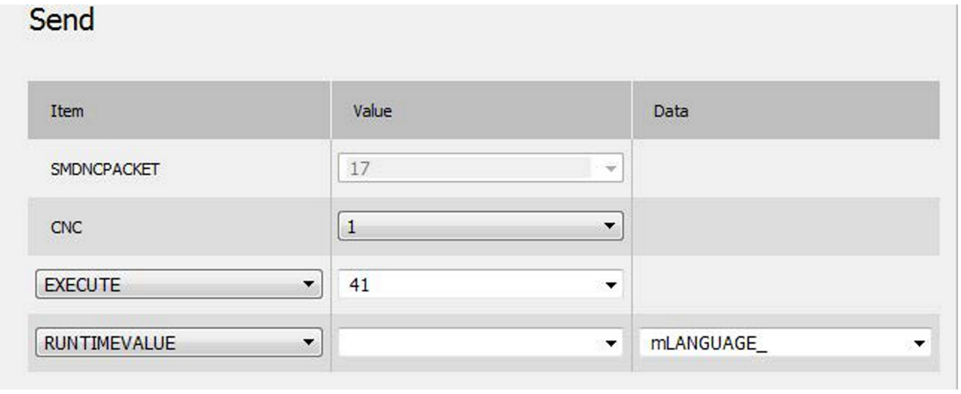
Possible return values of the response package:

- 0: English
- 1: German
- 2: Chinese (simplified)
- 3: Chinese (traditional)
- 4: Japanese
- 5: Alternative language

Request package of the client to ask for the language of the user interface:

```
<SMDNCPACKET Value="17">
```

```
  <CNC Value=1>
    <EXECUTE Value="41">
      <RUNTIMEVALUE>
        mLANGUAGE_
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```



Send

Item	Value	Data
SMDNCPACKET	17	
CNC	1	
EXECUTE	41	
RUNTIMEVALUE		mLANGUAGE_

Fig. 2: Request to determine the language of the user interface with AnyWhere2SM

Response package of the CNC for an English user interface:

```
<SMDNCPACKET Value="332553">
  <CNC Value=1>
    <EXECUTE Value="41">
      <RUNTIMEVALUE Value="0">
        mLANGUAGE_
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Checking language of the user interface

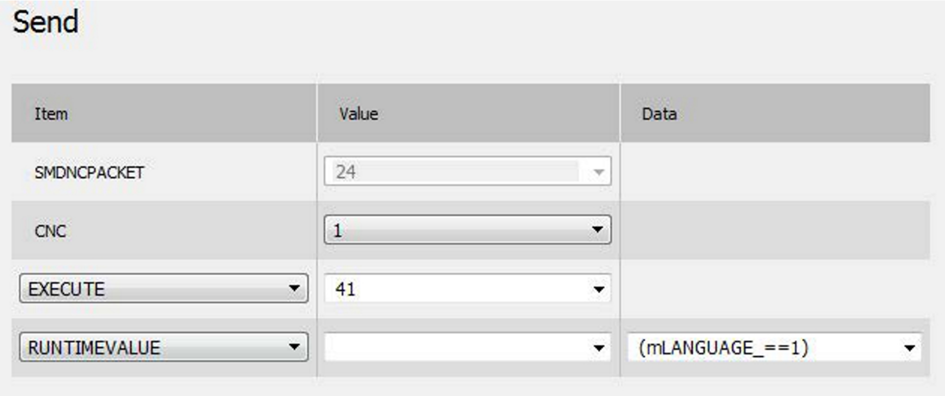
With the command object RUNTIMEVALUE you can check, if a certain language is set for the user interface.

Possible return values of the response package:

- 0: The language of the user interface does not correspond to the value requested.
- 1: The language of the user interface corresponds to the value requested.

Request package of the client to check the language setting of a German user interface:

```
<SMDNCPACKET Value="24">
  <CNC Value=1>
    <EXECUTE Value="41">
      </RUNTIMEVALUE>
        mLANGUAGE_==1
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```



Send

Item	Value	Data
SMDNCPACKET	24	
CNC	1	
EXECUTE	41	
RUNTIMEVALUE		(mLANGUAGE_==1)

Fig. 3: Request for checking the language of the user interface with AnyWhere2SM

Response package of the CNC for an English user interface:

```
<SMDNCPACKET Value="332553">
  <CNC Value=1>
    <EXECUTE Value="41">
      <RUNTIMEVALUE Value="0">
        mLANGUAGE_==1
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

6.5 Requesting and Setting Tool Parameters

Here it is described how to request and set tool parameters.

Handles for tool parameters

Possible handles to determine tool parameters:

- TParD: diameter D
- TParS: spindle speed S
- TParF: Z-axis feed rate F
- TParA: waiting time A
- TParZ: offset Z
- TParN: tool life N

In the request the handle is transferred with an index for the tool number, e.g. TParN(2) for T3. Please note, that the index counting starts with 0, so that T1 is specified with TParN(0), T2 with TParN(1) etc.

Determine tool life

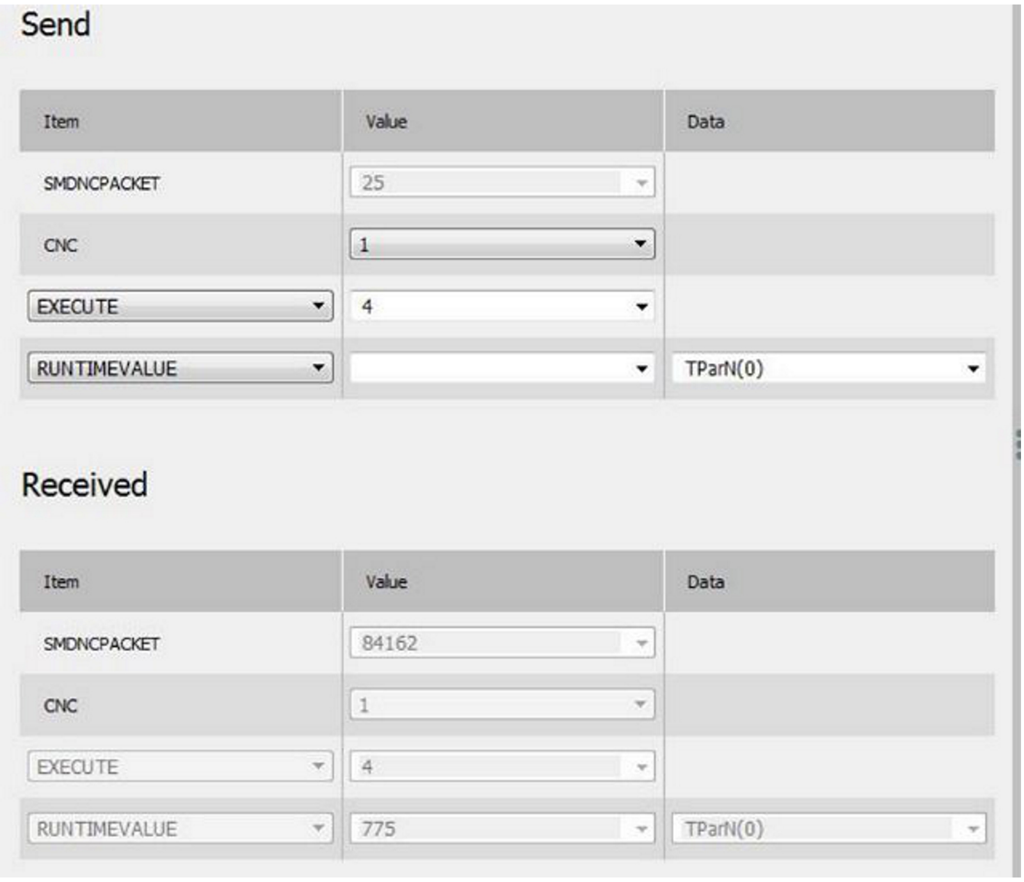
In this example the tool life of T1 is requested. The request uses the command object RUNTIMEVALUE.

Request package of the client to request the tool life of T1:

```
<SMDNCPACKET Value="25">
  <CNC Value=1>
    <EXECUTE Value="4">
      </RUNTIMEVALUE>
Please note, that the index counting starts with 0, so that T1 is specified with TParN(0), T2 with TParN(1)
etc.
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC for a tool life of 775:

```
<SMDNCPACKET Value="84162">
  <CNC Value=1>
    <EXECUTE Value="4">
      <RUNTIMEVALUE Value="775">
        %TParN(0)_
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```



Send

Item	Value	Data
SMDNCPACKET	25	
CNC	1	
EXECUTE	4	
RUNTIMEVALUE		TParN(0)

Received

Item	Value	Data
SMDNCPACKET	84162	
CNC	1	
EXECUTE	4	
RUNTIMEVALUE	775	TParN(0)

Fig. 4: Determining the tool life with AnyWhere2SM

Set tool life

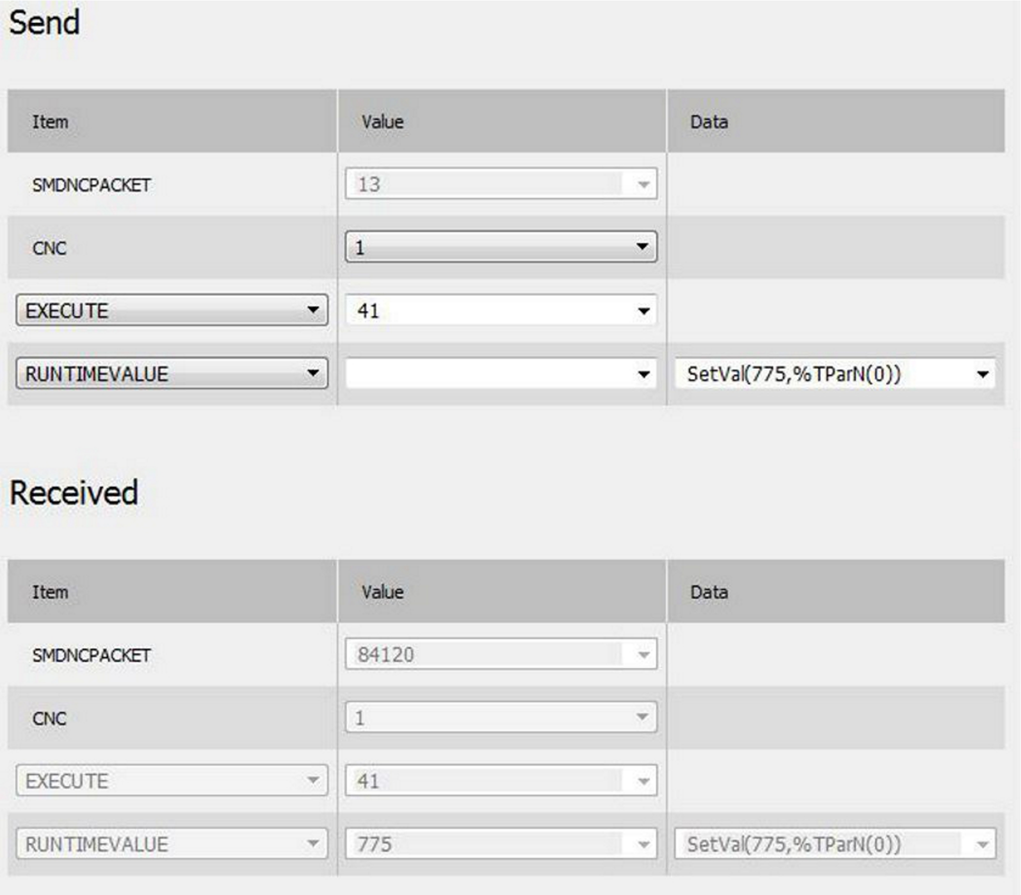
In this example the tool life for T1 is set. Thereby the desired value is transmitted to the handle TParN(0) with the function SetVal, page 85.

Request package of the client to set the tool life of 775 for T1:

```
<SMDNCPACKET Value="13">
  <CNC Value=1>
    <EXECUTE Value="41">
      </RUNTIMEVALUE>
        SetVal(775,%TParN(0))
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC after setting the tool life of 775 successfully:

```
<SMDNCPACKET Value="84120">
  <CNC Value=1>
    <EXECUTE Value="41">
      <RUNTIMEVALUE Value="775">
        %TParN(0)_
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```



Send

Item	Value	Data
SMDNCPACKET	13	
CNC	1	
EXECUTE	41	
RUNTIMEVALUE		SetVal(775,%TParN(0))

Received

Item	Value	Data
SMDNCPACKET	84120	
CNC	1	
EXECUTE	41	
RUNTIMEVALUE	775	SetVal(775,%TParN(0))

Fig. 5: Setting the tool life N=775 for T1 with AnyWhere2SM

6.6 Request Required Tools

In these examples the required tools are requested. The value is specified in the part program. For the request you can use the command object RUNTIMEVALUE or the data object @.

Handles for requested tools

Possible handles to determine the required tools:

- mlTotalHits: Total number of programmed drill strokes
- mlTotalPath: Value of the routing track programmed in total

Determine number of drill strokes

In the following example the total number of programmed drill strokes is requested with the command object RUNTIMEVALUE.

Request package of the client to request the programmed drill strokes:

```
<SMDNCPACKET Value="17">
  <CNC Value=1>
    <EXECUTE Value="41">
      </RUNTIMEVALUE>
        %mlTotalHits
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC for 2528 programmed drill strokes:

```
<SMDNCPACKET Value="332553">
  <CNC Value=1>
    <EXECUTE Value="41">
      <RUNTIMEVALUE Value="2528">
        %mlTotalHits
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Alternatively, it is also possible to define a data object for the required tools (with "SETHANDLE @mlTotalHits" for the number of drill strokes) which then is requested:

Request package of the client to define the data object:

```
<SMDNCPACKET Value="1">
  <CNC Value=1>
    <EXECUTE Value="1234">
      <SETHANDLE>
        0=%mlTotalHits
      </SETHANDLE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC to define the data object:

```
<SMDNCPACKET Value="342083">
  <CNC Value=1>
    <EXECUTE Value="1234">
      <SETHANDLE>
        0=%mlTotalHits
      </SETHANDLE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Request package of the client to request the data object with index 0:

```
<SMDNCPACKET Value="2">
  <CNC Value=1>
    <REQUEST Value="24">
      <@0>
      </@0>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC for 2528 programmed drill strokes:

```
<SMDNCPACKET Value="342084">
 <CNC Value=1>
  <REQUEST Value="24">
   <@0 Value="ADVISEOFF">
    2528
   </@0>
  </REQUEST>
 </CNC>
</SMDNCPACKET>
```

6.7 Requesting Operating Data

In these examples the operating data is requested. For this purpose, SMDDE offers different possibilities. You can determine the operating data via protocol messages or request the data from the CNC directly.

Determining operating data via protocol messages

The operating messages are logged. Therefore they are available via the automatic CNC notification with ADVISE by using the data object CNCERROR. CNCERROR provides all protocol messages. With ADVISE the notification of the CNC is made automatically, a special request is not necessary. Example for a response package after a program run:

```
<SMDNCPACKET Value="10579988">
 <CNC Value=1>
  <ADVISE Value="1">
   <CNCERROR Value="ADVISEON">
    C:\SMWDATA\CNC1\PINNING_PATTERN\DRILL_PROGRAM.EX2 *
S10:56:10,E11:26:28,R00:00:22,W00:02:51,H000004,B000,C000000;3364
   </CNCERROR>
  </ADVISE>
 </CNC>
</SMDNCPACKET>
```

Requesting operating data of the CNC

Alternatively, it is also possible to define data objects for operating data (with "SETHANDLE **@*HandleName* (*DataIndex*)**") and to request them afterwards.

Request package of the client to define a data object with the handle name "HSYS55_BETRZEITEN":

```
<SMDNCPACKET Value="11">
 <CNC Value=1>
  <EXECUTE Value="1234">
   <SETHANDLE>
    0=%HSYS55_BETRZEITEN (52)
   </SETHANDLE>
  </EXECUTE>
 </CNC>
```

```
</SMDNCPACKET>
```

Response package of the CNC to define the data object:

```
<SMDNCPACKET Value="342083">
  <CNC Value=1>
    <EXECUTE Value="1234">
      <SETHANDLE>
        0=%HSYS55_BETRZEITEN (52)
      </SETHANDLE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Request package of the client to request the data object with index 0:

```
<SMDNCPACKET Value="12">
  <CNC Value=1>
    <REQUEST Value="1234">
      <@0>
           0=%HSYS55_BETRZEITEN (52)
      </@0>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC for a requested operating time of 2:35:32 (hh:mm:ss):

```
<SMDNCPACKET Value="342084">
  <CNC Value=1>
    <REQUEST Value="1234">
      <@0 Value="ADVISEOFF">
        2:35:32
      </@0>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

6.8 Loading Diameter File

In this example a diameter file is loaded. For loading the file you use the command object COMMAND.

Request package of the client to load the file "MyDiameters.dia":

```
<SMDNCPACKET Value="11">
  <CNC Value=1>
    <EXECUTE Value="5">
      <COMMAND>
        CNCCOMMAND=@LF,C:\SMWDATA\DIAMETER\MyDiameters.dia
      </COMMAND>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="1312234">
  <CNC Value=1>
    <EXECUTE Value="5">
      <COMMAND Value="1">
        CNCCOMMAND=@LF,C:\SMWDATA\DIAMETER\MyDiameters.dia
      </COMMAND>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

You can ask whether the execution was successful:

```
<SMDNCPACKET Value="15">
  <CNC Value=1>
    <REQUEST Value="4">
      <SYSSTATUS Value="ADVISEOFF">
      </SYSSTATUS>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC after successful execution:

```
<SMDNCPACKET Value="1312238">
  <CNC Value=1>
    <REQUEST Value="4">
      <SYSSTATUS Value="ADVISEOFF">
        DONE
      </SYSSTATUS>
    </REQUEST>
  </CNC>
</SMDNCPACKET>
```

7 Appendix

7.A CNC commands

This appendix contains CNC commands required for communication via SMDDE.

DNCM0: DNC Transmission Mode 0

DNC transmission mode 0

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
DNCM	Set DNC transmission mode 0	10.01.001

Requirement

Software that can evaluate DNC event messages is installed on the server computer (e.g. SMWinDNC by SIEB & MEYER).

The CNC command DNCM0 defines that a message is transmitted via the DNC interface after an event has occurred and that a set of messages is generated and transmitted via the DNC interface at the end of a run.

For more details refer to the description of the CNC command DNCM,.

Syntax of a single message

dd/*mm*/*yy hh*:*mm*:*ss progName* * *eventTxt*

Meaning of the elements:
- *dd*/*mm*/*yy* = system date of the CNC
- *hh*:*mm*:*ss* = system time of the CNC
- *progName* = file name of the part program
- * = separator between the file name and the event message
- *eventTxt* = event message as plain text

Example for a single message

When the part program DRILL.SM5 was executed on 2010-05-23, the "stop" became active at 02:21:35 p.m. The following event message is generated:

23/05/10 14:21:35 DRILL.SM5 * STOP

Syntax of the event messages at the end of a run

S*hh*:*mm*:*ss*,E*hh*:*mm*:*ss*,R*hh*:*mm*:*ss*,W*hh*:*mm*:*ss*,H*hhhhhh*,B*bbbb*,C*cccccc*,I*hh*:*mm*:*ss*
TC*nnnn*,TE*eeee*,TT*hh*:*mm*:*ss*
LC*nnnn*,LE*eeee*,LT*hh*:*mm*:*ss*
ST*hh*:*mm*:*ss*,*hh*:*mm*:*ss*,…
OT*hh*:*mm*:*ss*,OH*hhhhhhhh*,OC*cccccccc*
AR*hh*:*mm*:*ss*,AH*hhhhhh*,AP*ppp*,ZS*zzzzzzzz*,*eventTxt*,EC*nnnn*

B*nnn*,Z*zzzzzzzz*,T*nn*,H*hhhhhh*,X*x.xxx*Y*y.yyy*

Meaning of the elements

- 1st line: times and counters
  - S = start time
  - E = end time
  - R = duration of the run
  - W = waiting time between two runs
  - H = number of drill strokes
  - B = number of broken tools (see also the line with the broken tool message)
  - C = length of all routing tracks in millimeters
  - I = waiting time during the run
- 2nd line: tool change
  - TC = number of tool changes
  - TE = number of tool change errors
  - TT = duration of all tool changes
- 3rd line: loading processes
  - LC = number of loading processes
  - LE = number of loading errors
  - LT = duration of all loading errors
- 4th line: running time of the spindles since initial operation
  - ST = total running time of all spindles (the indicated time are separated by a comma)
- 5th line: counter values since initial operation
  - OT = total execution time
  - OH = total number of drill strokes
  - OC = length of all routing tracks in meters
- 6th line: periodic messages
  - AR = duration of a run
  - AH = number of drill strokes
  - AP = current value of the progress bar
  - ZS = spindle selection
  - *eventTxt* = current event message
  - EC = current event number
- 7th line an following: broken drill message for every broken drill
  - B = number of the broken tool
  - Z = spindle number as binary value
  - T or D = current tool number or current tool diameter
  - H = drill stroke counter for broken tool
  - XY = position of the broken tool

7.A

Example for a set of messages at the end of a run

The following event messages are generated and transmitted at the end of a run.

```
S07:03:05,E07:10:03,R00:06:58,W00:00:00,H000654,B02,C002138,I00:00:00
TC0009,TE0000,TT00:01:16
LC0000,LE0000,LT00:24:00
ST2102:07:23,2091:42:15,1914:26:12
OT2102:12:23,OH00930582,OC002137
AR00:06:58,AH000654,AP999,ZS00000110,MOWORK,EC0000
B001,Z00000010,D00.800,H000765,X043.642Y153.321
B002,Z00000010,D00.800,H001852,X123.321Y182.443
```

Related topics

CNC command "DNCM1: DNC Transmission Mode 1", page 75

CNC command "DNCM2: DNC Transmission Mode 2", page 76

CNC command "DNCM3: DNC Transmission Mode 3", page 78

CNC command "DNCM4: DNC Transmission Mode 4", page 81

DNCM1: DNC Transmission Mode 1

DNC transmission mode 1

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
DNCM1	Set DNC transmission mode 1	10.01.001

The CNC command DNCM1 defines that a message is transmitted via the DNC interface after an event has occurred and that a set of messages is generated and transmitted via the DNC interface at the end of a run.

For more details refer to the description of the CNC command DNCM,.

Syntax of a single message

eventTxt;*eventNo*

Meaning of the elements:
- *eventTxt* = event message as plain text
- *eventNo* = event number of the event message

Example for a single message

The <STOP> key is pressed during the execution of a part program. The following event message is generated:

STOP;3016

Note: To get the real event number, the value 3000 must be subtracted from every event number (3016 − 3000 = 16).

Syntax of the event messaged at the end of a run

S*hh:mm:ss*,E*hh:mm:ss*,R*hh:mm:ss*,W*hh:mm:ss*,H*hhhhhh*,B*bbbb*,C*cccccc*,I*hh:mm:ss*;3360
TC*nnnn*,TE*eeee*,TT*hh:mm:ss*;3361
LC*nnnn*,LE*eeee*,LT*hh:mm:ss*;3362
ST*hh:mm:ss*,*hh:mm:ss*,*…*;3363
OT*hh:mm:ss*,OH*hhhhhhhh*,OC*cccccccc*;3364
AR*hh:mm:ss*,AH*hhhhhh*,AP*ppp*,ZS*zzzzzzzz*,*eventTxt*,EC*nnnn*;3378
B*nnn*,Z*zzzzzzzz*,T*nn*,H*hhhhhh*,X*x.xxx*Y*y.yyy*;3380

Meaning of the elements
- 1st line: times and counters, code = 3360
  - S = start time
  - E = end time
  - R = duration of the run
  - W = waiting time between two runs
  - H = number of drill strokes
  - B = number of broken tools (see also the line with the broken tool message)
  - C = length of all routing tracks in millimeters
  - I = waiting time during the run

- 2nd line: tool change, code = 3361
  - TC = number of tool changes
  - TE = number of tool change errors
  - TT = duration of all tool changes
- 3rd line: loading processes, code = 3362
  - LC = number of loading processes
  - LE = number of loading errors
  - LT = duration of all loading errors
- 4th line: running times of the spindles since initial operation, code = 3363
  - ST = total running time of all spindles (the indicated time are separated by a comma)
- 5th line: counter values since initial operation, code = 3364
  - OT = total execution time
  - OH = total number of drill strokes
  - OC = length of all routing tracks in meters
- 6th line: periodic messages, code = 3378
  - AR = duration of a run
  - AH = number of drill strokes
  - AP = current value of the progress bar
  - ZS = spindle selection
  - *eventTxt* = current event message
  - EC = current event number
- 7th line an following: broken tool message for every broken tool, code = 3380, ⋯
  - B = number of the broken tool
  - Z = spindle number as binary value
  - T or D = current tool number or current tool diameter
  - H = drill stroke counter for broken tool
  - XY = XY position of the broken tool

Example for a set of messages at the end of a run

The following event messages are generated and transmitted at the end of a run.

```
S07:03:05,E07:10:03,R00:06:58,W00:00:00,H000654,B02,C002138,I00:00:00;3360
TC0009,TE0000,TT00:01:16;3361
LC0000,LE0000,LT00:24:00;3362
ST2102:07:23,2091:42:15,1914:26:12;3363
OT2102:12:23,OH00930582,OC002137;3364
AR00:06:58,AH000654,AP999,ZS00000110,MOWORK,EC0000;3378
B001,Z00000010,D00.800,H000765,X043.642Y153.321;3380
B002,Z00000010,D00.800,H001852,X123.321Y182.443;3381
```

Related topics

CNC command "DNCM0: DNC Transmission Mode 0", page 73

CNC command "DNCM2: DNC Transmission Mode 2", page 76

CNC command "DNCM3: DNC Transmission Mode 3", page 78

CNC command "DNCM4: DNC Transmission Mode 4", page 81

DNCM2: DNC Transmission Mode 2

DNC transmission mode 2

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
DNCM2	Set DNC transmission mode 2	10.01.001

The CNC command DNCM2 defines that a message is transmitted via the DNC interface after an event has occurred and that a set of messages is generated and transmitted via the DNC interface at the end of a run.

For more details refer to the description of the CNC command DNCM,.

Syntax of a single message

eventTxt;*eventNo*

Meaning of the elements:
- *eventTxt* = event message as plain text
- *eventNo* = event number of the event message

Example for a single message

The <STOP> key is pressed during the execution of a part program. The following event message is generated:

STOP;3016

Note: To get the real event number, the value 3000 must be subtracted from every event number (3016 − 3000 = 16).

Syntax of the event messaged at the end of a run

OPID setting;3360
progline1;3361
progline2;3362
progline3;3363
S*hh:mm:ss*,E*hh:mm:ss*,R*hh:mm:ss*,W*hh:mm:ss*,H*hhhhhh*,B*bbbb*,C*cccccc*,I*hh:mm:ss*;3364
TC*nnnn*,TE*eeee*,TT*hh:mm:ss*;3365
LC*nnnn*,LE*eeee*,LT*hh:mm:ss*;3366
ST*hh:mm:ss*,*hh:mm:ss*,…;3367
OT*hh:mm:ss*,OH*hhhhhhhh*,OC*cccccccc*;3368
AR*hh:mm:ss*,AH*hhhhhh*,AP*ppp*,ZS*zzzzzzzz*,*EreignisTxt*,EC*nnnn*;3378
B*nnn*,Z*zzzzzzzz*,T*nn*,H*hhhhhh*,X*x.xxx*Y*y.yyy*;3380

Meaning of the elements
- 1st line: user identification, code 3360
- 2nd to 4th line: program lines, codes 3361, 3362 and 3363
- 5th line: times and counters, code 3364
  - S = start time
  - E = end time
  - R = duration of the run
  - W = waiting time between two runs
  - H = number of drill strokes
  - B = number of broken tools (see also the line with the broken tool message)
  - C = length of all routing tracks in millimeters
  - I = waiting time during the run
- 6th line: tool change, code 3365
  - TC = number of tool changes
  - TE = number of tool change errors
  - TT = duration of all tool changes
- 7th line: loading processes, code 3366

  - LC = number of loading processes
  - LE = number of loading errors
  - LT = duration of all loading errors
- 8th line: running times of the spindles since initial operation, code 3367
  - ST = total running time of all spindles (the indicated time are separated by a comma)
- 9th line: counter values since initial operation, code 3368
  - OT = total execution time
  - OH = total number of drill strokes
  - OC = length of all routing tracks in meters
- 10th line: periodic messages, code 3378
  - AR = duration of a run
  - AH = number of drill strokes
  - AP = current value of the progress bar
  - ZS = spindle selection
  - *eventTxt* = current event message
  - EC = current event number
- beginning from the 11th line: broken tool message for every broken tool, codes 3380, ⋯
  - B = number of the broken tool
  - Z = spindle number as binary value
  - T or D = current tool number or current tool diameter
  - H = drill stroke counter for broken tool
  - XY = position of the broken tool

Example for a set of messages at the end of a run

The following event messages are generated and transmitted at the end of a run.

ABC;3360
(TEST);3361
X12.345Y23.456T1M31;3362
X12.567Y23.456;3363
S07:03:05,E07:10:03,R00:06:58,W00:00:00,H000654,B02,C002138,I00:00:00;3360
TC0009,TE0000,TT00:01:16;3361
LC0000,LE0000,LT00:24:00;3362
ST2102:07:23,2091:42:15,1914:26:12;3363
OT2102:12:23,OH00930582,OC002137;3364
AR00:06:58,AH000654,AP999,ZS00000110,MOWORK,EC0000;3378
B001,Z00000010,D00.800,H000765,X043.642Y153.321;3380
B002,Z00000010,D00.800,H001852,X123.321Y182.443;3381

Related topics

CNC command "DNCM0: DNC Transmission Mode 0", page 73

CNC command "DNCM1: DNC Transmission Mode 1", page 75

CNC command "DNCM3: DNC Transmission Mode 3", page 78

CNC command "DNCM4: DNC Transmission Mode 4", page 81

DNCM3: DNC Transmission Mode 3

DNC transmission mode 3

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
DNCM3	Set DNC transmission mode 3	10.01.001

The CNC command DNCM3 defines that a message is transmitted via the DNC interface after an event has occurred and that a set of messages is generated and transmitted via the DNC interface at the end of a run.

For more details refer to the description of the CNC command DNCM,.

Syntax of a single message

progName * _eventTxt_;_eventNo_

Meaning of the elements:

- _progName_ = file name of the part program
- * = separator between the file name and the event message
- _eventTxt_ = event message as plain text
- _eventNo_ = event number of the event message

Example for a single message

The <STOP> key is pressed during the execution of a part program "DRILL.SM5". The following event message is generated:

DRILL.SM5 * STOP;3016

Note: To get the real event number, the value 3000 must be subtracted from every event number (3016 − 3000 = 16).

Syntax of the event messages at the end of a run

progName * _OPID setting_;3360
progName * _progline1_;3361
progName * _progline2_;3362
progName * _progline3_;3363
progName * S_hh:mm:ss_,E_hh:mm:ss_,R_hh:mm:ss_,W_hh:mm:ss_,H_hhhhhh_,B_bbbb_,C_cccccc_,I_hh:mm:ss_;3364
progName * TC_nnnn_,TE_eeee_,TT_hh:mm:ss_;3365
progName * LC_nnnn_,LE_eeee_,LT_hh:mm:ss_;3366
progName * ST_hh:mm:ss_,_hh:mm:ss_,_…_;3367
progName * OT_hh:mm:ss_,OH_hhhhhhhh_,OC_cccccccc_;3368
progName * AR_hh:mm:ss_,AH_hhhhhh_,AP_ppp_,ZS_zzzzzzzz_,_eventTxt_,EC_nnnn_;3378
progName * B_nnn_,Z_zzzzzzzz_,T_nn_,H_hhhhhh_,X_x.xxx_Y_y.yyy_;3380

The file name of the part program is transmitted at the beginning of every message line. The asterisk (*) serves as separator.

Meaning of the elements

- 1st line: user identification, code 3360
- 2nd to 4th line: program lines, codes 3361, 3362 and 3363
- 5th line: times and counters, code 3364
  - S = start time
  - E = end time
  - R = duration of the run
  - W = waiting time between two runs
  - H = number of drill strokes
  - B = number of broken tools (see also the line with the broken tool message)
  - C = length of all routing tracks in millimeters
  - I = waiting time during the run
- 6th line: tool change, code 3365

  - TC = number of tool changes
  - TE = number of tool change errors
  - TT = duration of all tool changes
- 7th line: loading processes, code 3366
  - LC = number of loading processes
  - LE = number of loading errors
  - LT = duration of all loading errors
- 8th line: running times of the spindles since initial operation, code 3367
  - ST = total running time of all spindles (the indicated time are separated by a comma)
- 9th line: counter values since initial operation, code 3368
  - OT = total execution time
  - OH = total number of drill strokes
  - OC = length of all routing tracks in meters
- 10th line: periodic messages, code 3378
  - AR = duration of a run
  - AH = number of drill strokes
  - AP = current value of the progress bar
  - ZS = spindle selection
  - *eventTxt* = current event message
  - EC = current event number
- beginning from the 11th line: broken tool message for every broken tool, codes 3380, ⋯
  - B = number of the broken tool
  - Z = spindle number as binary value
  - T or D = current tool number or current tool diameter
  - H = drill stroke counter for broken tool
  - XY = position of the broken tool

Example for a set of messages at the end of a run

The following event messages are generated and transmitted at the end of a run.

```
DRILL.SM5 * ABC;3360
DRILL.SM5 * (TEST);3361
DRILL.SM5 * X12.345Y23.456T1M31;3362
DRILL.SM5 * X12.567Y23.456;3363
DRILL.SM5 * S07:03:05,E07:10:03,R00:06:58,W00:00:00,H000654,B02,C002138,I00:00:00;3360
DRILL.SM5 * TC0009,TE0000,TT00:01:16;3361
DRILL.SM5 * LC0000,LE0000,LT00:24:00;3362
DRILL.SM5 * ST2102:07:23,2091:42:15,1914:26:12;3363
DRILL.SM5 * OT2102:12:23,OH00930582,OC002137;3364
DRILL.SM5 * AR00:06:58,AH000654,AP999,ZS00000110,MOWORK,EC0000;3378
DRILL.SM5 * B001,Z00000010,D00.800,H000765,X043.642Y153.321;3380
DRILL.SM5 * B002,Z00000010,D00.800,H001852,X123.321Y182.443;3381
```

Related topics

CNC command "DNCM0: DNC Transmission Mode 0", page 73

CNC command "DNCM1: DNC Transmission Mode 1", page 75

CNC command "DNCM2: DNC Transmission Mode 2", page 76

CNC command "DNCM4: DNC Transmission Mode 4", page 81

DNCM4: DNC Transmission Mode 4

DNC transmission mode 4

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
DNCM4	Set DNC transmission mode 4	10.12.000

Requirement

Software that can evaluate DNC event messages is installed on the server computer (e.g. SMWinDNC by SIEB & MEYER).

The CNC command DNCM4 defines that a message is transmitted via the DNC interface after an event has occurred and that a set of messages is generated and transmitted via the DNC interface at the end of a run.

For more details refer to the description of the CNC command DNCM,.

Syntax of a single message

dd/mm/yy hh:mm:ss progName * _eventTxt_;_eventNo_

Meaning of the elements:

- _dd/mm/yy_ = system date of the CNC
- _hh:mm:ss_ = system time of the CNC
- _progName_ = file name of the part program
- * = separator between the file name and the event message
- _eventTxt_ = event message as plain text
- _eventNo_ = event number of the event message

Example for a single message

The <STOP> key is pressed during the execution of a part program "DRILL.SM5" on 10-05-23 at 02:21 p.m. The following event message is generated:

23/05/10 14:21:35 DRILL.SM5 * STOP;3016

Note: To get the real event number, the value 3000 must be subtracted from every event number (3016 − 3000 = 16).

Syntax of the event messages at the end of a run

dd/mm/yy hh:mm:ss progName * OPID setting;3360
dd/mm/yy hh:mm:ss progName * progline1;3361
dd/mm/yy hh:mm:ss progName * progline2;3362
dd/mm/yy hh:mm:ss progName * progline3;3363
dd/mm/yy hh:mm:ss progName * Shh:mm:ss,Ehh:mm:ss,Rhh:mm:ss,Whh:mm:ss,Hhhhhhh,Bbbbb,Ccccccc,Ihh:mm:ss;3364
dd/mm/yy hh:mm:ss progName * TCnnnn,TEeeee,TThh:mm:ss;3365
dd/mm/yy hh:mm:ss progName * LCnnnn,LEeeee,hh:mm:ss;3366
dd/mm/yy hh:mm:ss progName * SThh:mm:ss,hh:mm:ss,…;3367
dd/mm/yy hh:mm:ss progName * OThh:mm:ss,OHhhhhhhhh,OCcccccccc;3368
dd/mm/yy hh:mm:ss progName * ARhh:mm:ss,AHhhhhhh,APppp,ZSzzzzzzzz,eventTxt,ECnnnn;3378
dd/mm/yy hh:mm:ss progName * Bnnn,Zzzzzzzzz,Tnn,Hhhhhhh,Xx.xxxYy.yyy;3380

The date, time and file name of the part program are transmitted at the beginning of every message line. The asterisk (*) serves as separator.

Meaning of the elements

- 1st line: user identification, code 3360
- 2nd to 4th line: program lines, codes 3361, 3362 and 3363
- 5th line: times and counters, code 3364
  - S = start time
  - E = end time
  - R = duration of the run
  - W = waiting time between two runs
  - H = number of drill strokes
  - B = number of broken tools (see also the line with the broken tool message)
  - C = length of all routing tracks in millimeters
  - I = waiting time during the run
- 6th line: tool change, code 3365
  - TC = number of tool changes
  - TE = number of tool change errors
  - TT = duration of all tool changes
- 7th line: loading processes, code 3366
  - LC = number of loading processes
  - LE = number of loading errors
  - LT = duration of all loading errors
- 8th line: running times of the spindles since initial operation, code 3367
  - ST = total running time of all spindles (the indicated time are separated by a comma)
- 9th line: counter values since initial operation, code 3368
  - OT = total execution time
  - OH = total number of drill strokes
  - OC = length of all routing tracks in meters
- 10th line: periodic messages, code 3378
  - AR = duration of a run
  - AH = number of drill strokes
  - AP = current value of the progress bar
  - ZS = spindle selection
  - *eventTxt* = current event message
  - EC = current event number
- beginning from the 11th line: broken tool message for every broken tool, codes 3380, ⋯
  - B = number of the broken tool
  - Z = spindle number as binary value
  - T or D = current tool number or current tool diameter
  - H = drill stroke counter for broken tool
  - XY = position of the broken tool

Example for DNCM4: Messages at the end of a run

The following event messages are generated and transmitted at the end of a run.

23/05/10 14:21:35 DRILL.SM5 * ABC;3360
23/05/10 14:21:35 DRILL.SM5 * (TEST);3361
23/05/10 14:21:35 DRILL.SM5 * X12.345Y23.456T1M31;3362
23/05/10 14:21:35 DRILL.SM5 * X12.567Y23.456;3363
23/05/10 14:21:35 DRILL.SM5 * S07:03:05,E07:10:03,R00:06:58,W00:00:00,H000654,B02,C002138,I00:00:00;3360
23/05/10 14:21:35 DRILL.SM5 * TC0009,TE0000,TT00:01:16;3361
23/05/10 14:21:35 DRILL.SM5 * LC0000,LE0000,LT00:24:00;3362
23/05/10 14:21:35 DRILL.SM5 * ST2102:07:23,2091:42:15,1914:26:12;3363

```
23/05/10 14:21:35 DRILL.SM5 * OT2102:12:23,OH00930582,OC002137;3364
23/05/10 14:21:35 DRILL.SM5 * AR00:06:58,AH000654,AP999,ZS00000110,MOWORK,EC0000;3378
23/05/10 14:21:35 DRILL.SM5 * B001,Z00000010,D00.800,H000765,X043.642Y153.321;3380
23/05/10 14:21:35 DRILL.SM5 * B002,Z00000010,D00.800,H001852,X123.321Y182.443;3381
```

Related topics

CNC command "DNCM0: DNC Transmission Mode 0", page 73

CNC command "DNCM1: DNC Transmission Mode 1", page 75

CNC command "DNCM2: DNC Transmission Mode 2", page 76

CNC command "DNCM3: DNC Transmission Mode 3", page 78

NDDE: SMDDE Transmission

SMDDE transmission

SIEB & MEYER	Excellon	M49/M83
✓	✓	–

Command	Description	Version
NDDE	Deactivate SMDDE transmission and close connection.	10.12.027
NONDDE	Activate SMDDE transmission	10.12.027

☞ This command is only required, if modes are not set with the CNC command JOBL or NOJOBL.

NDDE

The command reception via SMDDE is deactivated.

- The CNC ignores commands sent by a higher-ranking controller (e.g. for activating automatic loading/unloading).
- In addition, the SMDDE connection is closed.

NONDDE

The command reception via SMDDE is activated.

- The CNC receives commands sent by a higher-ranking controller (e. g. for activating automatic loading/unloading).

7.B CNC 8x.00 script language

Abs: Calculate Absolute Value

Calculate absolute value

CNC 4x.00	CNC 8x.00
–	✓

Command	Description	Version
Abs(*expression*)	Result (without sign) of an integer expression	10.06

Argument	Description
expression	Integer value or mathematical expression

The function “Abs” converts the result of the integer expression into a value without sign (= absolute value).

- The integer values inlcude all numbers (without decimal positions).
- Number range: −2147483648, .., −3, −2, −1, 0 , 1, 2, 3, .. 2147483647

Example

Examples for the ABS function:

`Abs (-7)`	Result = 7
`Abs ( 3 * (%ZVALUE - 7000) )`	The result is always a positive value.

SetVal: Transfer Value to Handle

Transfer value to handle

CNC 4x.00	CNC 8x.00
–	✓

Command	Description	Version
SetVal(*expression*, *handle*)	Transfer expression to CNC.	10.06

Argument	Description
expression	Any expression
handle	CNC object

The function “SetVal” transfers a value to a CNC object (handle).

- An overview of all objects is provided in the section section 7.C "CNC Objects", page 91.

Example: singe condition

The user flag 7 shall be cleared depending on the current Z-value.

- User flag = HIGH, if the Z-value is smaller than 10.0 mm.
- User flag = LOW, if the Z-value is higher than 10.0 mm.
- Note:The internal counting process of user flags starts with "0".

To facilitate readability of the statement we recommend to program the desired flag number and to decrease the number in the expression by 1.

There are different way to program the solution.

```
SetVal( ComZ < 10.0, UserFlag(7-1) )
```

The user flag 7 is set/cleared depending on the result of the condition.

```
SetVal( if ComZ < 10.0 then true
else false, UserFlag(7-1) )
```

If Then, the user flag 7 is set and if False the user flag 7 is cleared.

```
SetVal( (ComZ < 10.0)? true : false,
UserFlag(5-1) )
```

If the condition is met, the first result value is transfered (true). Otherwise the 2nd result values is transfered (false).

Example: nested condition

Depending on the current position of the X-axis the counter 5 shall be set to a certain value.

- Counter = 1, if the X-axis position is smaller than or equal to 10.0 mm.
- Counter = 2, if the X-axis position is smaller than or equal to 20.0 mm.
- Counter = 3, if the X-axis position is higher than 20.0 mm.
- Note:The internal counting process of counters starts with "0".

To facilitate readability of the statement we recommend programming the desired counter number and decreasing the number in the expression by 1.

```
SetVal( if AuXPos <= 10.0 then 1
else ( if AuXPos <= 20.0 then 2 else
3 ), WZStdCnt(5-1) )
```

The counter 5 includes a value that depends on the X-position.

Trigger: Record Value Modification

Record value modification

CNC 4x.00	CNC 8x.00
–	✓

Command	Description	Version
Trigger(*cncObject*)	Generate protocol message when the value of cncObject changes.	10.06
TriggerChange(*cncObject*)	Generate protocol message when the value of cncObject changes.	10.06
T(*cncObject*)	Generate protocol message when the value of cncObject changes.	10.06

Argument	Description
cncObject	CNC object (see section 7.C "CNC Objects", page 91)

Generates a protocol message when the value of the CNC object changes.

Example

A protocol message shall be generated whenever the value of the Z-counter changes.

```
Trigger(%ZVALUE)
```

Monitor the Z-value.

TriggerDn: Record Value Decrease

Record value decrease

CNC 4x.00	CNC 8x.00
–	✓

Command	Description	Version
TriggerDn(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value decreases.	10.06
TriggerDown(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value decreases.	10.06
TDn(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value decreases.	10.06

Argument	Description
cncObject	CNC object (see section 7.C "CNC Objects", page 91)
initialValue	Start value, default value = FALSE
threshold	Threshold value, default value = 0

Generates a protocol message when the value of the CNC object is smaller than the start value.

- The start value can be any expression. If no start value exists, the value is "0" (= FALSE).
- The condition is only met, if the threshold value is not reached. If no min. deviation exists, the value is "1".
- Monitoring the state of an input, for example, requires programming a tolerance value (e.g. "0.5").

Example

A protocol message shall be generated whenever the state of the serial input IN07 changes from HIGH to LOW.

`TriggerDn(%SerInput(IN07), 1, 0.5)`	Monitors the input IN07.

TriggerUp: Record Value Increase

Record value increase

CNC 4x.00	CNC 8x.00
–	✓

Command	Description	Version
TriggerUp(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value increases.	10.06
TriggerDown(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value increases.	10.06
TDn(*cncObject*, [[*initialValue*][, *threshold*]])	Generate protocol message when the value increases.	10.06

Argument	Description
cncObject	CNC object (see section 7.C "CNC Objects", page 91)
initialValue	Start value, default value = FALSE
threshold	Threshold value, default value = 0

Generates a protocol message when the value of the CNC object is higher than the start value.

- The start value can be any expression. If no start value exists, the value is "0" (= FALSE).
- The condition is only met, if the threshold value is passed. If no threshold value exists, the value is "1" (= FALSE).
- Monitoring the state of an input, for example, requires programming a tolerance value (e.g. "0.5").

Example

A protocol message shall be generated whenever the state of the serial input IN07 changes from LOW to HIGH.

```
TriggerUp(%SerInput(IN07), 0, 0.5)
```

Monitors the input IN07.

7.B.1 Subobjects

The following subobjects can be programmed for the CNC objects “SerInput” and “SerOutput”.

List of subobjects for the CNC object “SerInput”.

SerInput	Explanation
AX1METER	Move the axes at 1 m/min, only.
ZPNEUUP	Switch at top for pneumatic Z-axis
ZPNEUDN	Switch at bottom for pneumatic Z-axis
SFUSTPMA	SFU stop from the machine
MANCOLOP	Spindle collet manually opened
SERVICE	CNC in service mode
EMERSTOP	Emergency stop
LIGHTBAR	Light barrier
POSITSTOP	Position stop
TABLESTOP	Table stop
SPINSEL1 .. SPINSEL12	Spindle selection 1 to 12
Z00.DRILBRK + z	Drill breakage input Z0
Z00.TCALSWT + z	Tool calibrating switch Z0
Z00.PRESWT + z	Preswitch Z0
X00.PRESWT + x	Preswitch X0
Y00.PRESWT + y	Preswitch Y0
A00.PRESWT + a	Preswitch Aux0
Z00.PRFTSWT + z	Pressure foot switch Z0
Z.DRILBRK	Drill breakage input Z1
Z.TCALSWT	Tool calibrating switch Z1
Z.PRESWT	Preswitch Z1
X.PRESWT	Preswitch X1
Y.PRESWT	Preswitch Y1
A.PRESWT	Preswitch Aux1
Z.PRFTSWT	Pressure foot switch Z1
IN00 .. IN127	Free input

List of subobjects for the CNC object “SerOutput”.

SerOutput	Explanation
ZPNEUM	Move the Z-axis pneumatically.
PRESBOOT	Pressure foot
COLOPEN	COLLET OPEN
PISTONUP	Piston up
SPSWMACH	Spindle switch of machine
VACUUMON	Vacuum unit on
M21OUTP	M21 output
G11OUTP	G11 output
CALIBRT	CNC is calibrating
WORKING	CNC is working
G83OUTP	Depth-controlled drilling/routing
CNCSTOP	CNC stops
TPULSEBF	Test pulse before drill stroke
TPULSEAF	Test pulse after drill stroke
DRHITCNT	Drill stroke counter
ROUTOUTP	Output for routing
RESETDSIM	Reset "Single diameter control"
PROGSTOP	Programmed stop (M20,M28,..)
Z00.SPINREL	Spindle relay Z0
Z.SPINREL	Spindle relay Z1
TOOLWRN	Output "Tool Warning"
MACHERR	Error to the machine
CNCRESET	ESC reset of the CNC
CNCRPM0	Flag of the CNC: speed 0
CNCRPMOK	Flag of the CNC: Speed reached
CNCRPMR	Flag of the CNC: frequency converter during acceleration of deceleration of the spindle
CNCFCOS	Flag of the CNC: output of frequency converter off
PRESLOCK	Pressure foot lock
PRESUP	Pressure foot up
STATOUT1 .. STATOUT12	Station select 1 .. 12
X00.POSOUTP + x	Flag: 'X01' > XOUTPOS
Y00.POSOUTP + y	Flag: 'Y01' > YOUTPOS
X.POSOUTP	Flag: 'X01' > XOUTPOS
Y.POSOUTP	Flag: 'Y01' > YOUTPOS
Z00.PRESHIGH	Pressure foot output Z0
Z.PRESHIGH	Pressure foot output Z1
OU00 .. OU127	Free output OU00 to OU127

7.C CNC Objects

This section describes name of CNC objects required for various tasks.

The name of a CNC object often must be transmitted during the communication with the CNC.

- To switch the user interface to another page, for example, the name of the page must be sent as CNC object.
- This section will help you to determine the correct object name.
- For further support, please contact the SIEB & MEYER service (e-mail: cnc-service@sieb-meyer.de).

7.C.1 Programming CNC Objects

A CNC object can be programmed for different sequences in various commands as transfer parameter.

- SMDDE command RUNTIMEVALUE. The result is provided in the return value of the command object RUNTIMEVALUE. The result is always of data type STRING.
- Sequence command SEND PC
- CNC command @RV,SETVAL
- Application Msg2SM (for example in a batch file).

Information on using CNC handles.

- A percent sign must precede the name of a CNC object (e.g. "%ZVALUE").
- A CNC object can be a part of an arithmetic or logical expression.
- (SMDDE) The result is provided in the return value of the command object RUNTIMEVALUE.
- (SMDDE) The result (also a numerical value) is always of the data type STRING
- (SMDDE) If the result is a boolean value (true, false), either "0" or "1" is returned (0 = false, 1 = true).

Example (TCP/IP): Determine Z-plane

In this example, the currently set Z-plane is requested.

- At the same time, the CNC shall add 2.0 mm to the Z-value.
- Numerical values must always be transmitted as integer values.
- The resolution depends on the CNC setting (0.001 mm, 0.0001 inch etc.)

Request package of the client

```
<SMDNCPACKET Value="626">
 <CNC Value="1">
  <EXECUTE Value="1235">
   <RUNTIMEVALUE Value="1">
    %ZVALUE+2000
   </RUNTIMEVALUE>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <EXECUTE Value="1235">
```

```
      <RUNTIMEVALUE Value="3450">
        %ZVALUE+2000
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

The result is returned in the attribute Value of the command object <RUNTIMEVALUE> (3450 = 3.45 mm).

Example (TCP/IP): Determine the setting of the CNC command BROK

In this example, the current setting of the broken tool monitoring is requested.

Request package of the client

```
<SMDNCPACKET Value="627">
  <CNC Value="1">
    <EXECUTE Value="1236">
      <RUNTIMEVALUE Value="1">
        %ComynBROK
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332557">
  <CNC Value="1">
    <EXECUTE Value="1236">
      <RUNTIMEVALUE Value="0">
        %ComynBROK
      </RUNTIMEVALUE>
    </EXECUTE>
  </CNC>
</SMDNCPACKET>
```

The result is returned in the attribute “Value” of the command object RUNTIMEVALUE (0 = NOBROK = tool monitoring is deactivated).

7.C.2 Determine the name of a CNC object.

Proceed as follows to determine a name of a CNC object of the CNC 8x.00.

- Log in as user with the access authorization "user level = 7".
- Select the navigator entry “Installation settings”, the entry “System - [Display and help]” and set the entry “Editing PAG sides and MEN menus allowed?” to “Yes”.
- Switch to the page on which you find the value of which you need the name (e.g. page “Working [Work - status]”).
- Double click with the right mouse button on the display field of the desired value(e.g. the H-value).
- A dialog window appears, in which the entry in the field “Handle” is of interest. Mark the entry (e.g. FIXZ 1) and copy it into the clipboard (CTRL+C).
- Click on “Cancel” to close the dialog.

- Select the navigator entry “Installation settings”, the entry “System - [Display and help]” and set the entry “Editing PAG sides and MEN menus allowed?” to “No”.
- Log in as user with standard access authorization.

Example (TCP/IP)

In this example, the currently set Z-plane is requested.

- At the same time, the CNC shall add 2.0 mm to the Z-value.
- Numerical values must always be transmitted as integer values.
- The resolution depends on the CNC setting (0.001 mm, 0.0001 inch etc.)

Request package of the client

```
<SMDNCPACKET Value="626">
 <CNC Value="1">
  <EXECUTE Value="1235">
   <RUNTIMEVALUE Value="1">
    %ZVALUE+2000
   </RUNTIMEVALUE>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

Response package of the CNC

```
<SMDNCPACKET Value="332553">
 <CNC Value="1">
  <EXECUTE Value="1235">
   <RUNTIMEVALUE Value="3450">
    %ZVALUE+2000
   </RUNTIMEVALUE>
  </EXECUTE>
 </CNC>
</SMDNCPACKET>
```

The result is returned in the attribute “Value” of the command object (3450 = 3.45 mm).

7.C.3 Names of CNC Screen Pages

Switching the screen page can be programmed in different sequences.

- SMDDE command CHANGECLIENT
- Sequence command SEND PC
- CNC command @RV,SETVAL
- Application Msg2SM (for example in a batch file).

Examples of page names in the CNC 8x.00.

Page context	Page	Page name
Work WORK e.g. WORK__ZEROPOINT	Work status	DEFAULT
	M-functions 1	MFUNC1
	M-functions 2	MFUNC2
	M-functions 3	MFUNC3
	Jog keys	Jogkeys
	Z-axis	Stroke

Page context	Page	Page name
	Zero point	ZEROPOINT
	Safety zone	SafetyZone
	Check	Check
	Job list	Joblist
	Automatic list	AutoList
	Board	PATS
Tools TOOLS e.g. TOOLS__TOOLS2	Tools (stat.)	TOOLS
	Tools (static 2)	TOOLS2
	Tools (dyn.)	TOOLSDYN
	Magazines	MAGAZINE
	Magazine links	MAGAZINE-LINKS
	T-conversion	TCONVERT
	Measurements	MEASUREMENT
	Tool tolerance	TOOLTOLERANCE
	Diameter table	DIAMETERTABLE
	Index table	INDEXTABLE
	Peck drilling	PECKDRILLING
	Broken drill	BROKENTOOLS
Diagnosis DIAG e.g. DIAG__USERFLAGS1	User flags 1-48	Userflags1
	User flags 49-96	Userflags2
	User flags 97-128	Userflags3
	Camera data	CameraData
	Camera settings	CameraSettings
Current program EDIT e.g. EDIT	*program name*	Program
System SYSTEM e.g. SYSTEM__PROTOCOLFILES	Overview	OVERVIEWOPERATINGDATA
	Current	OPERATINGDATA
	History	OPERATINGHISTORY
	Protocol file	PROTOCOLFILES

Determine page name

Proceed as follows to determine a page name of the CNC 8x.00 user interface:

- Log in as user with the access authorization "user level = 7".
- Choose “System”, page “View and help” and set the entry “Editing PAG sides and MEN menus allowed?” to “Yes”.
- Switch to the page of which you need the CNC-internal name (for example “Tools [magazines]”.
- Move the mouse pointer into the display area of the page.
- Press the keys SHIFT+CTRL and double click on the right-hand mouse button.
- The dialog “Select element in page ”… appears. If another dialog appears, click on “Cancel” and repeat the process.
- Click on the button “Page”.
- A dialog appears, in which the entry in the field “Remark” is of interest. Mark the entry (e.g. MAGAZINE) and copy it into the clipboard (CTRL+C).
- Click on “Cancel” to close the dialog.
- Choose “Installation settings”, page “System - [Display and help]” and set the entry “Editing PAG sides and MEN menus allowed?” to “No”.
- Log in as user with standard access authorization.

7.C.4 CNC Keynames

Pressing a CNC key can be programmed in different sequences.

- SMDDE command CNCKEY
- Sequence command SEND PC
- CNC command @RV,SETVAL
- Application Msg2SM (for example in a batch file).

Examples of keynames of the CNC 8x.00.

Keyname	Meaning
BLOCK	Activate/deactivate block operation (Note: Here you must determine the status first!)
CALIB	Calibrate all axes. Calibrate the Z-axes first, then the XY-axes.
CALIBA	Calibrate A-axis.
CALIBXY	Calibrate XY-axes
CALIBZ	Calibrate Z-axes.
NOTOOL	Activate/deactivate NOTOOL function (Note: Here you must determine the status first!)
OPTSTOP	Optional stop The execution is interrupted when the M20 command is executed.
START	(Re)start execution.
STOP	Interrupt execution

7.D PC Objects

This section describes name of PC objects required for various tasks.

The name of a PC object often must be transmitted during the communication with the CNC.

- PC objects are used for transmitting keystrokes to the CNC, for example.
- This section lists frequently used PC objects.
- For further support, please contact the SIEB & MEYER service (e-mail: cnc-service@sieb-meyer.de).

7.D.1 PC Commands

A PC command can be programmed in different sequences.

- SMDDE command PCCOMMAND
- Sequence command SEND PC
- CNC command @RV,SETVAL
- Application Msg2SM (for example in a batch file).

To prevent the display of dialog windows during the execution of a PC command, add the statement ".NODIALOG" behind the command (e.g. "SHUTDOWN.NODIALOG").

Examples of PC commands of the CNC 8x.00.

PC Commands	Meaning
SHUTDOWN	Shut down CNC software.
LOGOUT	Log out CNC user.

7.D.2 PC Keys

A PC keystroke can be programmed differently:

- with SMDDE command PCKEY
- with sequence command SEND PC
- with CNC command @RV,SETVAL
- for application Msg2SM (for example in a batch file).

Examples of PC keys in the CNC 8x.00:

Character(s)	PC Key
A .. Z	Letter keys A...Z
0 .. 9	Digits 0...9
\E	Escape key
\N	Enter key
\R	Enter key
\S	Space bar
\T	Tab key
\D	Cursor key "arrow down "
\U	Cursor key "arrow up"
\K	Cursor key "arrow left"
\L	Cursor key "arrow right"

7.E File Types

The table lists all file types that can be interpreted automatically by the CNC. The file type corresponds to the standard file type extension. The column “Explanation” shows where the file is saved in the CNC.

type	Explanation
ATP	Current tool parameters. The data are in the tool tables Tools (stat.) and Tools (dyn.).
DIA	Diameter table. The data are in the tool table Diameter.
INF	Information file. The data in the file are interpreted as CNC commands.
IDX	Diameter compensation values. The data are in the tool table Index.
PGM	Part program The data are in the tool tables Tools (stat.) and Tools (dyn.).
PRL	Job list. The data are in the automatic list.
SSU	Subprogram. The data are in the subprogram memory.
any	The files are interpreted as part program (see PGM type).

Tab. 1: Files types of the CNC 8x.00

8 Index: CNC Commands

9 Index: SMDDE Objects